Moscow Institute of Physics and Technology
(National Research University)

as a manuscript

Prokhorenkova Liudmila Aleksandrovna

MODELS OF COMPLEX NETWORKS AND ALGORITHMS ON GRAPHS

Dissertation Summary

for the purpose of obtaining academic degree
Doctor of Sciences in Computer Science

Moscow – 2021

Dissertation topic

Many real-world systems can be represented by networks whose vertices (nodes) are items and edges (links) are relations between these items. Countless empirical studies demonstrated that many observed networks share some typical properties: heavy-tailed degree distribution, small diameter, community structure, etc. Numerous random graph models have been proposed to reflect and predict important quantitative and topological aspects of real-world networks. Such models are of use in experimental physics, bioinformatics, information retrieval, and data mining [2, 9]. Studying the properties of complex networks and their models is essential for understanding their formation principles, predicting their future behavior, and developing effective algorithms.

Probably the most extensively studied property of networks is their vertex degree distribution. For the majority of studied real-world networks, the degree distribution was shown to approximately follow the power law [9]. This phenomenon is often explained by a principle called *preferential attachment* [6] that lies behind numerous models of complex networks [10, 17, 22].

Another key characteristic of real networks is their community structure characterized by the presence of highly interconnected groups of vertices relatively well separated from the rest of the network. For example, in social networks, communities are formed by users with similar interests; in citation networks, they represent the papers in different areas; on the Web, communities may correspond to pages on related topics, etc. The presence of communities highly affects, e.g., the promotion of products via viral marketing, the spreading of infectious diseases, computer viruses and information, and so on [47]. Thus, being able to identify communities is an important and actively studied research problem [14, 23, 30].

Beyond community detection, there are other important applications of graph analysis that will be discussed below in more detail: detecting influential nodes [45], graph-based nearest neighbor search [4], and others.

Objectives and goals of the dissertation The goal of the dissertation is twofold. First, analyze the properties of existing models of complex net-

works and develop new realistic models with desirable quantitative and topological properties. The second goal is to apply graph-based techniques to various practical tasks: community detection, publication date prediction, detecting influential nodes, and graph-based nearest neighbor search.

Key results

Models of complex networks and their analysis

- We propose a wide class of models called *Generalized Preferential Attachment* that includes many existing models. For the whole class, we rigorously analyze the degree distribution, local and global clustering coefficients, and degree correlations [20, 21, 35, 37].
- We prove a general result that the global clustering coefficient tends to zero for all graphs with power-law degree distribution with an infinite variance (assuming that degrees are sampled i.i.d. from a regularly varying distribution with an infinite variance) [36, 37].
- We analyze the asymptotic behavior of modularity (a measure that characterizes the community structure of a graph) in many random graph models, including d-regular, preferential attachment, and spatial preferential attachment graphs [38].
- We develop a novel principle called *preferential placement* that allows for generating structures with a power-law distribution of cluster sizes [13].
- We propose a new model called *recency-based preferential attachment*. This model is shown to give the best fit for the part of the Web related to media content. The basic properties of this model are theoretically analyzed [24, 40].

Community detection

- We theoretically and empirically compare likelihood-based community detection algorithms based on different null models. We propose a more theoretically grounded null model for this task [42].

- We systematically analyze the following problem: given only the infection times, find communities of highly interconnected nodes. We thoroughly compare existing and new approaches on several large datasets and show that the most stable performance and the most significant improvement over the current state-of-the-art are achieved by our proposed simple heuristic approaches [43].
- We address an important problem of choosing a proper performance measure for community detection algorithms. We use a theoretic approach: develop a list of desirable properties for performance measures and formally check each property for each relevant measure [15].

Other applications of graph analysis

- Using the proposed recency-based model, we propose a new algorithm for dating web pages [39].
- We propose a new algorithm for quick detection of high-degree nodes in complex networks [5].
- We obtain novel time and space complexity guarantees for graph-based nearest neighbor search algorithms [41].

Publications and approbation of research

First-tier publications

1. L. Ostroumova Prokhorenkova. General results on preferential attachment and clustering coefficient. Optimization Letters, 11(2):279–298, 2017. Web of Science Q2, Scopus Q2.
2. A. Krot and L. Ostroumova Prokhorenkova. Local clustering coefficient in generalized preferential attachment models. Internet Mathematics, 2017. Scopus Q2. (Main co-author; the author of this thesis formulated the problem, suggested the proof techniques, and guided the research.)

3. A. Krot and L. Ostroumova Prokhorenkova. Assortativity in generalized preferential attachment models. Internet Mathematics, 2017. Scopus Q2. (Main co-author; the author of this thesis formulated the problem, suggested the proof techniques, and guided the research.)

4. L. Ostroumova Prokhorenkova and E. Samosvat. Recency-based preferential attachment models. Journal of Complex Networks, 4(4):475-499, 2016. Scopus Q1, Web of Science. (Main co-author; the author of this thesis formulated and proved all theoretical results.)

5. L. Prokhorenkova and A. Tikhonov. Community detection through likelihood optimization: in search of a sound model. In The World Wide Web Conference, pages 1498–1508, 2019. CORE A*. (Main co-author; the author of this thesis suggested the ILFR model and conducted the theoretical analysis.)

6. L. Prokhorenkova, A. Tikhonov, and N. Litvak. Learning clusters through information diffusion. In The World Wide Web Conference, pages 3151–3157, 2019. CORE A*. (Main co-author; the author of this thesis developed the proposed algorithms and designed the experiments.)

7. M. Gösgens, A. Tikhonov, and L. Prokhorenkova. Systematic analysis of cluster similarity indices: How to validate validation measures. In International Conference on Machine Learning, pages 3799–3808. PMLR, 2021. CORE A*. (Main co-author; the author of this thesis formulated the problem and guided the research.)

8. L. Ostroumova Prokhorenkova, P. Prokhorenkov, E. Samosvat, and P. Serdyukov. Publication date prediction through reverse engineering of the Web. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pages 123–132, 2016. CORE A*. (Main co-author; the main idea was proposed in inseparable cooperation with E. Samosvat; the author of this thesis developed the algorithmic details, partially conducted experiments, and guided the research.)

9. K. Avrachenkov, N. Litvak, L. Ostroumova Prokhorenkova, and E. Suyargulova. Quick detection of high-degree entities in large directed networks. In 2014 IEEE International Conference on Data Mining, pages 20–29. IEEE, 2014. CORE A*. (Main co-author; the author of this thesis proposed the algorithm, performed theoretical analysis together with N. Litvak, and conducted the experimental analysis.)

10. L. Prokhorenkova and A. Shekhovtsov. Graph-based nearest neighbor search: From practice to theory. In International Conference on Machine Learning, pages 7803–7813. PMLR, 2020. CORE A*. (Main co-author; the author of this thesis suggested the problem, guided the research, and proved the results of Sections 4.1 and 4.2 — analysis of greedy graph-based search and all auxiliary results.)

Second-tier publications

1. L. Ostroumova, A. Ryabchenko, and E. Samosvat. Generalized preferential attachment: tunable power-law degree distribution and clustering coefficient. In International Workshop on Algorithms and Models for the Web-Graph, pages 185–202. Springer, 2013. Scopus. (The author of this thesis proved Theorem 3.)

2. L. Ostroumova Prokhorenkova. Global clustering coefficient in scale-free weighted and unweighted networks. Internet Mathematics, 12(1-2):54–67, 2016. Scopus.

3. L. Ostroumova Prokhorenkova, P. Prałat, and A. Raigorodskii. Modularity of complex networks models. Internet Mathematics, 2017. Web of Science. (The author of this thesis proved the results of Section 4.)

4. A. Dorodnykh, L. Ostroumova Prokhorenkova, and E. Samosvat. Preferential placement for community structure formation. In International Workshop on Algorithms and Models for the Web-Graph, pages 75–89. Springer, 2017. Scopus, Web of Science. (Main co-author; the author of this thesis formalized the model and proved the theoretical results.)

5. D. Lefortier, L. Ostroumova, and E. Samosvat. Evolution of the Media Web. In International Workshop on Algorithms and Models for the Web-Graph, pages 80–92. Springer, 2013. Scopus. (The author of this thesis, in inseparable cooperation with E. Samosvat, conducted the theoretical analysis of the proposed model.)

Invited talks at international conferences

1. August 2014 – "Crawling of new web pages", 8th Russian Summer School in Information Retrieval, Nizhny Novgorod, Russia.
2. June 2017 – "Some general results on preferential attachment and clustering coefficient", 7th International Conference on Network Analysis, Nizhny Novgorod, Russia.
3. September 2020 – "Some applications of graphs and probability theory to machine learning", 17th Workshop on Algorithms and Models for the Web Graph, Online.

Contributed talks at international conferences

1. November 2012 – "Generalized preferential attachment", Workshop on Internet Topology and Economics, Atlanta, USA.
2. June 2013 – "Preferential attachment models and their generalizations", Franco-Russian workshop on Algorithms, complexity and applications, Moscow, Russia.
3. August 2013 – "Recency-based preferential attachment models", International Conference on Random Structures and Algorithms, Poznan, Poland.
4. September 2013 – "Recency-based preferential attachment models", Palanga Conference in Combinatorics and Number Theory, Palanga, Lithuania.
5. October 2013 – "Timely crawling of high-quality ephemeral new content", International conference on Machine learning and Very Large Data Sets, Moscow, Russia.

6. October 2013 – "Generalized preferential attachment", Workshop on Random Graphs and their Applications, Moscow, Russia.

7. October 2013 – "Timely crawling of high-quality ephemeral new content", ACM International Conference on Information and Knowledge Management, San Francisco, USA.

8. December 2013 – "Generalized preferential attachment", Workshop on Algorithms and Models for the Web Graph, Cambridge, USA.

9. December 2013 – "Evolution of the Media Web", Workshop on Algorithms and Models for the Web Graph, Cambridge, USA.

10. April 2014 – "Crawling Policies Based on Web Page Popularity Prediction", 36th European Conference on Information Retrieval, Amsterdam, Netherlands.

11. July 2014 – "Model of the Media Web and its application to crawling of ephemeral web pages", Talk at the 11th International Vilnius Conference on Probability Theory and Mathematical Statistics, Vilnius, Lithuania.

12. July 2014 – "Recency-based preferential attachment models", International Conference Sum(m)it240, Budapest, Hungary.

13. December 2014 – "Quick Detection of High-degree Entities in Large Directed Networks", IEEE International Conference on Data Mining, Shenzhen, China.

14. December 2014 – "Global clustering coefficient in scale-free networks", 11th Workshop on Algorithms and Models for the Web Graph, Beijing, China.

15. May 2015 – "Global clustering coefficient in scale-free networks", 5th International Conference on Network Analysis, Nizhny Novgorod, Russia.

16. April 2016 – "Global clustering coefficient in scale-free networks", Workshop Critical and collective effects in graphs and networks, Moscow, Russia.

17. May 2017 – "Modularity of complex networks models", Second workshop Critical and collective effects in graphs and networks, Moscow, Russia.

18. June 2017 – "Preferential placement for community structure formation", 14th Workshop on Algorithms and Models for the Web Graph, Toronto, Canada.

19. July 2017 – "Modularity of random graph models", 39th Conference on Stochastic Processes and their Applications, Moscow, Russia.

20. August 2017 – "Modularity of random graph models", 18th International Conference on Random Structures and Algorithms, Gniezno, Poland.

21. May 2018 – "Community detection through likelihood optimization: in search of a sound model", Workshop on graphs, networks, and their applications, Moscow, Russia.

22. May 2019 – "Learning clusters through information diffusion", Conference on graphs, networks, and their applications, Moscow, Russia.

23. May 2019 – "Community detection through likelihood optimization: in search of a sound model", The Web Conference, San Francisco, USA.

24. May 2019 – "Learning clusters through information diffusion", The Web Conference, San Francisco, USA.

25. July 2019 – "Using synthetic networks for parameter tuning in community detection", 16th Workshop on Algorithms and Models for the Web Graph, Brisbane, Australia.

26. July 2020 – "Graph-based nearest neighbor search: from practice to theory", 37th International Conference on Machine Learning (ICML), Online.

27. July 2021 – "Systematic Analysis of Cluster Similarity Indices: How to Validate Validation Measures", 38th International Conference on Machine Learning (ICML), Online.

Analysis of complex networks models

The evolution of complex networks attracted much attention in recent years. In particular, numerous random graph models have been proposed to reflect and predict important quantitative and topological aspects of growing real-world networks. Studying such models and their properties is extremely important as it helps to understand fundamental principles underlying the formation of complex networks, predict the future behavior of networks, and construct efficient algorithms for network analysis.

Generalized Preferential Attachment

The results of this section are based on the papers [20, 21, 35, 37].

The most extensively studied property of networks is their vertex degree distribution. For the majority of studied real-world networks, the portion of vertices of degree d decreases approximately as $d^{-\gamma-1}$, usually with $1 < \gamma < 2$ [9]. Such networks are often called scale-free. Regarding the *cumulative* degree distribution, the portion of vertices of degree greater than d decreases as $d^{-\gamma}$. The most well-known approach to the modeling of complex networks with a power-law degree distribution is the *preferential attachment* [6]. The main idea of this approach is to add vertices one by one, and each new vertex connects to several existing vertices with probabilities proportional to their degrees. Numerous models are based on the idea of preferential attachment: Bollobás–Riordan [10], Buckley–Osthus [11], Holme–Kim [17], RAN [48], and many others. We propose a unified framework that allows for obtaining rigorous theoretical results simultaneously for all such models.

PA-class of models

Let G_m^n ($n \geqslant n_0$) be a graph with n vertices $\{1, \ldots, n\}$ and mn edges obtained as a result of the following process. We start at the time n_0 from an arbitrary graph $G_m^{n_0}$ with n_0 vertices and mn_0 edges. On the $(n+1)$-th step ($n \geq n_0$), we make the graph G_m^{n+1} from G_m^n by adding a new vertex $n+1$ and m edges connecting this vertex to some m vertices from the set $\{1, \ldots, n, n+1\}$. We denote by d_v^n the degree of a vertex v in G_m^n. If for

some constants A and B the following conditions are satisfied

$$\mathrm{P}\left(d_v^{n+1} = d_v^n \mid G_m^n\right) = 1 - A\frac{d_v^n}{n} - B\frac{1}{n} + O\left(\frac{\left(d_v^n\right)^2}{n^2}\right),\ 1 \leqslant v \leqslant n\ , \quad (1)$$

$$\mathrm{P}\left(d_v^{n+1} = d_v^n + 1 \mid G_m^n\right) = A\frac{d_v^n}{n} + B\frac{1}{n} + O\left(\frac{\left(d_v^n\right)^2}{n^2}\right),\ 1 \leqslant v \leqslant n\ , \quad (2)$$

$$\mathrm{P}\left(d_v^{n+1} = d_v^n + j \mid G_m^n\right) = O\left(\frac{\left(d_v^n\right)^2}{n^2}\right),\ 2 \leqslant j \leqslant m,\ 1 \leqslant v \leqslant n\ , \quad (3)$$

$$\mathrm{P}(d_{n+1}^{n+1} = m + j) = O\left(\frac{1}{n}\right),\ 1 \leqslant j \leqslant m\ , \quad (4)$$

we say that the random graph process G_m^n is a model from the PA-class. Here we require $2mA + B = m$ and $0 \leqslant A \leqslant 1$.

Even if we fix A and m, we still do not specify a concrete procedure for constructing a network since we do not completely define the joint distribution of m endpoints of new edges. Therefore, there is a range of models possessing very different properties and satisfying the conditions (1)–(4). For example, the Bollobás–Riordan [10], the Holme–Kim [17], and the RAN [48] models belong to the PA-class with $A = 1/2$ and $B = 0$. The Buckley–Osthus model [11] belongs to the PA-class with $A = \frac{1}{2+\beta}$ and $B = \frac{m\beta}{2+\beta}$.

It turns out that some rigorous results can be proven for the whole PA-class without specifying a concrete model. Such results generalize previous theoretical analyses made for each model independently.

Degree distribution in the PA-class

Denote by $N_n(d)$ the number of vertices of a given degree d in G_m^n. We prove the following result on the expectation of $N_n(d)$.

Theorem 1. *For every $d \geqslant m$ we have $\mathbb{E}N_n(d) = c(m,d)\left(n + O\left(d^{2+\frac{1}{A}}\right)\right)$, where*

$$c(m,d) = \frac{\Gamma\left(d + \frac{B}{A}\right)\Gamma\left(m + \frac{B+1}{A}\right)}{A\Gamma\left(d + \frac{B+A+1}{A}\right)\Gamma\left(m + \frac{B}{A}\right)} \overset{d\to\infty}{\sim} \frac{\Gamma\left(m + \frac{B+1}{A}\right) d^{-1-\frac{1}{A}}}{A\Gamma\left(m + \frac{B}{A}\right)}$$

and $\Gamma(x)$ is the gamma function.

We also show that the number of vertices of a given degree d is concentrated around its expectation.

Theorem 2. *For any model from the PA-class and for any $\delta > 0$ there exists a function $\varphi(n) = o(1)$ such that*

$$\lim_{n\to\infty} \mathrm{P}\left(\exists\, d \leqslant n^{\frac{A-\delta}{4A+2}} : |N_n(d) - \mathbb{E}N_n(d)| \geqslant \varphi(n)\,\mathbb{E}N_n(d)\right) = 0\,.$$

Theorems 1 and 2 imply that the degree distribution in G_m^n follows (asymptotically) the power law with the parameter $1+\frac{1}{A}$. So, the *cumulative* degree distribution follows the power law with the parameter $\gamma = \frac{1}{A}$.

Clustering coefficient in the PA-class

Another important characteristic of a network is its clustering coefficient, a measure capturing the tendency of a network to form clusters, densely interconnected sets of vertices. Several definitions of the clustering coefficient can be found in the literature, for instance, the *global clustering coefficient* and the *average local clustering coefficient*. The global clustering coefficient $C_1(G)$ is the ratio of three times the number of triangles to the number of pairs of adjacent edges in G. The average local clustering coefficient is defined as follows: $C_2(G) = \frac{1}{n}\sum_{i=1}^{n} C(i)$, where $C(i) = \frac{T^i}{P_2^i}$ is the local clustering coefficient for a vertex i, T^i is the number of edges between the neighbors of the vertex i, and P_2^i is the number of pairs of neighbors. It is believed that for many real-world networks, both the average local and the global clustering coefficients tend to a non-zero limit as the networks become large. Thus, a natural question is: can we say something about the clustering coefficient in the whole PA-class?

T-subclass It turns out that models from the PA-class may have very different clustering coefficients even for fixed parameters A and m. Therefore, in order to be able to analyze the behavior of the clustering coefficients, we have to add some additional constraints. Thus, we define a *T-subclass*.

In order to belong to the T-subclass, a model has to satisfy the following property in addition to (1)–(4):

$$\mathrm{P}\left(d_i^{n+1} = d_i^n + 1, d_j^{n+1} = d_j^n + 1 \mid G_m^n\right) = e_{ij}\frac{D}{mn} + O\left(\frac{d_i^n d_j^n}{n^2}\right)\,. \qquad (5)$$

Here e_{ij} is the number of edges between vertices i and j in G_m^n and D is a non-negative constant. Note that this property still does not define the correlation between m edges completely. Note that the Bollobás-Riordan and the Buckley–Osthus models belong to the T-subclass with $D = 0$, the Holme–Kim model with $D = P_t \cdot (m-1)$, and the RAN model with $D = 3$.

Global clustering coefficient in the T-subclass Let us first discuss the behavior of the global clustering coefficient $C_1(G_m^n)$. The following theorem holds.

Theorem 3. *Let G_m^n belong to the T-subclass with $D > 0$. Fix any $\varepsilon > 0$, then*

(1) If $2A < 1$, then ***whp*** $\frac{6(1-2A)D-\varepsilon}{m(4(A+B)+m-1)} \leqslant C_1(G_m^n) \leqslant \frac{6(1-2A)D+\varepsilon}{m(4(A+B)+m-1)}$;

(2) If $2A = 1$, then ***whp*** $\frac{6D-\varepsilon}{m(4(A+B)+m-1)\log n} \leqslant C_1(G_m^n) \leqslant \frac{6D+\varepsilon}{m(4(A+B)+m-1)\log n}$;

(3) If $2A > 1$, then ***whp*** $n^{1-2A-\varepsilon} \leqslant C_1(G_m^n) \leqslant n^{1-2A+\varepsilon}$.

Note that in some cases ($2A \geqslant 1$, i.e., $\gamma \leqslant 2$) the global clustering coefficient $C_1(G_m^n)$ tends to zero (for any D) as the number of vertices grows. A generalization of this result to all scale-free graphs will be discussed below.

Local clustering coefficient in the T-subclass Let us now discuss the behavior of the local clustering coefficient. First, one can easily show that $C_2(G_m^n)$ does not tend to zero if the condition (5) holds with $D > 0$.

However, a deeper analysis of the local clustering is possible if we consider the function $C_2(d)$ — the local clustering coefficient for the vertices of degree d. It was previously shown that in real-world networks $C_2(d)$ usually decreases as $d^{-\psi}$ with some parameter $\psi > 0$. For some networks, $C_2(d)$ scales as d^{-1} [26].

It turns out that in *all* models of the T-subclass the local clustering coefficient $C_2(d)$ asymptotically behaves as $\frac{2D}{Am} \cdot d^{-1}$. Formally, we prove the following result.

Theorem 4. *Let G_m^n belong to the T-subclass of the PA-class. Then, for any $\delta > 0$ there exists a function $\varphi(n) = o(1)$ such that*

(1) if $2A \leqslant 1$*:* $\lim_{n\to\infty} \mathrm{P}\left(\exists\, d \leqslant n^{\frac{A-\delta}{4A+2}} : |C_2(d) - F(d)| \geqslant \frac{\varphi(n)}{d}\right) = 0,$

(2) if $2A > 1$*:* $\lim_{n\to\infty} \mathrm{P}\left(\exists\, d \leqslant n^{\frac{A(3-4A)-\delta}{4A+2}} : |C_2(d) - F(d)| \geqslant \frac{\varphi(n)}{d}\right) = 0,$

where $F(d) = \frac{2D}{d\,(d-1)\,m}\left(m + \sum_{i=m}^{d-1} \frac{i}{Ai+B}\right) \overset{d\to\infty}{\sim} \frac{2D}{mA} \cdot d^{-1}$.

Thus, despite the fact that the T-subclass generalizes many different models, it is possible to analyze the local clustering coefficient for all these models. It turns out that $C_2(d)$ asymptotically decreases as $\frac{2D}{Am} \cdot d^{-1}$. In particular, this result implies that one cannot change the exponent -1 by varying the parameters A, D, and m. This basically means that preferential attachment models in general are not flexible enough to model $C(d) \propto d^{-\psi}$ with $\psi \neq 1$.

Assortativity

Next, we consider another key measure in complex networks analysis that is called *assortativity coefficient* and was first introduced by Newman [32] as the Pearson's correlation coefficient for the degrees of adjacent nodes. However, this coefficient is known to have certain drawbacks when applied to scale-free networks [28]. Thus, a more informative way of analyzing assortativity is to consider the behavior of $d_{nn}(d)$ — the average degree of a neighbor of a vertex of degree d. A graph is called assortative if $d_{nn}(d)$ is an increasing function of d, whereas it is referred to as disassortative when $d_{nn}(d)$ is a decreasing function of d.

It was previously empirically shown that in some real-world networks $d_{nn}(d)$ behaves as d^{ν} for some ν, which can be positive (assortative networks) or negative (disassortative networks) [9]. Assortativity has many applications; for instance, it can be used in epidemiology. In social networks, we usually observe assortative mixing, so diseases targeting high degree individuals are likely to spread to other high degree nodes. On the other hand, biological networks are usually disassortative; therefore, vaccination strategies targeting the high degree vertices may quickly destroy the epidemic.

We are able to analyze $d_{nn}(d)$ in the whole T-subclass of models for $\gamma > 3$ (the case of finite variance). We prove that the expectation of $d_{nn}(d)$

asymptotically behaves as $\log(d)$ (up to a constant multiplier). However, this approximation works reasonably well only for very large values of d and for $d < 10^4$ we observe a different behavior which may look like d^ν for some $\nu > 0$.

Let us formulate the main results, while the details can be found in [20].

Theorem 5. *Let G_m^n belong to the T-subclass of the PA-class with $A < \frac{1}{2}$. Then, for any $\varepsilon > 0$ and for every $d = d(n) \geqslant m$,*

$$\mathbb{E}d_{nn}(d) = F(d)\left(1 + O\left(\frac{n^{2A+\varepsilon}d^{2+\frac{1}{A}}}{n} + \frac{d^{2+\frac{1}{A}}\log n}{\sqrt{n}}\right)\right),$$

where $F(d) \overset{d\to\infty}{\sim} \frac{Am+B}{A} \cdot \log(d)$ and the non-asymptotic expression for $F(d)$ can be found in [20].

According to Theorem 5, all networks from the T-subclass with $A < \frac{1}{2}$ are assortative. However, $\mathbb{E}d_{nn}(d)$ increases slowly, as $\log(d)$, unlike d^ν in real-world networks. It is also worth noting that in Theorem 5 we analyze only the average value of $d_{nn}(d)$ and proving concentration is left for future research.

Global clustering coefficient in scale-free networks

The results of this section are based on the papers [36, 37].

While the degree distribution of preferential attachment models allows adjustment to reality, the clustering coefficient is challenging to model in some cases. Indeed, for most real-world networks, the parameter γ of their cumulative degree distribution belongs to the interval $(1, 2)$. However, as discussed above (Theorem 3), once $\gamma < 2$ in a preferential attachment model, the global clustering coefficient decreases as the graph grows, which does not correspond to the majority of real-world networks. The main reason for this behavior is that the number of edges added at each step is constant; consequently, the number of triangles can grow only linearly with the number of vertices n, while the number of pairs of adjacent edges grows as $n^{2/\gamma}$.

Under some assumptions on the degree sequence, we rigorously prove that a model with a power-law degree distribution, with $\gamma < 2$, and with an

asymptotically constant global clustering coefficient *cannot exist*. In order to do this, we consider a sequence of graphs G_n (n refers to the number of vertices) with degrees *independently sampled* from a *regularly varying* distribution F with parameter γ of the cumulative distribution. Regularly varying distributions form a broad class of heavy-tailed distributions and generalize power-law distributions. Then, we assume that for a given outcome of the degree sequence, a graph can be built in any arbitrary way. We prove that if a simple graph has degrees sampled from a regularly varying degree distribution with an infinite variance ($1 < \gamma < 2$), then the global clustering coefficient for any such sequence of graphs tends to zero with high probability. Note that we do not assume any random graph model here.

Theorem 6. *For any $\varepsilon > 0$ and any α such that $0 < \alpha < \frac{1}{\gamma+1}$ with probability $1 - O(n^{-\alpha})$ the global clustering coefficient of G_n satisfies the following inequality*

$$C_1(G_n) \leqslant n^{-\frac{(2-\gamma)}{\gamma(\gamma+1)}+\varepsilon}.$$

The obtained result is especially interesting due to the fact that in many observed networks, the global clustering coefficient is considerably high [33]. However, it is hard to compare the asymptotic result of Theorem 6 with empirical measurements made on finite networks. More exciting is the fact that there are models of complex networks with asymptotically power-law degree distribution with infinite variance and with non-vanishing global clustering coefficient. For instance, such results have been obtained for the random intersection graphs [8]. This contradiction can be explained by our formalization of "power-law degree distribution". Indeed, we assume that degrees are sampled independently from a regularly varying distribution. The independence assumption is quite restrictive and may not hold for some realistic random graph models. In our analysis, we use this assumption to show that the largest degrees are sufficiently large. Without the independence assumption, the largest degrees can be smaller (i.e., we may have a cut-off in the empirical degree sequence), while the empirical degree sequence is still asymptotically regularly varying.

In addition to the upper bound obtained for the global clustering coefficient, we also present an algorithm that allows constructing graphs with

nearly maximum (up to $n^{o(1)}$ multiplier) clustering coefficient for the considered sequence of graphs [36].

Analysis of modularity

The results of this section are based on the paper [38].

The clustering coefficient is a basic characteristic for analyzing the tendency of a network to form highly interconnected clusters. A more advanced measure allowing to characterize the community structure is *modularity* [34]. Modularity is at the same time a global criterion to define communities, a quality function of community detection algorithms, and a way to measure the presence of community structure in a network. Many community detection algorithms are based on finding partitions with high modularity.

The main idea behind modularity is to compare the actual density of edges inside communities with the density one would expect to have if the vertices were attached randomly, regardless of community structure. Formally, for a given partition $\mathcal{A} = \{A_1, \ldots, A_k\}$ of the vertex set $V(G)$, let

$$q_{\mathcal{A}} = \sum_{A \in \mathcal{A}} \left(\frac{e(A)}{|E(G)|} - \frac{\left(\sum_{v \in A} \deg(v)\right)^2}{4|E(G)|^2} \right), \tag{6}$$

where $E(G)$ denotes the set of edges in G, $e(A) = |\{uv \in E(G) : u, v \in A\}|$ is the number of edges in the graph induced by the set A, and $\deg(v)$ is the degree of a vertex v.

The *modularity* of a graph G is

$$q^*(G) = \max_{\mathcal{A}} q_{\mathcal{A}}(G).$$

If $q^*(G)$ approaches 1 (the maximum), we observe a strong community structure. Conversely, if $q^*(G)$ is close to zero, we are given a graph with no community structure.

We theoretically investigate modularity in random d-regular graphs, graphs with bounded average degree, the preferential attachment model [10], and the spatial preferential attachment model [1].

Random d-regular graphs We consider the probability space of *random d-regular graphs* with uniform probability distribution. This space is denoted by $\mathcal{G}_{n,d}$, and asymptotics are for $n \to \infty$ with $d \geqslant 2$ fixed, and n even if d is odd. We first obtain a numerical upper bound on the modularity of $\mathcal{G}_{n,d}$. We also prove the following weaker but explicit bound that can be obtained using the expansion properties of random d-regular graphs that follow from their eigenvalues.

Theorem 7. *Let $d \in \mathbb{N} \setminus \{1, 2\}$ and $\varepsilon > 0$ be an arbitrarily small constant. Then, a.a.s., $q^*(\mathcal{G}_{n,d}) \leqslant \frac{2}{\sqrt{d}}$.*

Graphs with bounded average degree For graphs with bounded average degree, the following theorem holds.

Theorem 8. *Let $\{G_n\}$ be a sequence graphs, G_n is a connected graph on n vertices with the average degree $\frac{2|E(G_n)|}{n} \leqslant D$ for some constant D, and maximum degree $\Delta = \Delta(G_n) = o(n)$. Then, $q^*(G_n) \geqslant \frac{2}{D} - O\left(\sqrt{\frac{\Delta}{n}}\right) = \frac{2}{D} - o(1)$.*

Preferential Attachment model Next, we analyze modularity in the preferential attachment model G_m^n [10]. The following theorem easily follows from Theorem 8 and the fact that a.a.s. $\Delta(G_m^n) = O\left(n^{\frac{1}{2}+2\varepsilon}\right)$ for any $\varepsilon > 0$.

Theorem 9. *For any $\varepsilon > 0$ a.a.s. $q^*(G_m^n) \geqslant \frac{1}{m} - O\left(n^{-1/4+\varepsilon}\right) = \frac{1}{m} - o(1)$.*

However, this bound is not sharp, and we also prove a stronger lower bound. Here we present only the asymptotic result as $m \to \infty$, while the complete statement and the numerical values for small m can be found in [38].

Theorem 10. *A.a.s. $q^*(G_m^n) = \Omega\left(1/\sqrt{m}\right)$.*

Regarding the upper bound, we the following holds.

Theorem 11. *For any $\varepsilon > 0$ a.a.s. $q^*(G_2^n) \leqslant \frac{15+\varepsilon}{16}$. For any $m \geqslant 3$ a.a.s. $q^*(G_m^n) \leqslant \frac{15}{16}$.*

Spatial Preferential Attachment model We also study a so-called Spatial Preferential Attachment model [1]. This model combines preferential attachment with geometry by introducing "spheres of influence" whose volume grows with the degree of a vertex. The parameters of the model are the number of vertices n, the dimension of a latent space m, the link probability $p \in [0, 1]$, and two constants A_1, A_2 defining the probability of attachment, such that $0 < A_1 < \frac{1}{p}$ and $A_2 > 0$.

The SPA model is known to produce scale-free networks, which exhibit many characteristics of real-life networks. The following theorem shows that the modularity of the SPA model is asymptotically one, unlike d-regular and preferential attachment graphs.

Theorem 12. *Let $p \in (0, 1]$, $A_1, A_2 > 0$, and suppose that $pA_1 < 1$. Then, a.a.s., the modularity of the SPA model is $1 - O\left(n^{\max\{-1/m, -1+pA_1\}/2} \log^{9/2} n\right) = 1 - o(1)$.*

Preferential placement

The results of this section are based on the paper [13].

An important aspect of complex networks modeling is generating realistic community structures. Several empirical studies have shown that the community structure of different real-world networks has some typical properties: e.g., the cumulative community size distribution obeys a power law with some parameter [3, 12, 16]. Unfortunately, the widely used preferential attachment model and many other models fail to provide the desired clustering structure.

We propose a process called *preferential placement* that naturally generates clustering structures. We assume that all vertices are embedded in $\mathbb{R}^d$ for some $d \geqslant 1$. One can think that coordinates of this space correspond to latent features of vertices. The vertices appear one by one, and their positions are defined according to preferential placement: each new vertex chooses a 'parent' among the existing vertices uniformly at random and is then placed uniformly at some distance from the parent. The distance is sampled from a distribution Ξ. We argue that in order to obtain a realistic clustering structure, one should take Ξ to be a heavy-tailed distribution. In this case, according to the procedure described above, new vertices will

usually appear in the dense regions, close to some previously added vertices; however, due to the heavy tail of Ξ, from time to time, we get outliers that originate new clusters.

Our empirical studies confirm that a reasonable clustering structure is produced if Ξ has a power-law distribution with a proper parameter. We observe that the distribution of the cluster sizes (if clusters are obtained via the DBSCAN algorithm) follows the power law. Moreover, the obtained structure is hierarchical, which agrees with numerous practical observations.

We then discuss why we expect to observe a power-law distribution of cluster sizes from a theoretical point of view. The main difficulty with the analysis of clustering structures is that there are no standard definitions of clusters, both in graphs and metric spaces. Thus, we can only give some insights into the fact that the proposed algorithm is expected to generate a power-law distribution of cluster sizes. Namely, we make some strong assumptions and then rigorously prove that the distribution follows a power law.

Formally, let $F_t(s)$ denote the number of clusters of size s at step t. We assume that all clusters can only grow, they cannot merge or split. Then, at step $t+1$, a new cluster appears with probability $p(t) = \frac{c}{t^\alpha}$, $c > 0$, $0 \leqslant \alpha \leqslant 1$. Finally, given that a vertex $t+1$ does not create a new cluster, the probability to join a cluster C with $|C| = s$ is equal to $\frac{s}{t}$. The last assumption is motivated by the observation that the probability to choose a parent from some cluster C with $|C| = s$ is equal to $\frac{s}{t}$ by the definition of the model. These assumptions are quite strong and even not very realistic, but they allow us to analyze the behavior of $F_t(s)$ formally. Namely, we prove the following theorem (see [13] for the full statement).

Theorem 13. *Under the assumptions above, the following holds.*

1. *If* $\alpha = 0$ *and* $0 < c < 1$, *then:* $\mathbb{E}F_n(s) \sim \frac{c\,\Gamma\left(2+\frac{1}{1-c}\right)}{(2-c)} \cdot \frac{n}{s^{1+\frac{1}{1-c}}}$.

2. *If* $0 < \alpha \leqslant 1$, *then for any* $\epsilon > 0$*:* $\mathbb{E}F_n(s) \sim \frac{c\,\Gamma(3-\alpha)}{2-\alpha} \cdot \frac{n^{1-\alpha}}{s^{2-\alpha}}$.

To sum up, if the probability $p(n)$ of creating a new cluster is of order $\frac{1}{n^\alpha}$ for $\alpha > 0$, then the distribution of cluster sizes follows a power law with

parameter $2-\alpha$ growing with $p(n)$ from 1 to 2; if $p(n)=c$, $0<c<1$, then the parameter grows with c from 2 to infinity. Informally, this theorem allows us to connect the distribution Ξ with the distribution of the obtained cluster sizes, since Ξ affects the probability $p(n)$.

Finally, note that preferential placement allows one to generate the positions of nodes in a latent space. Then, to obtain a graph, one can use any existing spatial graph model. In [13] we analyze several possible options and show that the obtained graphs indeed have desirable properties, including a realistic degree distribution and a power-law distribution of the cluster sizes.

Recency-based preferential attachment

The results of this section are based on the papers [24, 40].

We end the first part by describing a new principle called *recency-based preferential attachment*. This idea was suggested in [24] based on a detailed empirical study of the part of the Web related to media content. Using publicly available data, we analyze the evolution of incoming and outgoing links from and to media pages. In particular, in addition to the degree distribution, for graphs evolving in time, we also define a so-called *recency property*. Namely, denote by $e(T)$ the fraction of edges connecting nodes whose age difference is larger than T. We observe that media pages tend to connect to pages of similar age, and $e(T)$ decreases exponentially fast, which is not the case for preferential attachment models.

Thus, the idea is to generalize the preferential attachment principle in the way that the probability to cite a page p is proportional to the *attractiveness* of p, which is some function of $d(p)$ (current degree of p), $q(p)$ (intrinsic quality of p), and $a(p)$ (current age of p). Different attractiveness functions are considered:

$$\mathrm{attr}(p) = q(p)^{\alpha_1} \cdot d(p)^{\alpha_2} \cdot e^{-\frac{a(p)}{\tau}\cdot\alpha_3},$$

where $(\alpha_1,\alpha_2,\alpha_3)\in\{0,1\}^3$ and τ corresponds to the mean lifetime of the decaying attractiveness. For example, $\mathrm{attr}(p)=d(p)$ leads to preferential attachment, while $\mathrm{attr}(p)=q(p)\cdot d(p)$ leads to the fitness model.

To depict the recency property of the Media Web, one has to include the

recency factor $e^{-\frac{a(p)}{\tau}}$ in the attractiveness function. Our mean-field approximation analysis and computer simulations show that in order to have the power-law degree distribution with a realistic exponent, the attractiveness function $\mathrm{attr}(p) = q(p)\, e^{-\frac{a(p)}{\tau}}$ should be chosen. Moreover, the distribution of qualities q should follow the power law. The superiority of this form of the attractiveness function was also confirmed by analyzing the likelihood of real data given the model.

The proposed principle was further formalized and theoretically analyzed in [40]. Here, we focus on the attractiveness function $q(p) \cdot e^{-\frac{a(p)}{\tau}}$. Formally, we construct a sequence of random graphs $\{G_n\}$. The sequence is parametrized by a positive integer constant m (vertex outdegree) and an integer function $N(n)$. We also need a sequence of mutually independent random variables $\zeta_1, \zeta_2, \ldots$ with some given distribution taking positive values.

Each graph G_n is defined according to its own constructing procedure based on the idea of preferential attachment. At the beginning, we have two vertices and one edge between them (graph $\tilde{G}_2^n$). The first two vertices have inherent qualities $q(1) := \zeta_1$ and $q(2) := \zeta_2$. At the $t+1$-th step ($2 \leqslant t \leqslant n-1$) one vertex and m edges are added to $\tilde{G}_t^n$. New vertex $t+1$ has an inherent quality $q(t+1) := \zeta_{t+1}$. New edges are drawn independently, and they go from the new vertex to previous vertices. For each edge, the probability that it goes to a vertex i ($1 \leqslant i \leqslant t$) is equal to

$$\frac{\mathrm{attr}_t(i)}{\sum_{j=1}^{t} \mathrm{attr}_t(j)}, \text{ where } \mathrm{attr}_t(i) = q(i)\, e^{-\frac{t-i}{N(n)}}.$$

Further we assume that $N = N(n) \to \infty$ as n grows.

Assume that the random variables $\zeta_1, \zeta_2, \ldots$ have the Pareto distribution with the density function $f(x) = \frac{\gamma a^\gamma I[x>a]}{x^{\gamma+1}}$, where $\gamma > 1$, $a > 0$. Then, the expectation of the number of vertices with degree d in G_n (denoted by $N_n(d)$) decreases as $d^{-\gamma-1}$.

Theorem 14. *Let us define a constant α as follows: if $\gamma > 2$, then $\alpha = 2$; if $1 < \gamma \leqslant 2$, then α can be any constant such that $1 < \alpha < \gamma$. If $d = d(n)$*

increases with n *and* $d = o\left(\min\left\{\left(\frac{n}{N\log N}\right)^{\frac{1}{\gamma+1}},\ N^{\frac{\alpha-1}{\alpha+(\gamma+1)(\alpha+1)}}\right\}\right)$, *then*

$$\frac{\mathbb{E}[N_n(d)]}{n} = \frac{\gamma}{d^{\gamma+1}}\left(\frac{(\gamma-1)m}{\gamma}\right)^{\gamma}(1+o(1))\ .$$

The following theorem shows that the number of vertices of degree d is concentrated near its expectation.

Theorem 15. *For every* d *the following inequality holds:*

$$\mathrm{P}\left(|N_n(d) - \mathbb{E}[N_n(d)]| > \sqrt{Nn\log n}\right) = O\left(\frac{1}{\log n}\right)\ .$$

Note that for $d = o\left(\left(\frac{n}{N\log n}\right)^{\frac{1}{2(\gamma+1)}}\right)$ we have $\sqrt{Nn\log n} = o\left(\mathbb{E}[N_n(d)]\right)$ and Theorem 15 gives the concentration.

Finally, we show that the behavior of $e(T)$ is realistic, i.e., $e(T)$ decreases exponentially with T.

Theorem 16. *For any integer* T,

$$\mathbb{E}[e(T)] = e^{-\frac{T}{N}} + O\left(\frac{N}{n}\right)\ ,$$

$$\mathrm{P}\left(|e(T) - \mathbb{E}[e(T)]| \geqslant \sqrt{\frac{N\log n}{n}}\right) = O\left(\frac{1}{\log n}\right)\ .$$

Community detection

As discussed above, community structure is one of the most important graph properties. It is characterized by the presence of highly interconnected groups of vertices relatively well separated from the rest of the network. In social networks, communities (a.k.a. clusters) are formed by users with similar interests; in citation networks, they represent the papers in different areas; on the Web, communities may correspond to pages on related topics, etc. The presence of communities highly affects, e.g., the promotion of products via viral marketing, the spreading of infectious diseases, computer viruses, information, and so on. Identifying communities in a network

could help us exploit this network more effectively: clusters in citation graphs are helpful for finding similar scientific papers, discovering users with similar interests is important for targeted advertisement, clustering can also be used for network compression and visualization. Thus, in this section we discuss different aspects of *community detection* problem [14].

Community detection through likelihood optimization

The results of this section are based on the paper [42].

Among other algorithms proposed for community detection, the notable ones are methods based on statistical inference. In such methods, some underlying random graph model is assumed, the evidence is represented by the observed graph structure (its adjacency matrix), and the hidden variables to be inferred are the model's parameters together with community assignments. Such methods are appealing since they are theoretically sound and consistent: e.g., it has been proved that when the maximum-likelihood method is applied to networks generated from the same stochastic block model, it returns correct assignments of nodes to groups in the limit of large node degrees [7]. Also, likelihood can be used as a tool to formalize the notion of community.

The choice of the proper null model is essential for statistical inference algorithms as it highly affects their performance. The most commonly used model is called the *planted partition model* (PPM). In this model, the vertices are divided in k clusters and for each pair of vertices i, j we draw an edge between them with probability p_{in} if they belong to the same cluster and p_{out} otherwise, $p_{in} > p_{out}$.

However, the PPM model is unable to model the degree heterogeneity observed in most real-world networks. To overcome this issue, *degree-corrected stochastic block model* [18] and its simplified variant *degree-corrected planted partition model* (DCPPM) [31] were proposed. In DCPPM, the vertices are assigned to k clusters and edges are placed independently at random. The expected number of edges between vertices i and j is $\frac{d(i)d(j)}{2m}p_{in}$ if they belong to the same cluster or $\frac{d(i)d(j)}{2m}p_{out}$ otherwise. Here $d(i)$ is the desired degree of a vertex i and m is the total number of edges. However, we show that in DCPPM, for any reasonable choice of

p_{in} and p_{out}, the expected degree is not equal to $d(i)$. Motivated by this observation, as well as by the well-known LFR benchmark [23], we develop a one-parametric model which does not suffer from the above issue.

We argue that the inability of DCPPM to preserve the desired degree sequence is caused by the fact that the probability of an internal edge is independent of the size of the community and thus the expected fraction of internal edges *depends on the size of the community* the vertex belongs to. In contrast, similarly to the LFR model [23], we use the mixing parameter μ which controls this fraction and makes it equal for all vertices in the graph. Formally, we assume that all edges are independent and the expected number of edges between two vertices i and j is equal to $\frac{\mu d(i) d(j)}{2m}$ if $c(i) \neq c(j)$ or to $\frac{(1-\mu) d(i) d(j)}{D(C_q)} + \frac{\mu d(i) d(j)}{2m}$ if $c(i) = c(j) = q$. Here μ is the mixing parameter ($0 < \mu < 1$), $c(i)$ is the cluster assignment for a vertex i, and $D(C)$ is the sum of the degrees of all the vertices belonging to a cluster C. With this definition, the expected degree of a vertex i is equal to $d(i)$. Moreover, the proposed model has only one parameter μ instead of p_{in} and p_{out} in the planted partition models. We call the proposed model *Independent LFR* or ILFR.

We derive the exact formula of the likelihood for the proposed ILFR model. For simplicity, let us show its approximation:

$$\log L_{ILFR'}(\mathcal{C}, G, \mu) = m_{in} \log(1-\mu) + m_{out} \log \mu \\ - m_{out} \log 2m - \sum_C \frac{D_{in}(C)}{2} \log D(C) + \sum_i d(i) \log d(i) - m\,, \quad (7)$$

where m_{in} and m_{out} are the number of intra- and iter-cluster edges, respectively, and $D_{in}(C)$ is twice the number of edges induced by C. The optimal μ according to (7) is $\mu = \frac{m_{out}}{m}$.

The obtained likelihood allows us to apply likelihood-based methods to the ILFR model. Through extensive experiments, we compare the likelihood optimization algorithms based on three null models discussed above — PPM, DCPPM, and ILFR. We also demonstrate that the proposed ILFR model shows the best fit for various real-world networks according to the likelihood of observed structures.

Community detection based on cascade data

The results of this section are based on the paper [43].

Let us discuss a more challenging problem of inferring *community structure* of a given network based solely on *cascades* (e.g., information or epidemic) propagating through this network. Compared to the traditional community detection, the task is quite different because we do not have the network available; we have only cascade data observed on this network. For each cascade, we observe only infected nodes and their infection times.

Formally, assume that we observe a set of cascades $\mathcal{C} = \{C_1, \dots, C_r\}$ that propagate on a latent undirected network $G = (V, E)$ with $|V| = n$ nodes and $|E| = m$ edges. Each cascade $C \in \mathcal{C}$ is a record of observed node activation times, i.e., $C = \{(v_i, t^C_{v_i})\}_{i=1}^{n_C}$, where v_i is a node, $t^C_{v_i}$ is its activation time in C, $|C| = n_C$ is a size of a cascade. Note that we do not observe who infected whom.

We further assume that G is partitioned into communities: $\mathcal{A} = \{A_1, \dots, A_k\}$, $\cup_{i=1}^k A_i = V$, $A_i \cap A_j = \emptyset$ for $i \neq j$. We expect to observe a high intra-community density of edges compared to inter-community density. By observing only a set of cascades $\mathcal{C}$ we want to find a partition $\mathcal{A}'$ similar to $\mathcal{A}$.

We propose and analyze two types of approaches for this task: based on likelihood maximization under specific model assumptions and based on clustering of a surrogate network.

Approaches based on likelihood maximization are called ClustOpt and GraphOpt. In ClustOpt, we assume the cascade model in which each activated node can infect all other susceptible nodes independently after an exponentially distributed time. If a susceptible node belongs to the same community, then the infection rate is α_{in}, otherwise it is α_{out}, $\alpha_{out} < \alpha_{in}$. Epidemic stops at time T_{max}. This is a simplified epidemic model since spreading depends only on the community structure and not on the graph G. We derive the formula for the likelihood of cascades $L(\mathcal{C}, \mathcal{A})$ for this model. Note that the likelihood depends on the parameters α_{in} and α_{out}. We propose the following algorithm and refer to [43] for the details.

GraphOpt is based on a more complex epidemic model introduced in [44]. In this model, an activated node infects its neighbors after an

ALGORITHM 1: ClustOpt

1. Find initial partition $\mathcal{A}_{init}$;
2. Find $\hat{\alpha}_{in}, \hat{\alpha}_{out} = \arg\max_{\alpha_{in},\alpha_{out}} \log L(\mathcal{C}, \mathcal{A}_{init})$;
3. For fixed $\hat{\alpha}_{in}, \hat{\alpha}_{out}$ find $\hat{\mathcal{A}} = \arg\max_{\mathcal{A}} \log L(\mathcal{C}, \mathcal{A})$.

exponentially distributed time with intensity α. All nodes recover simultaneously at some threshold time T_{max} and the epidemic stops. For simplicity, we further assume T_{max} to be fixed, but our methods allow varying T_{max} for different epidemics. We propose an expectation-maximization-based method, where a graph G is a latent variable. We assume the following generative probabilistic process. For a given partition $\mathcal{A}$ of n nodes, we construct a graph G according to the ILFR model discussed in the previous section. Then, based on G, we generate a set of cascades $\mathcal{C}$ according to the epidemic model $\mathrm{P}(\mathcal{C}|G)$. We observe $\mathcal{C}$ and our aim is to recover the partition $\bar{\mathcal{A}}$ which maximizes the likelihood, i.e., $\bar{\mathcal{A}} = \arg\max_{\mathcal{A}} \mathrm{P}(\mathcal{C}|\mathcal{A})$. We propose the following algorithm and refer to [43] for its derivation and details.

ALGORITHM 2: GraphOpt

1. Choose some initial partition $\hat{\mathcal{A}}$ and graph $\hat{G}$;
2. Update $\hat{G}$: $\hat{G} = \arg\max_{G} \left(\log \mathrm{P}(\mathcal{C}|G) + \log \mathrm{P}(G|\hat{\mathcal{A}}) \right)$;
3. Update $\hat{\mathcal{A}}$: $\hat{\mathcal{A}} = \arg\max_{\mathcal{A}} \log \mathrm{P}(\hat{G}|\mathcal{A})$;
4. Iterate (2)-(3) until convergence.

The second group of algorithms is based on constructing a surrogate graph $\hat{G}$ and then clustering this graph (using, e.g., the Louvain community detection algorithm). It is crucial that $\hat{G}$ does *not* need to be similar to G, it just needs to capture the community structure on an aggregated level. For instance, in the algorithm Path, we connect all consecutive nodes that participated in one cascade. In the algorithm Clique, we connect all pairs of nodes that participated in a cascade by weighted edges (weights depend on the time difference between the infection times).

We conduct extensive experiments on various real-world networks to compare the proposed approaches with the existing baselines. Surprisingly, we conclude that the most stable performance is obtained by our proposed surrogate-graph-based heuristics that are agnostic to a partic-

ular graph structure and epidemic model. These heuristics work equally well on different networks and for epidemics of different types.

Analysis of cluster similarity indices

The results of this section are based on the paper [15].

When developing and analyzing community detection algorithms, it is crucial to be able to validate the results, i.e., measure the performance of different algorithms and compare the results. We demonstrate that this problem is crucial: dozens of cluster similarity measures exist, they often disagree with each other, these disagreements do affect which algorithms are preferred in applications, and this can lead to degraded performance in real-world systems.

We propose a theoretical framework to tackle this problem: we develop a list of desirable properties and conduct an extensive theoretical analysis to verify which indices satisfy them. This allows for making an informed choice: given a particular application, one can first select properties that are desirable for the task and then identify indices satisfying these. Our approach leads to recommendations that considerably differ from how validation indices are currently being chosen by practitioners. Some of the most popular indices are even shown to be dominated by previously overlooked ones.

Let us briefly discuss the properties that are fully described in [15]. By $V(A, B)$ we denote a validation (similarity) index applied to two partitions A and B of a given set of elements.

1. First, the numerical value that an index assigns to a similarity must be easily interpretable. In particular, it should be easy to see whether the candidate clustering is maximally similar to (i.e., coincides with) the reference clustering. Formally, we require that $V(A, A) = c_{\max}$ is constant and is a strict upper bound for $V(A, B)$ for all $A \neq B$. This property is called *maximal agreement*.

2. Similarity is intuitively understood as a symmetric concept. Therefore, a good similarity index is often expected to be *symmetric*, i.e., $V(A, B) = V(B, A)$ for all partitions A, B.

3. Running time is crucial for clustering tasks on large datasets, and algorithms and validation indices with superlinear time can be infeasible. So, we say that an index has *linear complexity* when its worst-case running time is linear in the number of elements.

4. For some applications, a distance interpretation of dissimilarity may be desirable: whenever A is similar to B, and B is similar to C, then A should also be somewhat similar to C. Thus, we say that an index satisfies the *distance* property if it is linearly transformable to a distance metric.

5. When one clustering is changed to resemble the other clustering more, the similarity score ought to improve. Hence, we require an index to be *monotone* w.r.t. changes that increase the similarity. This property is fully formalized in [15].

6. Finally, the *constant baseline* property is arguably the most significant: it is less intuitive than the other ones and may lead to unexpected consequences in practice. Informally, a good similarity index should not prefer a candidate clustering B over another clustering C just because B has many or few clusters. This intuition can be formalized using random partitions: assume that we have a reference clustering A and two random partitions B and C. Intuitively, both random guesses are equally bad approximations of A. Therefore, we require the similarity value of a random candidate w.r.t. the reference partition to have a fixed expected value c_{base} (independent of A or the sizes of B). However, this does require a careful formalization of random candidates, which is done in [15].

Our main results are summarized in Tables 1 and 2, and we refer to [15] for formal definitions of the indices and all the properties.

Other applications

In this part, we discuss other applications of graph analysis. First, we cover publication date estimation, where the proposed algorithm is

Table 1: Requirements for general similarity indices

	Max. agreement	Symmetry	Distance	Lin. complexity	Monotonicity	Const. baseline
NMI	✓	✓	✗	✓	✓	✗
NMI_{max}	✓	✓	✓	✓	✗	✗
FNMI	✓	✗	✗	✓	✗	✗
VI	✓	✓	✓	✓	✓	✗
SMI	✗	✓	✗	✗	✗	✓
FMeasure	✓	✓	✗	✓	✗	✗
BCubed	✓	✓	✗	✓	✓	✗
AMI	✓	✓	✗	✗	✓	✓

Table 2: Requirements for pair-counting indices

	Max. agreement	Min. agreement	Symmetry	Distance	Lin. complexity	Monotonicity	Strong monotonicity	Const. baseline	As. const. baseline	Type of bias
R	✓	✓	✓	✓	✓	✓	✓	✗	✗	⤭
AR	✓	✗	✓	✗	✓	✓	✗	✓	✓	
J	✓	✗	✓	✓	✓	✓	✗	✗	✗	↘
W	✗	✗	✗	✗	✓	✗	✗	✗	✗	↘
D	✓	✗	✓	✗	✓	✓	✗	✗	✗	↘
CC	✓	✓	✓	✗	✓	✓	✓	✓	✓	
S&S	✓	✓	✓	✗	✓	✓	✓	✓	✓	
CD	✓	✓	✓	✓	✓	✓	✓	✗	✓	

motivated by the recency-based model discussed above. Then, we discuss the problem of detecting high-degree vertices in large complex networks. Finally, we address the problem of nearest neighbor search and discuss graph-based algorithms for this problem.

Publication date prediction

We start with an example of how the *recency-based model* introduced above can be used to improve the quality of publication date prediction. The results of this section are based on the paper [39].

The task is to detect a document's *publication date*. Knowing web page publication dates is essential, for instance, for computing features for recency-sensitive ranking of web documents. Unfortunately, the publication dates of a large share of web pages cannot be reliably determined. The most common way to determine the publication date of a web page is *content-based*, i.e., to find this date in the HTML body of this page. However, pages may contain no or several candidate dates to choose from; these dates can be written in different formats and for different time zones, etc. In some other cases, a web page's publication date can be considered equal to the date of the first crawl of that page. However, due to resource

constraints, not all websites are re-crawled frequently enough to make it possible to detect new pages immediately after their publication.

The proposed algorithm combines content-based methods of date extraction with *link-based methods*. The first stage of the algorithm is to extract candidate dates from the URL and the HTML body of each page and choose the most probable publication date from among the candidates. For some pages, it is possible to detect highly reliable *anchor dates*, which will be fixed for the rest of the algorithm. For some other pages, candidate dates can also be extracted, but they are less reliable, and their estimates can be improved at the third stage; such dates are called *seed dates*. For the rest of the pages, content-based date extraction is simply impossible.

At the second stage of our algorithm, we choose approximate dates for all pages without the seed or anchor dates. For this, we use and compare several *date propagation* methods, where we iteratively propagate known dates using, for instance, averaging over the neighbors.

The obtained *initial dates* can further be improved at the third stage by our *likelihood optimization* method based on the model [24]. In [24], the publication dates of web pages are used to predict the evolution of the Web link structure. Here we do the reverse operation, i.e., we use the recency-based model to estimate the publication dates of web pages. Given only the currently observed link structure, we apply "reverse engineering" to reveal the whole process of the Web's evolution. Namely, we find such publication dates which maximize the probability that the web graph observed in reality is produced by this model.

The particular model we assume has several parameters. For each page p, we have the number of outgoing links m_p, its intrinsic quality q_p, and the publication time t_p. Besides these page-specific parameters, we also have the rate of attractiveness decay λ, an auxiliary constant c ($c > \lambda$), and the number of pages n. At the beginning, we have n pages and no links between them. Each page p has its publication time t_p. Then, for each page p, we generate m_p outgoing links. All links are modeled as mutually independent random variables that determine their target pages. The probability of a page r to be chosen as a target page for a link from a page p is proportional to the relative attractiveness of r according to p, which is a function of q_r

(intrinsic quality of r) and the age difference $a_{p,r}$ for the pages p and r, that is, $t_p - t_r$. Note that in the model the difference $a_{p,r}$ can be negative, i.e., there is a possibility of an edge between p and r with $t_p < t_r$. In a real web graph such a link can be added at a moment $t > t_r$ if p was updated at t. The attractiveness function is defined as follows:

$$\mathrm{attr}(q_r, a_{p,r}) = \begin{cases} q_r \cdot e^{-\lambda a_{p,r}} \cdot \left(1 - \frac{e^{-ca_{p,r}}}{2}\right) & \text{if } a_{p,r} \geqslant 0\,, \\ q_r \cdot e^{-\lambda a_{p,r}} \cdot \frac{e^{ca_{p,r}}}{2} & \text{if } a_{p,r} < 0\,. \end{cases} \tag{8}$$

First of all, the attractiveness of r is proportional to its quality. Second, the attractiveness decreases with the age of r, i.e., older pages are less popular. These two multipliers are proposed and motivated in [24]. The third multiplier is the sigmoid function that replaces the indicator $\mathbb{1}_{a_{p,r} \geqslant 0}$ from [24]. This is done to make the probabilities of edges differentiable. The sigmoid also allows us to avoid degenerate likelihood since all probabilities become greater than zero.

We use this model to estimate the publication dates of web documents. Namely, we are given an oriented graph (nodes are the web documents and edges are the links between them), and we assume that this graph is constructed according to the procedure described above. We are given the observed values of some parameters (the numbers of outgoing links m_p and some anchor publication dates t_p). We want to find the rest of the unknown values to maximize the probability that the observed graph is constructed under the described model. The parameters with unknown values are the rate of attractiveness decay λ, the constant c, the qualities of all pages q_p, the publication times for non-anchor pages t_p.

We optimize the unknown parameters using gradient descent. The formulas for the likelihood and its derivatives can be found in [39].

In the paper, we evaluate the proposed algorithm on two datasets: the web crawled dataset obtained by Yandex (4M pages from 70 hosts) and the MemeTracker public dataset consisting of blog posts and news articles (12M pages from 250K hosts).

Detection of high-degree nodes in large networks

The results of this section are based on the paper [5].

We address the problem of quick detection of high-degree entities (nodes) in large online social networks. The entities can be, for example, users, interest groups, user categories, geographical locations, etc. For instance, one can be interested in finding a list of Twitter users with many followers or Facebook interest groups with many members. The practical importance of this problem is attested by many companies that continuously collect and update statistics about popular entities, usually using the degree of an entity as an approximation of its popularity.

With the full search, one can find top-k in-degree nodes in a directed graph G of size N with $\mathrm{O}(N)$ complexity. For very large networks, even such linear complexity is a too high cost to pay. Furthermore, the data of social networks is typically available only to managers of social networks and can be obtained by other interested parties only through API (Application Programming Interface) requests, while the rate of allowed API requests is usually very limited.

Formally, let V be a set of N entities, typically users, that can be accessed using API requests. Let W be another set of M entities (possibly equal to V). We consider a bipartite graph (V, W, E), where a directed edge $(v, w) \in E$, with $v \in V$, and $w \in W$, represents a relation between v and w. For instance, for the Twitter graph, V is a set of Twitter users, $W = V$, and $(v, w) \in E$ means that v follows w or that v retweeted a tweet of w. Note that any directed graph $G = (V, E)$ can be represented equivalently by the bipartite graph (V, V, E). One can also suppose that V is a set of users and W is a set of interest groups, while the edge (v, w) represents that the user v belongs to the group w. Our goal is to quickly find the top in-degree entities in W.

Let n be the allowed number of requests to API. Our algorithm consists of two steps. We spend n_1 API requests on the first step and n_2 API requests on the second step, with $n_1 + n_2 = n$.

First step We start by sampling uniformly at random a set A of n_1 nodes $v_1, \ldots, v_{n_1} \in V$, the nodes are sampled independently. For each node in A, we record its out-neighbors in W. In practice, we bound the number of recorded out-links by the maximal number of IDs that can be retrieved within one API request. Thus the first stage uses exactly n_1 API requests.

For each $w \in W$, we identify $S[w]$, which is the number of nodes in A that have a (recorded) edge to w.

Second step We use n_2 API requests to retrieve the actual in-degrees of the n_2 nodes with the highest values of $S[w]$. The idea is that the nodes with the largest in-degrees in W are likely to be among the n_2 nodes with the largest $S[w]$. For example, if we are interested in the top-k in-degree nodes in a directed graph, we hope to identify these nodes with high precision if k is significantly smaller than n_2.

In [5] we empirically demonstrate that our algorithm outperforms other known methods by a large margin. For instance, we need only one thousand API requests to find the top-100 most followed users, with more than 90% precision, in the online social network Twitter with approximately a billion registered users. An important contribution of this work is the analysis of the proposed algorithm using Extreme Value Theory — a branch of probability that studies extreme events and properties of largest order statistics in random samples. Using this theory, we derive an accurate prediction for the algorithm's performance. We show that the number of API requests for finding the top-k most popular entities is sublinear in the number of entities. Moreover, we formally show that the high variability of the entities, expressed through heavy-tailed distributions, is the reason for the algorithm's efficiency.

Analysis of graph-based nearest neighbor search

The results of this section are based on the paper [41].

Let us discuss another important practical application of graph analysis. Many methods in machine learning, pattern recognition, coding theory, and other research areas are based on nearest neighbor search (NNS). In particular, the k-nearest neighbor method is included in the list of top 10 algorithms in data mining [46]. Since modern datasets are mostly massive (both in terms of the number of elements n and the dimension d), reducing the computation complexity of NNS algorithms is of the essence. The nearest neighbor problem is to preprocess a given dataset $\mathcal{D}$ so that for an arbitrary forthcoming query vector q, we can quickly (in time $o(n)$) find its nearest neighbors in $\mathcal{D}$.

Many efficient methods exist for the NNS problem. Recently, graph-based approaches were shown to demonstrate superior performance over other types of algorithms in many large-scale applications of NNS [4]. Most graph-based methods are based on constructing a nearest neighbor graph (or its approximation), where nodes correspond to the elements of $\mathcal{D}$, and each node is connected to its nearest neighbors by directed edges. Then, for a given query q, one first takes an element in $\mathcal{D}$ (either random or fixed predefined) and makes greedy steps towards q on the graph: at each step, all neighbors of a current node are evaluated, and the one closest to q is chosen. Various heuristics were proposed to speed up graph-based search [29].

While there is much evidence empirically showing the superiority of graph-based NNS algorithms in practical applications, there is very little theoretical research supporting this. We make a step in this direction. Our analysis assumes the uniform distribution on a sphere, and we mainly focus on the dense regime $d \ll \log n$.

Formally, assume that we are given a dataset $\mathcal{D} = \{\mathbf{x}_1, \dots \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^{d+1}$ and assume that all elements of $\mathcal{D}$ belong to a unit Euclidean sphere, $\mathcal{D} \subset \mathcal{S}^d$. This special case is of particular importance for practical applications since feature vectors are often normalized. For a given query $q \in \mathcal{S}^d$ let $\bar{\mathbf{x}} \in \mathcal{D}$ be its nearest neighbor. The aim of the exact NNS is to return $\bar{\mathbf{x}}$, while in c, R-ANN (approximate *near* neighbor), for given $R > 0$, $c > 1$, we need to find such $\mathbf{x}'$ that $\rho(q, \mathbf{x}') \leqslant cR$ if $\rho(q, \bar{\mathbf{x}}) \leqslant R$. By $\rho(\cdot, \cdot)$ we denote a spherical distance.

We assume that the elements $\mathbf{x}_i \in \mathcal{D}$ are i.i.d. random vectors uniformly distributed on $\mathcal{S}^d$. Random uniform datasets are considered to be the most natural "hard" distribution for the ANN problem. Hence, obtaining theoretical guarantees for such datasets is an important step towards understanding the limits and benefits of graph-based NNS algorithms.[1] We further assume that a query vector $q \in \mathcal{S}^d$ is placed uniformly within a distance R from the nearest neighbor $\bar{\mathbf{x}}$ (since c, R-ANN problem is defined conditionally on the event $\rho(q, \bar{\mathbf{x}}) \leqslant R$).

[1] From a practical point of view, real datasets are usually far from being uniformly distributed. However, in our experiments we show that *uniformization* and *densification* applied to a *general* dataset may improve some graph-based NNS algorithms [41].

We assume that the dimensionality $d = d(n)$ grows with n. We distinguish three fundamentally different regimes in NN problems: dense with $d \ll \log(n)$; sparse with $d \gg \log(n)$; moderate with $d = \Theta(\log(n))$.

Assuming that we have constructed a graph G on the elements of $\mathcal{D}$ and we are given a query q, we sample a random element $\mathbf{x} \in \mathcal{D}$ and perform a graph-based greedy descent: at each step, we measure the distances between the neighbors of a current node and q and move to the closest neighbor, while we make progress.

Plain NN graphs

We first investigate how this greedy search over NN graphs works in dense and sparse regimes. In the dense regime, when $d \ll \log(n)$, we take any $M > 1$ and let $G(M)$ be a graph obtained by connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ iff $\rho(\mathbf{x}_i, \mathbf{x}_j) \leqslant \arcsin\left(M\, n^{-1/d}\right)$. We prove the following theorem.

Theorem 17. *Assume that* $\log\log n \ll d \ll \log n$ *and we are given some constant* $c \geqslant 1$. *Let* M *be a constant such that* $M > \sqrt{\frac{4c^2}{3c^2-1}}$, *then, with probability* $1 - o(1)$, $G(M)$*-based NNS solves* c, R*-ANN for any* R *(or the exact NN problem if* $c = 1$*); time complexity is* $\Theta\left(d^{1/2} \cdot n^{1/d} \cdot M^d\right) = n^{o(1)}$; *space complexity is* $\Theta\left(n \cdot d^{-1/2} \cdot M^d \cdot \log n\right) = n^{1+o(1)}$.

In other words, for the dense regime, the main term of the time complexity is $n^{1/d} \cdot M^d$ for some constant M. Here M^d corresponds to the complexity of one step and $n^{1/d}$ to the number of steps.

In the sparse regime, when $d \gg \log(n)$, for any M, $0 < M < 1$, let $G(M)$ be a graph obtained by connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ iff $\rho(\mathbf{x}_i, \mathbf{x}_j) \leqslant \arccos\left(\sqrt{\frac{2M \ln n}{d}}\right)$. The following theorem holds.

Theorem 18. *For any* $c > 1$ *let* $\alpha_c = \cos\left(\frac{\pi}{2c}\right)$ *and let* M *be any constant such that* $M < \frac{\alpha_c^2}{\alpha_c^2+1}$. *Then, with probability* $1 - o(1)$, $G(M)$*-based NNS solves* c, R*-ANN (for any* R *and for spherical distance); time complexity of the procedure is* $\Theta\left(n^{1-M+o(1)}\right)$; *space complexity is* $\Theta\left(n^{2-M+o(1)}\right)$.

Interestingly, as follows from the proof, in the sparse regime, the greedy algorithm converges in at most two steps with probability $1 - o(1)$ (on a uniformly distributed dataset).

Effect of long-range edges

According to the discussion above, if $d \ll \sqrt{\log n}$ (very dense setting), the number of steps becomes the main term of time complexity. In this case, it is reasonable to reduce the number of steps by adding so-called long-range links (or shortcuts) — some edges connecting elements that are located far from each other — which may speed up the search on early stages of the algorithm.

Our approach to adding such long-range links is motivated by Kleingerg's analysis [19], where a 2-dimensional grid supplied with some random long-range edges is considered. Kleinberg assumed that in addition to the local edges, each node creates one random outgoing long link, and the probability of a link from u to v is proportional to $\rho(u, v)^{-r}$. He proved that for $r = 2$, the greedy graph-based search finds the target element in $O\left(\log^2 n\right)$ steps, while any other r gives at least n^{φ} steps with $\varphi > 0$. This result can be extended to *constant* $d > 2$: in this case, one should take $r = d$ to achieve polylogarithmic number of steps.

Following [19], we draw long-range edges with the following probabilities:

$$\mathrm{P}(\text{edge from } u \text{ to } v) = \frac{\rho(u, v)^{-d}}{\sum_{w \neq u} \rho(u, w)^{-d}}\,. \tag{9}$$

Theorem 19. *Under the conditions of Theorem 17, sampling* $\Theta(\log n)$ *independent long-range edges for each node according to* (9) *reduces the number of steps to* $O(\log n)$ *(with probability* $1 - o(1)$*).*

Importantly, in contrast to [19], we assume $d \to \infty$. Theorem 19 implies that long-range edges allow to guarantee $O\left(\log n\right)$ steps, while plain NN graphs give $\Theta\left(n^{1/d}\right)$. Hence, reducing the number of steps is reasonable if $\log n < n^{1/d}$, which means that $d < \frac{\log n}{\log \log n}$.

However, it is non-trivial how to apply Theorem 19 in practice due to the dependence of probabilities in (9) on d: real datasets usually have a low intrinsic dimension even when embedded to a higher-dimensional space [27], and the intrinsic dimension may vary over the dataset. Thus, it is hard to choose a proper value of d in (9).

However, as we discuss in [41] in more detail, one can make the distribution in (9) dimension-free. For this, we reformulate the probabilities in

terms of *ranks* instead of *distances*. Let us sort all the elements by their closeness to some element u. Then, we define the following probability of adding an edge from u to another element:

$$\mathrm{P}(\text{edge to } k\text{-th neighbor}) = \frac{1/k}{\sum_{i=1}^{n} 1/i} \sim \frac{1}{k \ln n}\,. \tag{10}$$

This distribution is *dimension independent* and for uniform d-dimensional datasets it gives the same guarantees as (9).

Effect of beam search

Beam search is a heuristic algorithm that explores a graph by expanding the most promising element in a limited set. It is widely used in graph-based NNS algorithms as it drastically improves the accuracy [29]. The following theorem shows that beam search provably reduces graph-based NNS complexity in our setting.

Theorem 20. *Let $M > 1$, $L > 1$ be such constants that $M^2\left(1 - \frac{M^2}{4L^2}\right) > 1$ and let $\log\log n \ll d \ll \log n$. Assume that we use beam search with $\frac{C\,L^d}{\sqrt{d}}$ candidates (for a sufficiently large C) and we add $\Theta(\log n)$ long-range edges. Then, $G(M)$-based NNS solves the exact NN problem with probability $1 - o(1)$. The time complexity is $O\left(L^d \cdot M^d\right)$.*

As a result, beam search allows us to significantly reduce degrees of a graph, which finally leads to time complexity reduction. To show that, we can take $M = \sqrt{\frac{3}{2}}$ and any $L > \sqrt{\frac{9}{8}}$. Then, the main term of the time complexity can be reduced to $\left(\frac{27}{16}\right)^{d/2}$, which is less than $2^{d/2}$ from Theorem 17.

Conclusion

This thesis is based on published papers [5, 13, 15, 20, 21, 24, 35, 36, 37, 38, 39, 40, 41, 42, 43].

In papers [13, 20, 21, 24, 35, 36, 37, 38, 40], we analyze the properties of existing models of complex networks and develop new realistic models with desirable quantitative and topological properties.

In papers [15, 42, 43], we study several aspects of community detection: the choice of a null model for likelihood-based community detection, community detection based on cascade data, and the problem of choosing a proper validation index for community detection algorithms.

In papers [5, 25, 41], we discuss other applications of graph analysis: likelihood-based publication date estimation, detecting high-degree vertices in large complex networks, and theoretical analysis of graph-based algorithms for nearest neighbor search.

The main results submitted for defense are the following:

- A new class of *Generalized Preferential Attachment* models and theoretical results obtained for all models in this class (degree distribution, local and global clustering coefficients, and degree correlations).
- A general statement that the global clustering coefficient tends to zero with size for all graphs with a power-law degree distribution with an infinite variance (assuming that the degrees are sampled independently).
- Theoretical analysis of modularity for d-regular, preferential attachment, and spatial preferential attachment graphs.
- A novel principle called *preferential placement* that allows for generating structures with a power-law distribution of cluster sizes; the analysis of the obtained structures.
- A new model called *recency-based preferential attachment* and the analysis of its properties.
- Analysis of likelihood-based community detection algorithms based on different null models; new ILFR model for this task.
- Systematic analysis of community detection based on information propagation; new effective methods for this problem.
- Theoretical analysis of performance measures for community detection algorithms.

- A new algorithm for dating web pages based on likelihood optimization under the recency-based model.
- A new algorithm for quick detection of high-degree nodes in complex networks.
- Theoretical guarantees for graph-based nearest neighbor search algorithms.

Acknowledgments

The author is very grateful to Andrei Mikhailovich Raigorodskii for scientific consultancy during the research process.

Bibliography

[1] W. Aiello, A. Bonato, C. Cooper, J. Janssen, and P. Prałat. A spatial web graph model with local influence regions. *Internet Mathematics*, 5(1-2):175–196, 2008.

[2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[3] A. Arenas, L. Danon, A. Diaz-Guilera, P. M. Gleiser, and R. Guimera. Community analysis in social networks. *The European Physical Journal B*, 38(2):373–380, 2004.

[4] M. Aumüller, E. Bernhardsson, and A. Faithfull. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems*, 87:101374, 2020.

[5] K. Avrachenkov, N. Litvak, L. Ostroumova Prokhorenkova, and E. Suyargulova. Quick detection of high-degree entities in large directed networks. In *2014 IEEE International Conference on Data Mining*, pages 20–29. IEEE, 2014.

[6] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[7] P. J. Bickel and A. Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

[8] M. Bloznelis and V. Kurauskas. Clustering coefficient of random intersection graphs with infinite degree variance. *Internet Mathematics*, 2016.

[9] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

[10] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. In *The Structure and Dynamics of Networks*, pages 384–395. Princeton University Press, 2011.

[11] P. G. Buckley and D. Osthus. Popularity based random graph models leading to a scale-free degree sequence. *Discrete Mathematics*, 282(1-3):53–68, 2004.

[12] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[13] A. Dorodnykh, L. Ostroumova Prokhorenkova, and E. Samosvat. Preferential placement for community structure formation. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 75–89. Springer, 2017.

[14] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[15] M. M. Gösgens, A. Tikhonov, and L. Prokhorenkova. Systematic analysis of cluster similarity indices: How to validate validation measures. In *International Conference on Machine Learning*, pages 3799–3808. PMLR, 2021.

[16] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical review E*, 68(6):065103, 2003.

[17] P. Holme and B. J. Kim. Growing scale-free networks with tunable clustering. *Physical review E*, 65(2):026107, 2002.

[18] B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

[19] J. Kleinberg. The small-world phenomenon: an algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170, 2000.

[20] A. Krot and L. Ostroumova Prokhorenkova. Assortativity in generalized preferential attachment models. *Internet Mathematics*, 2017.

[21] A. Krot and L. Ostroumova Prokhorenkova. Local clustering coefficient in generalized preferential attachment models. *Internet Mathematics*, 2017.

[22] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 57–65. IEEE, 2000.

[23] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[24] D. Lefortier, L. Ostroumova, and E. Samosvat. Evolution of the media web. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 80–92. Springer, 2013.

[25] D. Lefortier, L. Ostroumova, E. Samosvat, and P. Serdyukov. Timely crawling of high-quality ephemeral new content. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 745–750, 2013.

[26] J. Leskovec. *Dynamics of large networks*. PhD thesis, Carnegie Mellon University, School of Computer Science, 2008.

[27] P.-C. Lin and W.-L. Zhao. Graph based nearest neighbor search: Promises and failures. *arXiv preprint arXiv:1904.02077*, 2019.

[28] N. Litvak and R. Van Der Hofstad. Uncovering disassortativity in large scale-free networks. *Physical Review E*, 87(2):022801, 2013.

[29] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world

graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[30] F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics reports*, 533(4):95–142, 2013.

[31] M. Newman. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *arXiv preprint arXiv:1606.02319*, 2016.

[32] M. E. Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.

[33] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

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

[35] L. Ostroumova, A. Ryabchenko, and E. Samosvat. Generalized preferential attachment: tunable power-law degree distribution and clustering coefficient. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 185–202. Springer, 2013.

[36] L. Ostroumova Prokhorenkova. Global clustering coefficient in scale-free weighted and unweighted networks. *Internet Mathematics*, 12(1-2):54–67, 2016.

[37] L. Ostroumova Prokhorenkova. General results on preferential attachment and clustering coefficient. *Optimization Letters*, 11(2):279–298, 2017.

[38] L. Ostroumova Prokhorenkova, P. Prałat, and A. Raigorodskii. Modularity of complex networks models. *Internet Mathematics*, 2017.

[39] L. Ostroumova Prokhorenkova, P. Prokhorenkov, E. Samosvat, and P. Serdyukov. Publication date prediction through reverse engineering of the web. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 123–132, 2016.

[40] L. Ostroumova Prokhorenkova and E. Samosvat. Recency-based preferential attachment models. *Journal of Complex Networks*, 4(4):475–499, 2016.

[41] L. Prokhorenkova and A. Shekhovtsov. Graph-based nearest neighbor search: From practice to theory. In *International Conference on Machine Learning*, pages 7803–7813. PMLR, 2020.

[42] L. Prokhorenkova and A. Tikhonov. Community detection through likelihood optimization: in search of a sound model. In *The World Wide Web Conference*, pages 1498–1508, 2019.

[43] L. Prokhorenkova, A. Tikhonov, and N. Litvak. Learning clusters through information diffusion. In *The World Wide Web Conference*, pages 3151–3157, 2019.

[44] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697*, 2011.

[45] A. Sheikhahmadi, M. A. Nematbakhsh, and A. Shokrollahi. Improving detection of influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, 436:833–845, 2015.

[46] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[47] Z.-K. Zhang, C. Liu, X.-X. Zhan, X. Lu, C.-X. Zhang, and Y.-C. Zhang. Dynamics of information diffusion and its applications on complex networks. *Physics Reports*, 651:1–34, 2016.

[48] T. Zhou, G. Yan, and B.-H. Wang. Maximal planar networks with large clustering coefficient and power-law degree distribution. *Physical Review E*, 71(4):046141, 2005.