Федеральное государственное бюджетное учреждение науки
Институт системного программирования им. В. П. Иванникова РАН

На правах рукописи

Дробышевский Михаил Дмитриевич

Методы и программные средства моделирования и генерации сложных сетей с сохранением графовых свойств

Специальность 05.13.11 —
«математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
кандидат физ.-мат. наук
Турдаков Денис Юрьевич

Москва — 2019

Оглавление

Стр.

Введение 4

Глава 1. Обзор подходов к моделированию случайных графов . 10
1.1 Основные понятия и определения 10
1.1.1 Особенности терминологии 10
1.1.2 Определения случайного графа 11
1.1.3 Известные графовые характеристики и признаки 13
1.2 Анализ релевантной литературы 22
1.2.1 Процедура сбора публикаций 22
1.2.2 Обзор обзоров 24
1.2.3 Существующие классификации моделей случайных графов 30
1.3 Таксономия подходов к моделированию случайных графов 34
1.3.1 Класс генеративных подходов 35
1.3.2 Класс управляемых признаками подходов 47
1.3.3 Класс предметно-специфичных подходов 55
1.4 Обсуждение 59
1.4.1 Комментарии к таксономии 59
1.4.2 Приложения моделей случайных графов 63
1.5 Генераторы графов, похожих на данный 70
1.5.1 Случайные графы скалярного произведения 72
1.6 Выводы к первой главе 73

Глава 2. Генерация графов, похожих на данный. Подход на основе вложения графа и алгоритм ERGG-dwc 75
2.1 Описание и постановка задачи 75
2.2 Подход на основе вложения графа — ERGG 78
2.3 Краткий обзор методов вложения направленных графов 79
2.3.1 Анализ и выводы 84
2.4 Метод генерации случайных графов на основе вложения графа — ERGG-dwc 84
2.4.1 Вложение + восстановление 86

Стр.

2.4.2 Аппроксимация распределения + сэмплирование 89
2.4.3 Атрибуты: метки сообществ и веса ребер 92
2.4.4 Вычислительная сложность алгоритма ERGG-dwc 94
2.5 Выводы ко второй главе 95

Глава 3. Программная реализация алгоритма ERGG-dwc и экспериментальные исследования 97
3.1 Описание программной системы 97
3.2 Экспериментальное исследование ERGG-dwc 99
3.2.1 Параметры метода вложения 100
3.2.2 Метод аппроксимации распределения 101
3.2.3 Корректность присвоения атрибутов 105
3.2.4 Производительность 106
3.3 Экспериментальное сравнение ERGG-dwc с другими методами . . 107
3.3.1 Методология 107
3.3.2 Измерение похожести 110
3.3.3 Измерение вариабельности 114
3.4 Пример использования: тестирование качества работы алгоритмов 121
3.5 Выводы к третьей главе 122

Заключение 125

Благодарности 126

Список сокращений и условных обозначений 127

Список литературы 129

Приложение А. Эксперименты в процессе разработки ERGG-dwc 147
А.1 Метод аппроксимации распределения 147
А.2 Атрибуты 148

Приложение Б. Экспериментальное сравнение ERGG-dwc с другими методами 149
Б.1 Измерение похожести 149
Б.2 Измерение вариабельности 165

Введение

Актуальность темы.

В реальном мире многие данные имеют графовую структуру: набор дискретных объектов, некоторые пары которых связаны между собой. Примеры охватывают разные сферы исследований: социальные сети (Фейсбук, Твиттер), биологические сети (метаболические и пищевые цепочки), графы цитирований, автономные системы (Интернет, граф взаимодействия компонентов программ) и т. д. Часто такие сети обладают нетривиальными топологическими свойствами и известны как *сложные сети* [1].

В рамках исследований сложных сетей возникает ряд вопросов: насколько надежна сеть Интернет? Как устроены общественные отношения, отраженные в социальных сетях? Какие законы управляют распространением болезней и информационными потоками и как ими можно управлять? Для поиска ответов активно разрабатываются математические модели сложных сетей, известные также как *случайные графы*[1]. Главным аспектом моделей случайных графов является точное отражение свойств, присущих реальным сетям, в том числе для адекватного предсказания их поведения в будущем. Первой моделью случайного графа принято считать модель Эрдеша-Реньи, предложенную в 1959 году, в которой каждая пара узлов независимо с заданной вероятностью соединяется ребром [2]. Позднее было обнаружено свойство безмасштабности во многих реальных сетях и предложены различные модели, его объясняющие, например, модель Барабаши-Альберт в 1999 году [3]. Последние десятилетия наука о сложных сетях стала активно развиваться, свой вклад в область внесли зарубежные и российские исследователи, в том числе Райгородский А. М., Берновский М. М., Кузюрин Н. Н. и другие.

Некоторые сети содержат приватную информацию в своих связях (например, Фейсбук), и их непосредственная публикация нарушает политику конфиденциальности данных, что затрудняет получение таких данных в исследовательских целях. Поэтому возникает задача анонимизации графа, то есть создания похожего графа, сохраняющего важные свойства оригинала, но достаточно от него отличающегося, чтобы обеспечить конфиденциальность.

[1] В настоящей работе термином *граф* называется математическая модель, термин *сеть* преимущественно используется, когда речь идет об объекте реального мира.

Известно, например, что простая перенумерация идентификаторов вершин социальной сети не предотвращает возможность выяснить существование связи между двумя пользователями [4].

Другое приложение моделей случайных графов состоит в создании искусственных данных для тестирования алгоритмов анализа сетей. Многие сети существуют в единственном экземпляре, при этом для проверки статистической значимости работы алгоритмов необходима выборка графов с похожими свойствами, при этом обеспечивающих некоторый разброс в свойствах графов. Таким образом, имеется потребность в моделях для генерации случайных графов, обеспечивающих баланс между похожестью и случайностью в смысле свойств графов. Кроме того, для тестирования масштабируемости алгоритмов, дополнительно необходимы выборки похожих графов с возможностью контроля их размера.

Общепринятого критерия похожести графов не существует, на практике используется ряд известных характеристик графов, по которым оценивается близость графов. В данной работе используются следующие характеристики:

- числовые: средняя степень вершины, взаимность ребер, ассортативность степеней, средний коэффициент кластеризации, эффективный диаметр гигантской компоненты, спектральный радиус;
- распределения: распределение степеней, кумулятивный коэффициент кластеризации, коэффициент кластеризации от степени вершины, распределение подграфов размера 3, достижимость вершин.

Под *похожестью* графов в данной работе понимается близость их числовых характеристик, а также некоторых распределений, для которых определена мера их сравнения, например, косинусная близость векторов-распределений подграфов в графе. *Вариабельностью* множества графов будем называть дисперсию их числовых характеристикв этом множестве.

Традиционно, разработка и использование модели случайных графов происходит по следующей схеме:

1. Извлечение статистических признаков и закономерностей из реальных данных.
2. Выбор признаков для моделирования графов, например, распределение степеней вершин, коэффициент кластеризации, диаметр.
3. Определение модели как вероятностного пространства возможных графов, обычно путем задания генеративной процедуры.

4. Сэмплирование случайных графов из заданного вероятностного пространства.

Главный недостаток такого подхода заключается в том, что разные графовые домены (социальный, биологический, автономные системы и т. п.) имеют свойства, которые могут быть неизвестны. Как следствие, модель, созданная для одного домена, может не подходить для других. Кроме того, нельзя заранее сказать, учитывает ли модель все существенные свойства реальной сети. В связи с этим, актуальным направлением является разработка алгоритмов, подходящих для произвольного графового домена, например, *обучение* на данном графе, то есть, автоматическое извлечение его признаков.

Направленность ребер графа является неотъемлемой особенностью для многих доменов, например, графов мобильных звонков и графов цитирований. Кроме того, степень связи между вершинами часто выражается *весом* ребра (продолжительность или количество звонков, количество цитирований). Во многих сетях вершины образуют структуру *сообществ*, соответствующую высокоуровневой организации узлов сети в более тесно связанные группы на основании общих интересов, ролей, функций. На момент написания работы не было обнаружено существующего метода генерации случайных графов контролируемого размера, способного автоматически обучаться на данном графе и учитывающего три перечисленные особенности графов: направленные ребра, взвешенные ребра, структура сообществ.

Целью данной работы является разработка метода и программного средства генерации случайных графов, похожих на данный, соответствующих следующим требованиям:

- автоматическое обучение на заданном графе;
- возможность генерировать графы контролируемого размера;
- одновременная поддержка трех особенностей графа: направленные ребра, взвешенные ребра и структура сообществ;
- похожесть генерируемых графов на исходный: отклонение по каждой характеристике не выше соответствующих отклонений у других современных методов;
- вариабельность генерируемых графов: разброс значений числовых характеристик близок к таковому у реальных графов из одного домена.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. На основе анализа существующих моделей случайных графов разработать и реализовать метод генерации случайных графов, удовлетворяющий указанным требованиям;
2. Провести экспериментальное исследование разработанного метода на соответствие требованиям, сравнение его с другими методами.

Основные положения, выносимые на защиту:

1. Предложен новый подход ERGG к генерации случайных направленных графов, похожих на данный, основанный на вложении графа в пространство размерности, много меньшей числа его вершин.
2. В рамках подхода ERGG предложен метод ERGG-dwc, решающий задачу генерации графов, похожих на данный и удовлетворяющих требованиям: автоматическое обучение на заданном графе, контролируемый размер генерируемых графов, поддержка направленных ребер, взвешенных ребер и структуры сообществ.
3. Создана программная система, в которой реализован прототип ERGG-dwc и проведено его экспериментальное сравнение с другими современными методами. Показано, что ERGG-dwc не уступает другим методам в похожести генерируемых графов, но превосходит их по вариабельности.

Научная новизна.

В данной работе впервые предложен подход к генерации случайных графов, основанный на вложении графа в пространство с размерностью, много меньшей числа вершин графа.

В рамках подхода разработан метод ERGG-dwc генерации случайных графов, способный автоматически обучаться на заданном графе и генерировать похожие графы произвольного размера. Экспериментально показано, что, в отличие от существующих методов, ERGG-dwc позволяет генерировать графы, похожие на исходный, и одновременно обладающие вариабельностью, близкой к реальной вариабельности графов из одного домена.

Новизной разработанного метода ERGG-dwc также является одновременное удовлетворение всех требований: автоматическое обучение на исходном графе, генерация графов контролируемого размера, поддержка направленных, взвешенных графов со структурой сообществ.

Теоретическая и практическая значимость.

Теоретическая значимость работы заключается в том, что впервые была проде-

монстрирована возможность генерации случайных графов на основе вложения графа. Было показано, что в рамках данного подхода можно генерировать случайные графы, похожие на данный. При этом метод ERGG-dwc в терминах похожести генерируемых графов не уступает другим современным подходам, а в вариабельности превосходит их. Доказаны теоремы о вычислительной сложности и масштабируемости разработанного алгоритма ERGG-dwc.

С практической точки зрения метод ERGG-dwc может применяться для решения задач:

- создание искусственных коллекций графовых данных в целях тестирования алгоритмов анализа сетей;
- анонимизация графовых данных, при которой необходимо сгенерировать граф, похожий по свойствам на исходный, но отличающийся достаточно для сохранения анонимности.

Апробация работы.

Основные результаты работы докладывались на следующих конференциях и семинарах:

- Открытая конференция ИСП РАН им. В.П. Иванникова 2016 (1–2 декабря 2016 года, Москва);
- Европейская конференция по машинному обучению и принципам и практике извлечения знаний из баз знаний ECML PKDD 2017 (18–22 сентября 2017 года, Скопье, Македония);
- Открытая конференция ИСП РАН им. В.П. Иванникова 2017 (30 ноября – 1 декабря 2017 года, Москва);
- Научные семинары «Управление данными и информационные системы» Института системного программирования РАН им. В.П. Иванникова (2017–2018 годы, Москва).

Личный вклад.

Все выносимые на защиту результаты получены лично автором.

Публикации.

Основные результаты по теме диссертации изложены в трех работах, опубликованных в изданиях, рекомендованных ВАК, и одном патенте. В статьях [5; 7] вместе с соавторами была поставлена задача и проводилась редакторская правка, остальная часть выполнена автором. В работе [6] автору принадлежит основная часть: разделы 2 – 5; эта работа получила награду “best student paper award” на конференции ECML PKDD 2017.

На основе разработанного метода ERGG-dwc получен патент (на изобретение):

– Патент РФ 2018/151619. “Network analysis tool testing”. Заявлен 20.02.17; получен 23.08.18.

Вклад автора в патент состоит в разработке концепции и ее реализации.

Объем и структура работы. Диссертация состоит из введения, трёх глав, заключения и двух приложений. Полный объём диссертации составляет 165 страниц, включая 40 рисунков и 9 таблиц. Список литературы содержит 180 наименований.

Глава 1. Обзор подходов к моделированию случайных графов

В данной главе дается обзор существующих подходов к генерации случайных графов. Вначале обсуждаются основные понятия и приводятся известные графовые свойства и метрики (раздел 1.1). Затем дается краткая сводка по существующим в литературе обзорам в области моделирования случайных графов и существующих классификаций подходов (раздел 1.2). Далее представлена таксономия, объединяющая найденные в литературе подходы, с подробными примерами конкретных методов и алгоритмов (раздел 1.3). Затем следует обсуждение предложенной таксономии, анализ рассмотренных идей в контексте основных приложений случайных графов (раздел 1.4) и выводы (раздел 1.6).

1.1 Основные понятия и определения

1.1.1 Особенности терминологии

Как и во многих областях знаний, в литературе, касающейся моделей случайных графов, часто одним и тем же понятиям соответствуют разные термины. Также может возникнуть неоднозначное соответствие русских терминов англоязычным. Во избежание неясностей, далее в работе будем придерживаться следующих терминологических тождеств:

– “граф” = “сеть” = “***graph***” = “***network***” = “***network graph***”
– “вершина” = “узел” = “***node*** = “***vertex***”
– “ребро” = “***edge***” = “***link***” = “***arc***”
– “модель графа” = “модель случайного графа” = “***random graph model***” = “***graph model***” = “***network model***”
– “графовый признак” = “***graph feature***” = “***graph pattern***”
– “графовая характеристика” = “***graph metric***” = “***graph measure***”

В то же время, разницу между некоторыми понятиями необходимо прояснить явно.

Модель случайного графа и генератор случайного графа *Модель* случайного графа является моделью в математическом смысле. Она описывает или задает статистический объект. *Генератор* случайного графа это алгоритм, результатом исполнения которого является (случайный) граф. Обычно генератор является реализацией некоторой модели. Иногда наоборот, сначала предлагается описание процедуры генерации, которая неявным образом задает соответствующую модель.

В данной работе по умолчанию используется термин “модель”, обобщая оба варианта.

Графовая метрика, графовый признак, характеристика графа В англоязычной литературе используют ***graph metric*** и ***graph measure*** для обозначения некоторой функции, определенной на графе, например, диаметр графа, распределение степеней вершин, спектр матрицы смежности. Слово “metric” обычно встречается в контексте оценки качества и имеет смысл меры качества модели. Однако использовать термины метрика или мера здесь может быть некорректно, поскольку указанные характеристики графа не удовлетворяют соответствующим математическим определениям. Поэтому в данной работе предпочтение отдается термину *характеристика* графа.

Термины *признак* (***feature***) и *паттерн* (***pattern***) чаще используются для выделения отличительных атрибутов отдельного графа, как качественного характера (“распределение степеней вершин имеет тяжелый хвост”, “высокий коэффициент кластеризации”) так и количественного (сама последовательность степеней вершин, значение диаметра). Этому понятию также соответствует слово “свойство”, обозначающее некоторую измеримую характерную особенность конкретного объекта.

1.1.2 Определения случайного графа

Что именно называют *случайным графом*? В большинстве случаев подразумевается модель Эрдеша-Реньи (Erdős-Rényi, ER), а точнее, одна из двух очень похожих моделей: $G(n, m)$ [2], введенная П. Эрдешем и А. Реньи, и $G(n, p)$, предложенная Э. Гильбертом (E. Gilbert) [8], обе в 1959 году. $G(n, m)$

дает одинаковую вероятность всем графам на n вершинах с m ребрами, а в $G(n,p)$ каждое ребро графа на n вершинах появляется независимо с заданной вероятностью p. Эти две модели наиболее популярны в приложениях и до сих пор активно изучаются.

Фактически, в литературе можно встретить различные понятия, стоящие за термином "случайный граф". Следующие четыре цитаты это иллюстрируют[1]:

[9] "Говорят, что сеть является случайной, когда вероятность существования ребра между двумя вершинами совершенно не зависит от атрибутов вершин. Другими словами, единственной релевантной функцией является распределение степеней $P(k)$."

[10] "Во всей общности, под случайным графом на фиксированном наборе вершин (n) мы подразумеваем случайную величину, принимающую значения на множестве всех ненаправленных графов на n вершинах. [...] Модель случайного графа задается последовательностью граф-значных случайных величин, по одной для каждого возможного значения n: $\mathcal{M} = (G_n; n \in \mathbb{N})$."

[11] "В общем, случайный граф есть модельная сеть, в которой некоторый определенный набор параметров принимает фиксированные значения, а в остальных отношениях сеть случайная."

[12] "[...] чтобы определить случайный граф на N узлах, мы должны задать множество $G \subseteq \{0,1\}^{N^2}$ разрешенных графов (конфигурационное пространство) вместе с вероятностным распределением $p(A)$ над этим множеством. Эта комбинация $\{G,p\}$ множества графов G с ассоциированными вероятностями называется ансамблем случайных графов. Эквивалентно, мы всегда можем положить $G = \{0,1\}^{N^2}$ и задать $p(A) = 0$ для всех неразрешенных графов A."

Вообще, случайным графом может называться любая модель, задающая вероятностное распределение над некоторым множеством графов. Например, Э. Колачик (E. Kolaczyk) [13] использует понятие модели графа как коллекции $\{\mathbb{P}_\theta(G), G \in \mathcal{G}, \theta \in \Theta\}$, где параметризованное вероятностное пространство $\mathbb{P}_\theta$ определено на некотором ансамбле $\mathcal{G}$ возможных графов. Здесь возникает два способа усложнения модели: с помощью определения $\mathbb{P}(G)$ или в нетривиальном ограничении множества $\mathcal{G}$ разрешенных графов. Во втором случае $\mathbb{P}(G)$ обычно считается равномерным, то есть соответствующий генератор случайно

[1] Здесь и далее перевод цитат и названий с английского выполнен автором.

выбирает граф из $\mathcal{G}$, что позволяет сделать аналитически более простую модель, — поэтому ER модель так популярна и хорошо изучена теоретически.

В настоящей работе термины “модель графа” и “модель случайного графа” используются в общем смысле, где над множеством возможных графов $G \in \mathcal{G}$ некоторым образом определено вероятностное распределение $\mathbb{P}(G)$.

1.1.3 Известные графовые характеристики и признаки

По мере развития науки о сетях было открыто множество закономерностей в графах реальных сетей и создано множество способов для измерения их характеристик. Разработка новых характеристик позволяет обнаруживать неизвестные ранее графовые паттерны, которые потом анализируются и вносят свой вклад в понимание природы сложных сетей. Наиболее значимые графовые признаки кладутся в основу моделей графов, с помощью которых можно объяснить появление наблюдаемых свойств.

В этом разделе приведены характеристики, наиболее часто встречающиеся в моделировании графов, начиная с топологических свойств графов, затем описывающие динамику графов и, наконец, характеристики, касающиеся атрибутов вершин и ребер. По ходу перечисления упомянуты соответствующие признаки, присущие графам.

Топология

Топологические характеристики сгруппированы в четыре класса, отражающие их основные аспекты: степени вершин, количество подграфов, свойства связности и спектральные свойства.

Степени вершин Степени вершин служат базисом для целого ряда важных собирательных графовых характеристик.

– *Распределение степеней вершин* (***node degree distribution***, DD). Неожиданным открытием Барабаши и Альберта в 1999 году явилось

то, что многие сети из разных доменов имеют распределение степеней, близкое к степенному закону (***power-law***) [3], то есть $P(d) \sim d^{-\gamma}$ со значением экспоненты $\gamma \geqslant 1$, обычно между 2 и 3. Независимость от какого-либо параметра масштаба в этом законе дала название свойству "безмасштабности" (***scale-free***). Тому же степенному закону обычно подчиняются распределения входящих степеней — считаются только ребра, входящие в вершину, — и исходящих степеней в направленных графах [14]. Позже было показано, что некоторые сети, такие как мобильные звонки, лучше описываются двойным Парето-логнормальным (***double Pareto-lognormal***, DPLN) распределением [15], — нечто среднее между Парето и логнормальным распределениями.

– *Ассортативность степеней* вершин (***degree assortativity***). В одном из вариантов ассортативность вычисляется как коэффициент корреляции между степенями соседних вершин. Положительная корреляция обнаруживается для социальных сетей: узлы с большим числом связей чаще соединяются с узлами с большой степенью и аналогично для малых степеней; такие сети называют ассортативными. Биологические и технологические сети часто являются дизассортативными (корреляция отрцательна) [16].
– *Совместное распределение степеней* (***joint degree distribution***). Для направленных графов количество входящих в узел и исходящих из узла ребер не являются независимыми величинами, что можно выразить с помощью их совместного распределения. Например, сайты веб графа WWW, имеющие много исходящих ссылок, также имеют много входящих ссылок [17].
– *dK-распределения.* dK-распределение показывает корреляцию степеней вершин в подграфе размера d. Для $d = 0$ это средняя степень вершин $\langle d_i \rangle$, для $d = 1$ — классическое распределение степеней, $d = 2$ соответствует совместному распределению $P(d_i, d_j)$. Для $d > 2$ характеристика является комбинацией совместных распределений степеней вершин для каждой возможной (связной) конфигурации ребер на d вершинах. Серия dK-распределений при увеличении d описывает все более и более сложные признаки данного графа, превращаясь в полное описание при $d = n$.

Подграфы Очень полезно подсчитывать количества треугольников (комбинаций из трех вершин) в графе, а также количество подграфов большего размера.

- *Коэффициент кластеризации* (***clustering coefficient***, CC). Коэффициент кластеризации это отношение количества замкнутых треугольников (с тремя ребрами) к числу всех связных треугольников (с двумя или тремя ребрами). Это отношение, будучи измеренным для всего графа, называется транзитивностью, однако чаще используется понятие среднего локального коэффициента кластеризации, когда отношение измеряется для каждого узла и усредняется по всем узлам. Для реальных сетей было обнаружено, что коэффициент кластеризации значительно выше, чем был бы при том же числе ребер, но если бы связи между узлами возникали независимо друг от друга, как в ER модели.
- *Коэффициент кластеризации как функция степени вершины.* Для некоторых сетей такая зависимость близка к степенному закону; это свойство сетей было ассоциировано с их иерархической структурой [18].
- *Распределение подграфов* (***subgraph distribution***). Распределение в графе подграфов из 3 или 4 вершин полезно в двух отношениях. Было показано, что, выступая в качестве вектора признаков, оно содержит достаточно информации, чтобы с хорошей точностью (около 90%) классифицировать графы по своим доменам [19]. В то же время нахождение статистически значимых подграфов в графе, так называемых графовых мотивов (***network motifs***), и их последующий анализ может пролить свет на принципы построения сетей, что оказалось особенно плодотворным в сфере биологии [20].

Связность Измерение расстояний между вершинами и степени их связанности в графе дает представление о его глобальной связности, достижимости узлов друг из друга и связных компонентах графа.

- *Эффективный диаметр* (***effective diameter***). В отличие от обычного диаметра графа, показывающего максимальное расстояние между всеми парами вершин, эффективный диаметр d_{eff} показывает, что подавляющее большинство (обычно берут 90%) пар вершин разделены не более чем d_{eff} ребрами, что является более информативной характеристикой. Известно, что социальные графы и всемирная паутина WWW

имеют малый эффективный диаметр (в 1999 году у веб-графа он был равен $5-7$), что было окрещено как эффект малого мира (***"small-world" effect***) [21]. Этот же факт выражают фраза "мир тесен", а также теория 6 рукопожатий.

– *Достижимость вершин* (***hop-plot***). Для заданной длины пути h вычисляется, какая доля всех пар вершин графа находится на расстоянии не более, чем h друг от друга. Характеристика агрегирует в себе несколько других известных мер расстояний в графе, например, эффективный диаметр и *среднюю длину кратчайшего пути* (***average shortest path length***). М. Фалутсос и др. [22] обнаружили, что граф Интернета демонстрирует степенной характер зависимости в этой характеристике: число связанных пар узлов растет как степень h при $h \ll d_{eff}$.

– *Компоненты связности* (***connected components***). Чаще всего сеть представляет собой связный граф или имеет одну большую (*гигантскую*, ***giant***) компоненту связности. В связи с этим часто возникает вопрос о наличии гигантской компоненты в случайном графе, или ее появлении при определенных условиях (фазовый переход) [23], что является предметом исследования в теории перколяций.

– *Структура сообществ* (***community structure***). Наличие групп узлов, более тесно связанных между собой чем с остальными узлами графа, характерно, в частности, для социальных сетей. Такие группы отражают группы пользователей, объединенных общими интересами в сообщества, а в биологических сетях, они соответствуют, например, множеству белков, выполняющих схожие функции. Степень выраженности сообществ в графе с точки зрения топологии измеряется модулярностью (***modularity***) [24]. Также для характеристики сообществ используют проводимость (***conductance***), отделимость (***separability***) и сплоченность (***cohesiveness***) [25].

Спектр Графовые свойства тесно связаны с его спектральными свойствами, т.е. собственными значениями, собственными векторами его матрицы смежности и матрицы Лапласа. Спектральный анализ используется для изучения процессов в сетях и разработки алгоритмов на графах. Например, собственный вектор Перрона-Фробениуса веб-графа лежит в основе поисковой системы

Google. В целом это является предметом спектральной теории графов [26]. В качестве ключевых спектральных характеристик рассмотрим следующие.

- *Спектральный радиус* ***sectral radius***. Максимальное собственное значение $|\lambda_1|$ матрицы смежности графа называется его спектральным радиусом. Поскольку для несвязного графа $|\lambda_1| = 0$, спектральный радиус вычисляют для гигантской компоненты. Радиус имеет свойство не возрастать при удалении вершин или ребер графа и служит как альтернативная мера размера графа. Ван Мигхем Пит и др. [27] показали, что чем меньше значение спектрального радиуса, тем выше устойчивость графа к распространению вирусов.
- *Алгебраическая связность* ***algebraic connectivity***. Второе наименьшее ненулевое собственное значение матрицы Лапласа называется алгебраической связностью графа. Также ее можно измерять для гигантской компоненты. Алгебраическая связность тем выше, чем более связным является граф. Соответствующий собственный вектор, вектор Фидлера, полезен при решении задачи разбиения графа (***graph partitioning***) [28].
- Распределение сингулярных значений матрицы смежности ***singular value distribution of the adjacency matrix***. В реальных сетях такое распределение подчиняется степенному закону для $n^{1/2} - n^{2/3}$ наибольших сингулярных значений [29].
- Распределение собственных значений матрицы Лапласа ***eigenvalue distribution of the Laplacian matrix***. Данное распределение тоже описывается степенным законом: k максимальных собственных значений $\lambda_k \sim k^{\alpha}$, где k порядка $n^{2/3}$, а степень α обычно варьируется в пределах $(2; 10)$ [29]. Также было замечено, что показатель степени часто близок к показателю степени в распределении вершин, — для графов, где эти два распределения подчиняются степенному закону.

Существует множество других топологических графовых характеристик, не столь важных для проектирования моделей графов, однако полезных при их оценке. Более подробный обзор можно найти в [18] и [30].

Динамика

Многие сети изменяются со временем, что выражается в появлении и исчезновении в графе ребер и узлов. Чаще всего происходит рост графов, то есть увеличение числа вершин n со временем t. Все статические топологические характеристики могут быть измерены как функции от времени или числа вершин в случае их монотонного роста, что позволяет обнаруживать новые, динамические паттерны.

Следующие паттерны были найдены при исследовании графов из социального домена (графы цитирования, коллаборации, социальные сети) и автономного (Интернет-коммуникации).

- *Степенной закон уплотнения* (***densification power law***). Число ребер m растет пропорционально степени от числа вершин: $m(t) \sim n(t)^{\alpha}$, где $1 < \alpha < 2$ [31].
- *Сокращение диаметра* (***shrinking diameter***). Несмотря на то, что размер гигантской компоненты увеличивается с добавлением в нее новых узлов, ее эффективный диаметр уменьшается [31].
- *Точка застывания* (***gelling point***). При наблюдении изменения характеристик растущего графа, отмечается наличие момента стабилизации, когда график диаметра показывает пик. После этой точки диаметр перестает расти и начинает сокращаться, а также проявляются некоторые закономерности: выполняется степенной закон уплотнения, размеры второй и третьей по величине компонент связности осциллируют возле определенной величины [32].

Атрибуты

Во сетях реального мира содержится много информации помимо их топологии. Вершины часто имеют атрибуты: данные профиля пользователя в социальной сети, известные свойства белка и т. д. Ребра также могут быть помечены метками времени, иметь веса и т. д. В этой работе рассматриваются только метки принадлежности узла сообществам и веса для ребер.

Метки сообществ вершин Социальные сети имеют явную структуру сообществ, образованную атрибутами пользователей. Такие сообщества в различных доменах имеют общие свойства [33]:

- размер сообществ и число сообществ, которым принадлежит один узел, имеют распределение с тяжелым хвостом;
- степенной закон уплотнения: в пределах одного графа количество ребер в сообществе c растет как степень от его размера, $m_c \sim n_c^{\alpha}$.

Также известны другие свойства сообществ, важные для создания моделей графов с сообществами: вероятность ребра между вершинами p_{ij} увеличивается с числом общих сообществ для i и j; вершины, лежащие в пересечении сообществ, соединены более плотно, чем вершины из непересекающихся частей сообществ; и другие.

Веса ребер *Вес ребра* (***edge weight***) обычно выражает силу связи между двумя узлами, — это может быть, например, количество совместных появлений пары слов в тексте, объем сетевого трафика между узлами сети. Также вес может служить указанием на наличие кратного ребра, например, выражающего общее количество цитирований одним автором другого. Для взвешенных графов был установлен следующий ряд степенных законов [32]:

- *Степенной закон для суммарного веса* (***weight power law***). Сумма весов всех ребер растет как степень от количества ребер в графе: $W(t) \sim m(t)^w$ с экспонентой $w > 1$.
- *Степенной закон для мощности вершины* (***snapshot power law***). Мощность вершины (***node strength***) s_i, определенная как сумма весов ее смежных ребер, зависит от степени этой вершины d_i по степенному закону: $s_i \sim d_i^w$. То же выполнено по отдельности для входящих и исходящих ребер в случае направленного графа.
- *Степенной закон для взвешенного собственного значения* (***weighted principal eigenvalue power law***). Наибольшее собственное значение взвешенной матрицы смежности W_{ij} растет как степень от количества ребер: $\lambda_{max}(t) \sim m(t)^{\beta}$, где экспонента β бывает около $0.5 - 1.6$.
- *Самоподобие увеличения веса* (***self-similar weight addition***). Темп увеличения суммарного веса на ребрах графа обладает свойствами самоподобия.

Краткая сумма по перечисленным характеристикам и паттернам дана в таблице 1. Отметим наличие десяти степенных законов, наблюдаемых в реальных сетях.

Таблица 1 — Сводная таблица описанных графовых характеристик и признаков. PL обозначает степенной закон; переменные в каждом ряду независимы.

Класс	**Характеристика**	**Часто наблюдаемые паттерны**
Степень	распредел. степеней	PL: $P(d) \sim d^{-\gamma}, \gamma \geqslant 1$, $\gamma \in (2; 10)$. DPLN
	ассортативность	> 0 в социальном, < 0 в биологическом и технологическом доменах
	dK-распределение	
Подграфы	CC	гораздо больше, чем в модели ER
	CC(d)	PL
	распредел. подграфов	
Связность	эффективный диаметр	эффект малого мира: $d_{eff} \approx 6$
	достижимость вершин	PL: $N(h) \sim h^{\alpha}$ для $h \ll d_{eff}$
	связные компоненты	есть гигантская компонента связности
	структура сообществ	высокая модулярность
Спектр	спектральный радиус	
	алгебраич. связность	
	сингуляр. значения A_{ij}	PL: $\lambda_k \sim k^{\alpha}$ для $k < n^{1/2}$, $\alpha \in (2; 10)$
	собственные значения матрицы Лапласа	PL: $\lambda_k \sim k^{\alpha}$ для $k < n^{2/3}$, $\alpha \in (2; 10)$
Динамика	зависимость $m(n)$	PL: уплотнение $m(t) \propto n(t)^{\alpha}, \alpha \in (1; 2)$
	$d_{eff}(t)$	сокращение со временем
	свойства от времени	наличие точки застывания
Метки сообществ	размер сообщества	распределение с тяжелым хвостом
	число сообществ у вершины	распределение с тяжелым хвостом
	зависимость $m_c(n_c)$	PL: уплотнение $m_c \sim n_c^{\alpha}$ для сообществ c
Веса ребер	суммарный вес	PL: $W(t) \sim m(t)^w, w > 1$
	мощность вершины	PL: $s_i \sim d_i^w$
	взвешенное собственное значение	PL: $\lambda_1(t) \sim m(t)^{\beta}, \beta \in (0.5; 1.6)$
	увеличение веса	самоподобие по времени

1.2 Анализ релевантной литературы

В этой части проанализируем наиболее значимые существующие обзоры и приведем существующие схемы классификации моделей графов, предложенные ранее.

При проведении обзора литературы было обнаружено более десятка крупных работ, посвященных моделированию случайных графов, которые рассмотрены в этой части. О широте исследований в этой области можно судить по названию книг “Сети: введение” Марка Ньюмена (Mark Newman) 2010 года [11] и “Введение в случайные графы” А. Фриза и М. Каронски (A. Frieze и M. Karoński) 2015 года [34].

1.2.1 Процедура сбора публикаций

Все публикации, связанные с моделями случайных графов, можно условно разделить на три типа:

1. **обзоры**: статьи, посвященные обзору или сравнению графовых моделей;
2. **нововведения**: работы, предлагающие новый подход или расширение уже существующего;
3. **приложения**: работы, описывающие применение существующих моделей к прикладной задаче.

Во время поиска было обнаружен, что последний класс слишком широк, чтобы его охватить. При наличии десятков обзоров и сотен работ, предлагающих новые модели, число работ с приложениями гораздо больше. Поэтому был проведен анализ существующих обзоров и сделана попытка рассмотреть наиболее значительные публикации из второго класса.

Запросы к базам данных В качестве источников публикаций были проанализированы три базы данных: Google Scholar, ACM Digital Library и IEEE Xplore Digital Library. За отправную точку была взята начальная коллекция

ранее известных статей ([6; 18; 31; 35–57]), которая итеративно расширялась результатами запросов к упомянутым базам.

Для Google Scholar были сагрегированы результаты от следующих запросов (сортировка по релевантности):

– ‘‘`(random OR artificial OR synthetic OR model OR modeling) (graph OR graphs OR network OR networks)`’’ — первые 150 статей;
– ‘‘`(random OR artificial OR synthetic OR model OR modeling OR modelling) (graph OR graphs OR network OR networks) (generation OR generating OR generator))`’’, — первые 150 статей;
– ‘‘`(random OR artificial OR synthetic OR model OR modeling OR modelling) (graph OR graphs OR network OR networks) (generation OR generating OR generator OR generative))`’’ — первые 50 статей для трех результатов с ограничением с 2009 года, с 2013 года и с 2016 года.

К сожалению, несмотря на вариативность запросов, какие-то важные работы могли быть упущены. В то же время, результаты запросов включают множество нерелевантных статей, поэтому отсечение по числу первых результатов послужило неким компромиссом.

Запросы для ACM Digital Library:

– “any field” matches all: `random graph network model generation`. Сортировка по релевантности, первые 50 статей;
– “abstract” matches all: `random graph network model generation`. Сортировка по релевантности, первые 50 статей;
– “abstract” matches all: ‘‘`random graphs`’’ `network model generator`, и “abstract” matches any: `review survey overview comparison`. Сортировка по релевантности, первые 30 статей;

Для IEEE Xplore Digital Library выполнен поиск по метаданным и взяты 10, 32 и 33 работ из следующих запросов:

– ‘‘`random graph`’’ `AND network AND model AND generator`;
– ‘‘`random graph`’’ `AND network AND model AND generation`;
– ‘‘`random graph`’’ `AND network AND model AND generating`.

По-видимому, Google Scholar индексирует большую часть искомых публикаций и выдает наиболее релевантные результаты, — в общей сложности около 300. С помощью баз ACM и IEEE коллекция была пополнена 70-ю и 46-ю статьями соответственно.

Для большей полноты была использована база книжных изданий Google Books, использованы ссылки в найденных работах, а также случайные работы, попадавшиеся во время исследования.

Статьи, датированные ранее 2003-го года не рассматривались (так как большинство из них мало актуальны), за исключением известных работ таких авторов как Пал Эрдеш, Альфред Реньи и Марк Ньюман.

1.2.2 Обзор обзоров

За последние 15 лет, пожалуй, наиболее значительные труды в рассматриваемой области представлены в монографиях [9; 11–13; 34; 58–69]. Обзоры и сравнения моделей случайных графов проведены в [1; 10; 36–41; 66; 70–73].

Чтобы создать общее представление о содержании крупных изданий, они было проанализированы с точки зрения нескольких тем, представляющих наибольший интерес в контексте работы. В таблице 2 дана сводка о степени охвата каждой из тем. Рассмотрим теперь, какие аспекты важны в контексте моделирования случайных графов и почему.

Описание моделей и генераторов

Основной интерес представляют модели и генераторы случайных графов. Каждая из представленных публикаций в большей или меньшей степени содержит описания существующих моделей и генераторов. Наибольшее внимание различным моделям уделяется в [11; 12; 34; 61; 63; 67; 74]. Книга Мэтью Пенроуза (Mathew Penrose) [59] целиком посвящена случайным геометрическим графам, Дженин Харрис (Jenine Harris) [68] — экспоненциальным моделям.

Таблица 2 — Обзор крупных изданий за последние за последние 15 лет с 2003 года. Для каждой темы указана степень ее раскрытия. Обозначения: ’-’ - отсутствует, ’1’ - тема незначительно затронута, ’2’ - тема раскрыта, ’3’ - есть подробное исследование, ’s’ - тема является основной.

Тема	Dorogovtsev & Mendes [58]	Penrose [59]	Newman, Barabasi, Watts [60]	Durrett [61]	Caldarelli [62]	Vespignani, Caldarelli [63]	Bonato [64]	Barrat [9]	Kolaczyk [13]	Newman [11]	Raigorodsky [66]	Lovász [65]	Chakrabarti [67]	Van Der Hofstad [69]	Frieze, Karoński [34]	Coolen, Annibale, Roberts [12]
Год	2003	2003	2006	2007	2007	2007	2008	2008	2009	2010	2011	2012	2012	2014	2015	2017
Описание моделей и генераторов	2	s	2	3	2	3	2	2	2	3	3	1	3	2	3	3
Классификация моделей и генераторов	1	-	-	1	-	1	-	2	2	2	1	-	2	-	-	2
Примеры и классификация реальных сетей	3	-	1	-	3	2	s	2	2	3	-	1	-	1	-	-
Графовые признаки и характеристики	2	-	1	-	2	2	s	2	3	3	-	-	3	1	-	1
Описание приложений	-	-	2	-	1	-	-	2	1	-	1	-	-	-	-	3
Алгоритмы и процессы на сетях	-	-	1	-	-	3	2	3	2	3	-	-	1	1	1	-
Теоретическое введение	3	1	-	-	2	2	2	1	3	2	3	1	1	2	1	2
Математические результаты	2	3	2	3	2	1	3	2	1	2	3	s	-	3	3	3
Приведены датасеты	-	-	-	-	-	-	-	-	1	-	-	-	1	-	-	-

Классификация моделей и генераторов

Число различных моделей и генераторов, предложенных к настоящему времени, исчисляется сотнями и даже тысячами, поэтому становится необходимым иметь более общее представление о них. Классификация существующих подходов может быть более информативной, чем описание конкретных алгоритмов.

В литературе не было обнаружено какой-либо исчерпывающей таксономии моделей, найденные обзоры неполны, либо устарели к настоящему времени. Поскольку традиционной классификации моделей нет, каждая работа предлагает свой взгляд на положение дел. Самые детальные классификации предлагаются в [36;67;71;72;74], а Д. Чакрабарти и К. Фалутсос (D. Chakrabarti и C. Faloutsos) [67] приводят таблицу из 24 генераторов, оцененных по нескольким параметрам. Альтернативные версии классификаций также имеются в [9–13; 63; 67].

Примеры и классификация реальных сетей

Графовые данные приходят из многих областей исследований и могут быть дифференцированы по доменам, например, социальный, биологический, и специфике графа, т. е. является ли граф большим или маленьким, направленным, взвешенным, содержит ли метаданные и т. д. В каждом графовом домене имеются свои характерные задачи, которые предъявляют к моделям графов специфические требования. Например, биологический граф с сотней узлов и социальный граф с миллионами узлов и миллиардами ребер требуют разных подходов к моделированию и вызывают разного рода проблемы.

Обычно принято выделять от 4 до 10 доменов сетей [1;9;11;58;62]. Иногда в них выделяются поддомены, — например, коллекция графов Konect [75] содержит 24 категории, — однако, общепринятой иерархии не существует. Согласно Марку Ньюмену [1;11], можно выделить 4 графовых домена:

– Технологический (***Technological***) — Интернет, телекоммуникации, электросети, транспортные сети и т. д.;

– Социальный (***Social***) — онлайновые социальные сети, сети аффилиаций, коллабораций и т. д.;
– Информационный (***Information***) — всемирная паутина, сети цитирований, одноранговые сети, лингвистические графы и т. д.;
– Биологический (***Biological***) — сети взаимодействий между белками, метаболические, нейронные сети, пищевые цепочки и т. д.

Графовые признаки и характеристики

Большинство сетей реального мира из разных областей имеют общие закономерности: степенной закон распределения степеней вершин, малый диаметр, высокий коэффициент кластеризации [76]. Эти признаки широко представлены на практике, но не отражены в классической ER модели, что порождает потребность в моделях, обладающих такими свойствами.

В контексте моделирования случайных графов знание графовых характеристик и признаков полезно по нескольким причинам:

– воспроизведение известных графовых признаков делает модели более реалистичными;
– графовые характеристики позволяют сравнивать результаты работы моделей случайных графов, тем самым являясь инструментом для оценки качества;
– анализ конкретных признаков способствует пониманию объекта изучения — например, графовые мотивы отражают поведенческие паттерны в биологических сетях.

Подробные сведения о графовых свойствах можно найти в [11; 13; 67] и [1; 9; 58; 62]; работа [64] полностью посвящена веб-графу.

Описание приложений

Приложения науки о сетях являются основной целью моделирования графов, они же устанавливают требования и ограничения на модели. При-

ложения моделей случайных графов могут быть рассмотрены с двух точек зрения. Во-первых, в каждом домене возникают специфические задачи, например Т. Коолен, А. Энибейл и Е. Робертс (T. Coolen, A. Annibale и E. Roberts) [12] рассматривают типичные задачи, возникающие в 5 выделяемых авторами доменах:

- электрические сети (***power grids***) — предупреждение отключения энергии;
- социальные сети (***social networks***) — поиск сообществ и их анализ, распространение инфекций, слухов и т. п.;
- пищевые цепочки (***food webs***) — пищевые отношения в экосистемах, вопросы стабильности;
- всемирная паутина (***world wide web***) — улучшение поисковых движков;
- белок-белковые взаимодействия (***protein-protein interactions***) — поиск графовых мотивов, исследование взаимодействий лекарств.

С другой стороны, определенные типы задач могут возникнуть в нескольких доменах. Такому взгляду более всего соответсвтуют работы [9; 60]. Задачи можно сгруппировать следующим образом:

- поиск в сети, навигация по сети;
- устойчивость к атакам и ошибкам (Интернет, социальные сети, электросети), надежность сети (теория перколяций);
- распространение информации, в т. ч. эпидемий, глобальные каскадные процессы.

Другие темы

Для полноты представления о рассмотренной литературе приведены следующие три темы, не имеющие прямого отношения к моделированию случайных графов.

Алгоритмы и процессы на сетях Модели случайных графов используются для разработки и тестирования различных сетевых алгоритмов и анализа процессов, происходящих в сетях. Четкого различия между понятиями алгоритма

и процесса на графе нет, поэтому проиллюстрируем их с помощью примеров. Примеры алгоритмов:

– предсказание топологии сети (***topology inference***): предсказание появления ссылок, вывод ассоциативных сетей, предсказание топологии с помощью томографии сети;
– майнинг графа (***graph mining***): поиск сообществ, вычисление модулярности, вычисление пэйдж-ранк и т. д.

Процесс на сети характеризуется случайными величинами X_i (статический) или $X_i(t)$ (динамический), определенными на узлах. Примеры процессов:

– статические: предсказание ближайшего соседа, Марковские случайные поля, ядерная регрессия;
– динамические: распространение информации, вирусов, модели эпидемий, сетевые потоки (трафик), и т. д.

Много исследований алгоритмов и процессов на сетях можно найти в работах [9; 11; 13; 63].

Математические результаты Одним из направлений исследований является теоретической анализ моделей и генераторов графов. Некоторые модели графов хорошо изучены ввиду их популярности и/или аналитической простоты, например, теория перколяции для ER модели. Свойства генераторов Кронекеровских графов широко изучены и разработано множество их расширений и модификаций.

Наиболее математически содержательные работы: [12; 34; 59; 61; 65; 69].

Теоретическое введение Наличие подробного теоретического введения необходимо для того, чтобы неспециалист в предметной области мог понять содержание без обращения к дополнительным источникам.

Последовательное введение в теорию графов содержится в [11;12;58;62;63]; а в работах [13; 64; 66; 69] также приводятся математические выкладки.

1.2.3 Существующие классификации моделей случайных графов

Небольшая доля публикаций по моделированию графов предлагает явную классификацию существующих моделей. Кроме того, авторы обычно рассматривают несколько категорий моделей, популярных только в определенной области, представляющей интерес, например, социальных сетей или биологии, поэтому такие классификации неполны.

Чаще всего модели графов делятся на два класса: статические и динамические. В статических моделях число узлов n фиксировано, а ребра определяются в соответствии с некоторыми правилами, основанными на атрибутах узлов в случае их наличия. Модель ER служит здесь простейшим примером, где атрибуты вообще не рассматриваются. Динамические модели предполагают, что узлы и ребра добавляются итеративно в зависимости от текущего состояния графа, — как, например, в процессе предпочтительного присоединения. Часто отдельный класс составляют экспоненциальные модели случайных графов (ERGM), где модель определяется набором условий на графовые статистики.

Модели общего назначения

Если не брать во внимание отдельные специфические модели, большинство популярных схем классификации моделей [9; 11; 13; 71] можно приблизительно свести к следующей:

1. Статические модели (***static***, ***equilibrium***):
   - классические случайные графы: модель ER;
   - модели с обобщенным распределением степеней (***generalized degree distribution***);
2. Динамические модели (***dynamic***, ***growth***, ***evolving***, ***non-equilibrium***):
   - предпочтительное присоединение и его расширения;
   - модели копирования (***copying***) и модели дублирования (***duplication***);
   - модели на основе оптимизации (***optimization-based***);
3. Другие модели:

– экспоненциальные модели (ERGM);
– модели малого мира (***small-world***).

Полезно также рассмотреть классификации, не вписывающиеся в приведенную схему. Так, Д. Чакрабарти и К. Фалутсос [67] предлагают следующую классификацию генераторов графов, состоящую из пяти категорий:

1. Генераторы случайных графов — узлы соединяются случайно;
2. Генераторы на основе предпочтительного присоединения — предпочтение (при соединении ребром) отдается узлам с большей степенью;
3. Генераторы на основе оптимизации — минимизация риска в условиях ограниченных ресурсов порождает степенной закон;
4. Географические модели — география узлов влияет на рост сети и ее топологию;
5. Интернет-специфичные генераторы — гибрид идей для воспроизведения признаков, специфичных для сети Интернет.

Альтернативный взгляд на подходы к моделированию графов развивают Т. Коолен и соавторы в [12]. Подробно рассматриваются ансамбли графов при наложении жестких и/или мягких ограничений:

1. Графы с ограничениями:
    а) мягкие ограничения (***soft constraints***) — генерируемые графы должны иметь заданные свойства в среднем (совпадает с определением ERGM);
    б) жесткие ограничения (***hard constraints***) — каждый генерируемый граф должен иметь заданное свойство;
2. Графы, определенные алгоритмом:
    а) алгоритмы роста сети — предпочтительное присоединение и его расширения;
    б) специальные модели — малого мира, геометрические, планарные, взвешенные и др.

Модели социальных сетей

Модели онлайновых социальных сетей являются очень востребованным и широко развивающимся направлением науки о сетях. В этой области

Р. Тойвонен (Riitta Toivonen et al.) и соавторы [36] предлагают следующую таксономию, хорошо вписывающуюся в обобщенную схему, представленную выше:

1. Эволюционные модели (***evolution models***) — добавление/удаление узлов/ребер происходит по стохастическим правилам, в зависимости от локальной структуры сети:
   - растущие (***growing***) — новые узлы добавляются до достижения графом определенного размера;
   - динамические (***dynamical***) — число узлов фиксировано, эволюция происходит, пока определенные статистики не перестанут меняться;
2. Модели на атрибутах узлов (***nodal attribute models***) — вероятность ребра зависит только от атрибутов узлов (принцип гомофилии, пространственные модели);
3. Экспоненциальные модели (ERGM).

Ф. Амблард (F. Amblard) и коллеги [72] проанализировали модели социальных сетей, опубликованные в Journal of Artificial Societies and Social Simulation за 17 лет (до 2015 года) и разделили их на 9 категорий:

1. Регулярные решетки (***regular lattices***);
2. Случайные графы (***random networks***) — преимущественно ER;
3. Модели малого мира (***small-world networks***);
4. Безмасштабные сети (***scale-free networks***) — преимущественно предпочтительное присоединение;
5. Пространственные модели (***spatial networks***) — построенные на основе пространственного распределения агентов, используя расстояния;
6. Иерархические структуры (***hierarchical structures***) — древовидные графы для организационных структур или фамильных сетей;
7. Графы родства (***kinship networks***) — двудольные графы для фамильных сетей;
8. Эмпирические сети (***empirical networks***) — сгенерированные из эмпирических данных из социальных сетей;
9. Другие виды моделей — ad hoc модели, разработанные строго под моделируемую систему.

М. Берновский и Н. Кузюрин [74], хотя и рассматривают ограниченный круг моделей, предлагают классификацию на основе сложности модели:

1. Случайные графы — ER модель и ее расширения;

2. Простейшие безмасштабные сети — модель Боллобаша (Bollobás) [77] и ее расширения, модель копирования и т. д.;
3. Более гибкие безмасштабные модели — модели с обобщенным распределением степеней: Чунг-Лу (Chung-Lu) [78], Янсон-Лучак (Janson-Łuczak) [79] и другие;

а также они предлагают дальнейшее разделение безмасштабных сетей:

1. С фиксированной экспонентой — степенной закон распределения степеней и другие свойства доказаны математически: модель Боллобаша-Риордана (Bollobás-Riordan) [77] и ее расширения);
2. С настраиваемой экспонентой — показатель степенного закона можно менять, исследуя фазовый переход: модели Чунг-Лу [78], Янсон-Лучака [79];
3. С неизвестными свойствами — свойства модели пока не доказаны: Forest Fire [31] и другие.

Интересный взгляд представлен в работе [37], где авторы разделяют 6 моделей на 3 категории по методу, лежащему в их основе:

1. Модели, управляемые признаками (***feature-driven***) — воспроизводящие статистические свойства графа: Барабаши-Альберт (Barabási-Albert) [76], ForestFire [31];
2. Модели, управляемые целью (***intent-driven***) — эмуляция процесса образования оригинального графа: Random Walk [80], Nearest Neighbor [80];
3. Модели, управляемые структурой (***structure-driven***) — выявление статистик в графовой структуре и попытка их воспроизвести: Стохастические Кронекеровские графы [44], dK-графы [81].

Таким образом, в доступной литературе, не было найдено удовлетворительного обзора существующих моделей случайных графов. Все попытки классификаций на момент работы мало актуальны из-за отсутствия многих современных моделей или далеко не полны. В следующем разделе предлагается собственное видение области и попытка дать всеобъемлющую таксономию подходов к моделированию случайных графов.

1.3 Таксономия подходов к моделированию случайных графов

В этом разделе автором предлагается иерархическая таксономия подходов, применяемых при создании моделей случайных графов. На верхнем уровне таксономия состоит из трех классов, в основе которых лежат различные мотивации (рисунок 1.1).

1. Класс **генеративный** (***generative***) подходов включает все механизмы генерации графа, изобретенные для качественного объяснения графовых паттернов. Порядок разработки: построение графа в соответствии с некоторыми правилами, а затем проверка свойств, которыми он обладает, на соответствие известным признакам.
2. Класс **управляемых признаками** (***feature-driven***) подходов фокусируется на создании модели, которая количественно воспроизводит требуемые графовые признаки. Порядок разработки обратный: имея набор желаемых графовых свойств, происходит разработка или настройка модели, для удовлетворения этих свойств.
3. Класс **предметно-специфичных** (***domain-specific***) подходов затрагивает методы генерации графов с дополнительными атрибутами, такими как структура сообществ и веса ребер.

Первые два класса охватывают все модели простых и ориентированных графов, в то время как класс предметно-специфичных подходов выходит за рамки моделирования простых ориентированных графов на другие типы графов, количество которых не ограничивается.

Каждый класс содержит несколько категорий, отражающих различные направления мысли. Верхнеуровневые категории делятся на подкатегории. Ниже следует подробное описание и анализ их всех с описанием конкретных моделей.

Заметим, что некоторые модели встречаются в нескольких категориях, поскольку они основаны более чем на одном походе. Хотя далеко не все релевантные модели упомянуты в каждой категории — целью является лишь проиллюстрировать идеи, — по возможности включено большинство известных моделей и генераторов графов.

Рисунок 1.1 — Таксономия подходов к моделированию случайных графов с краткими описаниями каждой категории.

1.3.1 Класс генеративных подходов

Начиная с самой простой модели ER, которая является наиболее общей и в то же время наименее реалистичной моделью случайного графа, было создано множество алгоритмов с попыткой объяснить известные паттерны реальных сетей. Модель Барабаши-Альберт демонстрирует степенной закон распределения степеней с помощью принципа предпочтительного присоединения. В модели Ваттса-Строгаца (Wattz-Strogats) для так называемых сетей “малого мира” достигается малый диаметр графа путем случайного соединения узлов в регулярной решетке. Модель Forest Fire показывает степенной закон уплотнения и свойство сокращения диаметра в эволюционирующих графах, она основана на рекурсивном процессе, напоминающем лесные пожары, и так далее. Дальнейшим развитием таких подходов является адаптация оригинальных идей, например, к ориентированным графам, введение новых эвристик и объединение различных функций в одной модели, поэтому большинство таких работ рассмотрено не будет, если они не содержат новых подходов.

Собранные в генеративном классе идеи представляют весь спектр подходов генерации случайных графов, которые удалось найти. Они сгруппированы

в 5 категорий второго уровня: ‘классические подходы’, ‘локальные правила’, ‘рекурсивные процедуры’, ‘атрибуты узлов’ и ‘топология из оптимизации’.

Классические подходы

Простейшая интерпретация случайности в модели случайного графа состоит в том, чтобы соединять каждую пару узлов независимо друг от друга. Одна из первых таких, модель ER [2] стала классической: на множестве n узлов каждое ребро появляется с постоянной вероятностью p. Хотя модель ER обладает нереалистичными свойствами (пуассоновское распределение степеней, очень низкий коэффициент кластеризации и т. д.), она очень богата теоретическими результатами, например, разработанной теорией фазовых переходов [82].

Другая известная конструкция, названная моделью малого мира, была нацелена на достижение малого диаметра в сочетании с высоким коэффициентом кластеризации. В модели Ваттса-Строгаца [21] на вход принимается регулярная решетка, где каждый узел имеет k соседей. Далее соединяются случайные пары узлов с некоторой вероятностью p. Существуют промежуточные значения p между 0 (регулярная решетка) и 1 (полностью случайный ER граф), соответствующие области “малого мира”, где коэффициент кластеризации все еще высок, а средняя длина пути достаточно уменьшилась.

Локальные правила

Следуя Васкесу (Vázquez) [80], под термином “локальные” будем подразумевать, что процесс роста графа определяется правилами, вовлекающими узел и его соседей. Одно из таких правил, мотивированное наблюдением, очень популярным в социальной сфере, называется “триадным замыканием” (***triadic closure***) [83]. Оно говорит о том, что вероятность ребра (u, v) выше, если узлы u и v имеют общего соседа. Это выражается высоким коэффициентом кластеризации в реальных графах по сравнению со случаем независимого соединения узлов в модели ER.

Принцип предпочтительного присоединения Два фактора: рост графа и идея присоединять новый узел с большей вероятностью к узлу, имеющему высокую степень, — вместе приводят естественным образом к степенному закону распределения степеней. Таким образом предпочтительное присоединение (***Preferential Attachment***, PA) используется в модели Барабаши-Альберт [3] для объяснения свойства безмасштабности, наблюдаемого во многих реальных сетях. Принцип PA широко используется в моделях графов и существует во множестве вариаций. В исходной форме утверждается, что вероятность образования ребра из нового узла в узел i пропорциональна степени этого узла: $p_{ij} \sim d_i$, нормализованной по всем узлам i, присутствующим в графе в данный момент. Однако, при этом оказывается предопределен фиксированный показатель степенного закона $\gamma = 3$ [3]. Дальнейшие шаги эволюции PA включают следующие.

- Введение новых параметров в PA правило, например, $p_{ij} = \dfrac{A + d_i}{\sum_i (A + d_i)}$ дает гибкий показатель экспоненты $\gamma = 2 + \frac{A}{\Delta m} \in [2, \infty)$, где Δm это количество новых ребер, добавляемых на каждом шаге, A — параметр [58].
- Модификация PA правил. В модели Боллобаша-Риордана [84] вначале строится граф G_1^n с n вершинами и $1 \cdot n$ ребрами. G_1^n получается из G_1^{n-1} добавлением одной вершины и одного ребра согласно PA правилу. Чтобы получить G_k^n с n вершинами и kn ребрами строится граф G_1^{kn}, его kn вершин разбиваются на группы по k вершин, которые схлопываются в одну с сохранением ребер (ребра внутри группы становятся петлями). Один из теоретических результатов модели состоит в том, что диаметр графа $\approx \dfrac{\ln n}{\ln \ln n}$, что согласуется с эмпирическим значением, равным 6 для Интернета в 1999 году.
- Нелинейное PA. Линейное по степени узла PA правило можно обобщить на произвольную функцию. Например, $p_{ij} \sim (1 + d_i)^{\beta} - \lambda$, где параметры β, λ специально подбираются: для реальных сетей подходящие β варьируются от 0 до 1.6 [85].

PA правило послужило базой для многих более поздних моделей, которые включают структуру сообществ в графе, более высокий коэффициент кластеризации [86] и так далее.

Принцип копирования Естественным механизмом процесса формирования сетей является дублирование ее частей с возможным добавлением изменений. Копирование элементов происходит в различных реальных сетях. Гены могут дублироваться в процессе эволюции, поэтому их ребра-взаимодействия дублируются в сетях взаимодействия белков. В веб-графе, а также в графах цитирования авторы могут переиспользовать большинство ссылок с одной страницы (работы) на другую по сходной теме.

Первоначальная формализация Джона Клейнберга и соавторов (Jon M Kleinberg et al) [87] включает в себя четыре процесса, действующих на каждой итерации: создание/удаление узла и создание/удаление ребра — с некоторыми вероятностями. Ключевым моментом модели является создание ребра. Узел v, для которого нужно добавить ребра, и число ребер k, которые нужно добавить, выбираются из заранее определенных распределений. С вероятностью β узел v связывается с k случайно выбранными узлами, а с вероятностью $(1-\beta)$ копируются ребра случайно выбранного узла u. Такая модель копирования дает степенной закон распределения степеней с показателем $\gamma = \frac{2-\alpha}{1-\alpha} \in [2;3]$ в зависимости от фактора роста $\alpha = \frac{\beta}{1-\beta}$. Также показано образование большого количества двудольных клик (как у веб-графа), создающих небольшой эффект наличия сообществ [88].

В модели Растущей сети с копированием (***Growing network model with copying***) [89] в дополнение к копированию ребер целевого узла u сам выбранный узел v также соединяется с u. Это дает эффект того, что число ребер m растет быстрее, чем число узлов n, — закон уплотнения, который наблюдается в реальных сетях.

В модель можно добавить аналог мутаций, как сделано в модели Дивергенции дублирования (***Duplication divergence model***) [80]. В предыдущих обозначениях, здесь после копирования ребер для каждого из соседей j одно из двух ребер (u, j) или (v, j) удаляется с вероятностью $1 - q_e$. В результате коэффициент кластеризации как функция степени узла показывает уменьшение по степенному закону с показателем степени зависящим от q_e.

Алгоритм репликации сложных сетей (***Replicating complex networks***, ReCoN) [56] копирует данный граф k раз, а затем применяется переключение ребер (***edge switching***) для связывания реплик между собой и рандомизации. Утверждается, что ReCoN сохраняет коэффициент Джини (***Gini coefficient***)

для распределения степеней, относительно высокий коэффициент кластеризации[2] и малый диаметр.

Часто принцип копирования структур присутствует во многих графовых моделях неявно или среди других механизмов. Например, в модели Forest Fire [31] новый узел присоединяется к соседям целевого узла (с вероятностью "горения") и продолжается рекурсивно. В алгоритме GScaler [49] входной граф разрезается на отдельные узлы с половинками ребер, эти элементы множатся их и затем соединяются в соответствии с заданной функцией корреляции степеней вершин.

Другие локальные правила В мире моделей растущих графов, вероятно, в качестве эволюции принципа PA появились различные другие локальные подходы. Была показана их способность объяснить важные графовые свойства, такие как ассортативность и обратно пропорциональная зависимость между коэффициентом кластеризации и степенью вершины [80]. Приведем теперь примеры различных локальных правил.

Модель Случайных блужданий, ***Random Walks*** [80]. Новый узел v соединяется со случайным из существующих узлом w. Затем с некоторой вероятностью q_e он соединяется с одним из своих соседей w'; если ребро создано, очередь переходит к соседу w' и так далее, порождая, таким образом, случайное блуждание. В одной из модификаций узел v пытается соединиться с каждым из соседей w, что похоже на обход в ширину. Такие правила случайного блуждания приводят к степенному закону входящих степеней и относительно высокому коэффициенту кластеризации.

Модель Ближайших соседей, ***Nearest Neighbors*** [80]. Новый узел v также соединяется с w, а затем с вероятностью p соединяется с одним из своих соседей. Помимо степенного закона распределения степеней, этот простой механизм обеспечивает две нетривиальные закономерности, наблюдаемые в социальных сетях: коэффициент кластеризации как функция степени узла подчиняется степенному закону; средняя степень соседа как функция степени узла возрастает.

Модель Лесного пожара, ***Forest Fire*** [31]. Первый шаг на каждой итерации такой же: новый узел v соединяется с w. Среди еще не посещенных

[2]Из-за раздельной процедуры переключения ребер внутри сообществ и между сообществами, коэффициент кластеризации не сильно падает. В общем случае переключение ребер не сохраняет это свойство.

его соседей выбирается x узлов, достижимых через исходящие ссылки, и y через входящие ссылки (или все что есть, если число соседей меньше). В узле v создаются исходящие ребра к выбранным узлам, которые отмечаются как посещенные, и процесс продолжается рекурсивно. Значения x и y выбираются из геометрических распределений, параметризованных с вероятностями прямого p и обратного rp горения. Эта модель демонстрирует ряд существенных особенностей: распределения входящих и исходящих степеней имеют тяжелые хвосты, выполнен степенной закон уплотнения и сокращающийся диаметр. Согласно экспериментам с социальными сетями, модель также показывает коэффициент кластеризации, соответствующий реальным данным [37].

Наиболее популярная локальная эвристика включает создание триадных замыканий. Они могут быть сформированы с некоторой вероятностью на каждой итерации алгоритма, например, соединяются два случайных соседа узла i, если они еще не связаны [90], или сосед соседа узла i соединяется с i [91]. Эти модели также могут использовать случайное удаление узлов с некоторой вероятностью на каждом шаге [90] или случайное удаление ребер [91]. Тогда монотонный рост сменяется динамичной эволюцией; процесс продолжается до тех пор, пока не будет достигнуто стационарное распределение (степеней вершин, средняя степень).

Рекурсивные процедуры

Одно из существенных открытий о структуре сложных сетей касается их самоподобной природы. Пользователи в социальных сетях, а также компьютерные сети образуют сообщества (более тесно связанные группы), состоящие в свою очередь из более мелких сообществ, и так далее. Другим признаком иерархической организации является свойство безмасштабности, вместе с высоким коэффициентом кластеризации [92]. В связи с этим был предложен ряд графовых моделей, основанных на рекурсивных алгоритмах.

Один из первых детерминистических методов начинается с небольшого начального графа G_0 (или одного корневого узла) и на каждом шаге создает N реплик текущего графа G_k. Реплики связаны друг с другом по такой же схеме, как G_0, например, корневой узел связывается со всеми узлами на

нижнем уровне [93]. Доказано, что детерминированная рекурсивная процедура дает степенной закон распределения степеней, высокий коэффициент кластеризации и его обратную зависимость от степени узла $CC(d) \sim d^{-1}$ [94]. Другие варианты алгоритма основаны на итеративном добавлении новых $a \cdot d_i$ узлов к каждому из существующих узлов i в сочетании с переключением ребер [95]; на замене каждого ребра двумя параллельными путями, состоящими из u и v ребер ((u, v)-цветок) [96]; и на замене каждого ребра начальным графом [97].

Одна из наиболее влиятельных идей в этой категории, основанная на рекурсивно определенной матрице, появилась в модели R-MAT [42]. Матрица смежности A_{ij} размером $2^k \times 2^k$ разбивается на четыре равные части рекурсивно до достижения одной ячейки. Для графа на $n = 2^k$ узлах, ребра определяются на основе этих разбиений в соответствии с четырьмя вероятностями a, b, c, d попадания в каждую четверть. Для сэмплирования очередного ребра выбор четверти совершается k раз, приводя в результате в ячейку (i, j) . Или, в другой интерпретации [43], изначально определенная матрица вероятностей A^1_{ij} умножается на себя с помощью произведения Кронекера k раз, что приводит к вероятностной матрице смежности $A^k_{ij} = (A^1_{ij})^{\otimes k}$.

Кроме математической простоты, эта рекурсивная процедура обеспечивает набор полезных графовых признаков, если соответствующим образом выбрать начальные параметры. А именно, полиномиальное распределение входящих и исходящих степеней, малый диаметр, мультиномиальное распределение собственных значений и собственных векторов, иерархическая структура сообществ. В более позднем варианте реализации, названной стохастическими Кронекеровскими графами (***Stochastic Kronecker Graphs***, SKG) [44], было также показано выполнение степенного закона уплотнения. Примечательно, что умножение Кронекера здесь имеет решающее значение, так как, например, декартово произведение графов или вышеупомянутая детерминированная конструкция [93] не порождают графы со степенным законом уплотнения.

Модель SKG глубоко изучена и имеет много расширений, по-видимому, благодаря ее математической простоте, сравнительно низкой сложности генерации и наличия специальной процедуры подгона параметров Kronfit [44]. Среди примеров — добавление случайного шума для преодоления осцилляций в распределении степеней [98]; введение связанных параметров для увеличения вариабельности графов при имитации графового домена [99]; введение мно-

жественной фрактальной структуры в модель для расширения пространства покрываемых моделью графов [100].

Похожая идея лежит в основе Генератора мультифрактальных сетей (***Multi-fractal network generator***, MFNG) [45]. В дополнение к рекурсивно заданной вероятности ребра ($p_{ij} = \prod_{q=1}^{k} p_{i_q} p_{j_q}$ с l вероятностями p_{i_q} в качестве параметров), узлы принадлежат рекурсивно определенным категориям. А именно, отрезок [0,1] разбивается на l частей различной длины, определенных дополнительными $l-1$ параметрами. Каждая часть итеративно разделяется в таких же соотношениях еще k раз, образуя тем самым категории. Затем узлы графа равномерно сэмплируются как точки в отрезке [0,1]. Эта схема задает более гибкую модель, кроме того, авторы предлагают процедуру подгона параметров.

Идея рекурсивного построения топологии хорошо согласуется с фрактальной структурой реальных сетей, заодно объясняя ряд степенных законов (распределение степеней, коэффициент кластеризации в зависимости от степени узла, собственных значений) и малый диаметр. Однако, алгоритмы, основанные на рекурсии, часто генерируют графы размером $n = n_0^k$ узлов, что может быть слишком неточно для практических целей.

Скрытые атрибуты

Основная идея состоит в предположении, что вероятность связывания узлов зависит от некоторых свойств этих узлов, выраженных атрибутами. Предпосылка, пришедшая из социальной сферы, называется принципом гомофилии, который утверждает, что сходство привлекает: люди близкого возраста, интересов, профессии, географического положения и т. д. вероятнее будут связаны ребром внутри сети [101]. Идея формализуется путем включения атрибутов узла в модель и определения вероятности ребра как функции атрибутов узла: $p_{ij} = f(\vec{a}_i, \vec{a}_j)$. Такие модели также называют “пространственными” (***spatial***) или говорят о “скрытом пространстве” (***latent space***) в том смысле, что узлы графа представляются точками в некотором пространстве социальных атрибутов. В этой категории выделим два направления: геометрические подходы и на основе меток узлов.

Геометрические Естественная интерпретация атрибутов узлов как географических координат полезна при моделировании специальных беспроводных сетей, сенсорно-приводных сетей и Интернета, где физическое расстояние между узлами напрямую влияет на их соединение друг с другом [102].

Многие подходы вписываются в следующую схему. Сначала выбирается n точек в 1 или 2-мерной области в евклидовом пространстве, обычно равномерно на $[0;1]^2$ или следуя пуассоновскому процессу. Затем сэмплируются ребра с вероятностью, основанной на расстоянии между узлами $dist(i,j)$. Используемая функция зависимости от расстояния варьируется между работами: экспоненциальное затухание $p_{ij} \sim e^{-\alpha dist(i,j)}$ в модели Ваксмана (Waxman) [103]; степенное затухание $p_{ij} \sim \frac{{d_i}^{\alpha}}{dist(i,j)^{\sigma}}$ в модели Юка (Yook) [104] с наилучшим соответствием для Интернета $\alpha = \sigma = 1$; ступенчатая функция: $p_{ij} = p_a$, если $dist(i,j) < H$, иначе $p_{ij} = p_b$ [105]. Задание распределения точек-вершин в виде смеси распределений, например, суммы многомерных нормальных распределений [106], естественным образом моделирует структуру сообществ.

Несмотря на хорошие показатели по коэффициенту кластеризации, корреляции степеней и структуре сообществ в моделях, случайные геометрические графы имеют пуассоновское распределение степеней [59]. Решение проблемы может заключаться в переходе от статической к динамической модели с использованием принципа PA, как это сделано в генераторе BRITE: $p_{ij} \sim d_j \cdot e^{-\alpha dist(i,j)}$ [107].

Если заменить расстояние между узлами на косинусное сходство их векторов, приходим к определению графов скалярного произведения (***dot-product graphs***). Узлы представлены точками в многомерном пространстве, вероятность ребер задается как функция от скалярного произведения (***dot product***) векторных представлений соответствующих узлов: $p_{ij} = f(\vec{r}_i \cdot \vec{r}_j)$ [46]. В генеративной модели для каждого узла выбираются векторы $\vec{u}$ и $\vec{v}$ независимо из вероятностных распределений U, V (а именно, из $\mathcal{U}^{\alpha}[0,1]$ — α степени равномерного распределения) и узлы связываются с вероятностью $p_{ij} = \vec{u}_i \cdot \vec{v}_j$. Включая разреженный случай, для которого $p_{ij} = \frac{\vec{u}_i \cdot \vec{v}_j}{n^b}$, $b \in (0,\infty)$, модель скалярного произведения тщательно изучена теоретически и показана ее способность генерировать графы со степенным законом распределения степеней, малым диаметром и высоким коэффициентом кластеризации [46]. Вектор узла при этом можно интерпретировать как список интересов соответствующего

пользователя в моделируемой социальной сети (люди с общими интересами чаще общаются) или как темы соответствующих веб-сайтов (схожие веб-сайты вероятнее будут связаны ссылкой).

Попытки моделирования сложных сетей в геометрическом контексте привели к предположению, что в основе их структуры лежит гиперболическая геометрия. Было показано, что неоднородность распределения степеней и высокий коэффициент кластеризации свидетельствуют о гиперболической природе графов [48], в частности, например, показатель степенного закона является функцией кривизны пространства. Другими словами, более естественная мера расстояний на графе основана на кратчайшей длине пути (геодезической) и является скорее гиперболической, чем евклидовой. Более того, иерархическая структура и древовидные структуры, распространенные в реальных сетях, лучше вписываются в гиперболическое пространство.

Популярные модели гиперболических случайных графов используют гиперболический диск радиуса $R = 2\log n + C$. n узлов являются случайно распределенными точками с радиальной плотностью $p(r) = \alpha\frac{\sinh(\alpha r)}{\cosh(\alpha R)-1}$ и равномерной по углу. Ребрами соединяются все пары узлов с гиперболическим расстоянием менее R между ними. В этом случае распределение степеней графа оказывается степенным с показателем $2\alpha + 1$, коэффициент кластеризации не убывает при $n \to \infty$ [108], размер второй по величине компоненты связности $O(polylog(n))$ [109], а также известны граничные оценки для диаметра [110].

Модель, называемая геометрическим неоднородным случайным графом (***Geometric Inhomogeneous Random Graphs***, GIRG) [111], по утверждению авторов, содержит (почти наверное) гиперболический случайный граф в качестве подкласса и, кроме того, технически более простая. Она представляет собой смесь модели Чунг-Лу и геометрический подход. Узлы — случайно распределенные точки $\vec{x}_i$ в d-мерном торе с евклидовой метрикой. Как и в модели Чунг-Лу, определяются веса узлов w_i, соответствующие ожидаемым степеням вершин. Функция вероятности ребра совмещает геометрические расстояния и подход Чунг-Лу: $p_{ij} = \Theta(\min\left\{\frac{1}{||\vec{x}_i - \vec{x}_j||^{\alpha d}} \cdot \left(\frac{w_i w_j}{W}\right)^{\alpha}, 1\right\})$. При соответствующих значениях параметров для GIRG доказан набор свойств для генерируемых графов: степенной закон распределения степеней, высокий коэффициент кластеризации, наличие гигантской связной компоненты, полилогарифмический

диаметр и небольшие разделяющие множества; средняя длина пути порядка $O(\log\log n)$ [112].

В заключение можно отметить, что выбор геометрии представления графа можно трактовать как выбор метрики в пространстве векторов его узлов. Стандартной метрикой является евклидова, скалярное произведение и косинусное сходство соответствуют сферической геометрии. Более сложной, но эффективной является гиперболическая метрика.

Метки узлов Помимо геометрического подхода, идея представления узла как вектора атрибутов имеет другие воплощения. Неизменным остается ключевое предположение о том, что вероятность ребра определяется сходством меток узлов.

В Графе случайного набора (***Random typing graph***, RTG) [47] процесс случайного набора символов используется для генерации последовательностей символов, разделенных пробелом. Каждое уникальное слово соответствует новому узлу. На каждом шаге алгоритма метки начального и конечного узлов генерируются параллельно друг с другом по одной букве l, каждая из которых имеет собственную вероятность появления p_l. Между вновь полученными узлами создается ребро либо увеличивается вес ребра, если оно уже существует. Дополнительно, для моделирования гомофилии и структуры сообщества вводится коэффициент дисбаланса $\beta < 1$, который уменьшает вероятность генерации различных букв в одной и той же позиции, то есть, совместная вероятность букв $p(a,b) = \beta p_a p_b$, в то время как $p(a,a) = p_a p_a$, — что позволяет чаще соединять узлы с похожими метками. В RTG модели возникает целых семь степенных законов: распределение степеней; уплотнение; количество замкнутых треугольников, в которых участвует узел; собственные значения матрицы смежности; наибольшее собственное значение как функция количества ребер m; общий вес ребер как функция m; мощность узла как функция его степени.

В моделях R-MAT [42] и SKG [44] начальная матрица вероятностей $\mathbf{A}^1$ может рассматриваться как функция близости отдельных атрибутов. Таким образом, каждый узел представляется уникальной последовательностью из k атрибутов, где k — степень кронекеровского произведения. Отсюда вероятность ребра равна произведению близостей для двух узлов. Таким образом, более высокие значения на диагонали исходной матрицы смежности соответствуют

принципу гомофилии, так как совпадение атрибутов узлов увеличивает вероятность появления между ними ребра.

Топология из оптимизации

Один интересный подход, заслуживающий отдельной категории, касается идеи вывода топологии сети как результата решения некоторой задачи оптимизации. Можно сказать, что организация многих биологических систем, Интернета, сетей связи сформировалась в результате адаптации к окружающей среде в условиях ряда ограничений и максимизации эффективности этой сети. Тогда структура сети может быть получена из оптимизации некоторой целевой функции.

Эвристически оптимизированная модель компромиссов (***Heuristically Optimized Trade-offs Model***) [113] предназначена для объяснения степенного закона распределения степеней в графе Интернета как следствия локальных компромиссов. Узлы в модели сэмплируются равномерно в единичном квадрате, каждый появляющийся новый узел i выбирает узел j для присоединения, минимизируя две цели: географическое расстояние до него $dist(i,j)$ и некую меру центральности этого узла h_j, например среднюю длину пути от j до всех других узлов графа, то есть $\alpha dist(i,j) + h_j \rightarrow min$. Промежуточные значения параметра α соответствуют появлению степенного закона как компромисса между географическими ограничениями и центральностью. Результат обобщили Бергер и др. (Berger et al) [114], которые показали, что конкуренция между стоимостью нового соединения и стоимостью пути вызывает поведение предпочтительного присоединения.

Различные простые топологии могут возникнуть из максимизации функции способности к выживанию: $\alpha\eta_E + (1-\alpha)\eta_R - C \rightarrow max$ [115]. Здесь η_E отражает эффективность функционирования системы, формализованную как обратная величина от средней длины пути в графе; η_R — устойчивость к потенциальному повреждению (например, удалению узла/ребра), нетривиально выражающаяся через размеры сильно связанных компонент после удаления узла; C относится к ресурсным ограничениям, измеряющим стоимость добавле-

ния узлов и ребер. С помощью численных симуляций были получены топологии “звезда”, “хаб”, “круг” и степенной закон для степеней вершин.

1.3.2 Класс управляемых признаками подходов

Ранние модели графов были нацелены на качественное объяснение основных закономерностей, наблюдаемых в реальных сетях, что отражено в классе генеративных подходов. Однако полезнее не только воспроизвести важные графовые признаки, но и иметь возможность контролировать их с помощью параметров. Если модель позволяет регулировать показатель степени степенного закона и коэффициент кластеризации, она становится гораздо более гибким и эффективным инструментом для анализа сетей. К сожалению, на практике свойства генерируемых графов зависят слишком сложным образом от параметров модели. Более того, известные графовые характеристики не являются независимыми и не могут принимать произвольные значения на одном графе. Для решения этих проблем вместе с моделью графа часто разрабатывается процедура оценки ее параметров для удовлетворения заданных требований. Алгоритм подгона модели является ключевым элементом подходов в этом классе.

В отличие от класса генеративных подходов, класс управляемых признаками подходов касается тех, которые либо принимают в качестве входных данных список признаков, которые необходимо воспроизвести в выходных графах, либо непосредственно подгоняют модель под заданный граф, неявно извлекая его признаки. Многие современные модели объединяют парадигмы обоих классов, например, SKG являлся лишь генератором графов, пока для него не была изобретена процедура подбора параметров Kronfit.

В этом классе выделено 3 категории подходов, каждая из которых включает множество вариаций: ‘аналитические подходы’, ‘оптимизация функции’ и ‘редактирование графа’.

Аналитические подходы

Логичный путь заключается в разработке алгоритма генерации графа таким образом, чтобы желаемые характеристики графа можно было аналитически выразить через параметры алгоритма. Такая модель удобна для анализа, позволяет точно контролировать признаки графа и, таким образом, полезна для исследований. Простейшие случаи включают реализацию определенной последовательности степеней вершин: либо явно заданной, либо сэмплированной из некоего семейства распределений, таких как степенной закон или двойное Парето-логнормальное (DPLN) [50].

Конфигурационная модель (***Configuration model***) [116] реализует заданную последовательность степеней узла $\{d_i\}$: каждому узлу i назначаются d_i половинок ребер, возможно направленных, которые затем соединяются случайным образом. В модели Ожидаемой степени (***Expected Degree model***), также известной как модель Чунг-Лу [117; 118], каждому узлу i задается ожидаемая степень w_i, вероятность ребра $p_{ij} \sim w_i w_j$. Обобщенный биномиальный граф (***Generalized Binomial graph***) [119] принимает всю вероятностную матрицу смежности A_{ij} в качестве входных параметров.

Эти простые модели хорошо изучены для различных показателей степенного закона распределения степеней на предмет появления связанных компонент, размера максимальных клик и т. д. [79]. Хотя и являясь плохими моделями для реальных сетей, такие конструкции широко используются в качестве нулевых моделей. Класс всех графов с одной и той же последовательностью степеней узлов является самой распространенной нулевой моделью, и в частности, используется в задаче поиска графовых мотивов [120].

Заметно более сложная задача — воспроизвести нужное распределение подграфов в графе. Триплетная модель (***Triplet model***) А. Вегнера [51] рассматривает генерацию одной из четырех возможных конфигураций ребер (имеющих от 0 до 3 ребер) на каждом триплете узлов в соответствии с вероятностями $p_1, \ldots, p_4$, или 16 вариантов в случае направленных ребер. Распределение подграфов генерируемого графа выражается четырьмя (или 16-ю) уравнениями, связывающими их с вероятностями генерации каждой конфигурации на исходном множестве вершин. Мультиплетная модель (***Multiplet model***), обобщение

на подграфы произвольного размера, также определена Вегнером и исследована, однако число уравнений быстро растет с увеличением размера подграфов.

Другая трудность возникает при удовлетворении требований на несколько графовых признаков одновременно в рамках аналитического подхода. Обычный способ состоит в том, чтобы итеративно модифицировать граф, последовательно добиваясь необходимых свойств. Задача реализации заданных последовательности степеней вершин и коэффициента кластеризации уже нетривиальна и не решена строго. Например, Л. Хит и Р. Парих (L. Heath и N. Parikh) [121] предлагают сначала итеративно добавлять треугольники, чтобы реализовать заданную последовательность числа треугольников у узла, а затем добавлять отдельные ребра, пока желаемая последовательность степеней не будет достигнута. Здесь результирующее распределение степеней будет выполнено точно, в то время как коэффициент кластеризации близок к ожидаемому, но может отклоняться в случае плотных графов, предположительно потому, что подгон степеней вершин нарушает настойку кластеризации, достигнутую на первом этапе.

Несмотря на отсутствие способов точно реализовать набор графовых признаков, часто на практике достаточно приблизительного их воспроизведения в обмен на возможность контроля за большим количеством параметров. Целая ветвь генераторов графов, предоставляющих множество параметров для настройки, служит для создания так называемых эталонных (***benchmark***) графов. Вероятно, наиболее известной является серия алгоритмов Ланчичинетти-Фортунато-Радиччи (Lancichinetti-Fortunato-Radicci, LFR) [52; 122] для генерации ориентированных взвешенных графов со структурой перекрывающихся сообществ. Метод позволяет настраивать показатели степенного закона для распределений входящих, исходящих степеней и размеров сообществ, вместе с их экстремальными значениями, параметр смешивания, управляющий степенью перекрытия сообществ и другие параметры. Такие модели графов обычно состоят из отдельных несложных компонентов, таких как ER и конфигурационная модели, используют жадные алгоритмы и обеспечивают довольно узкий спектр применимости, где параметры можно считать практически независимыми.

Оптимизация функции

В большинстве случаев параметры модели нетривиальным образом определяют свойства графа. Чтобы подогнать параметры для конкретного графа или для заданных значений характеристик, вовлекается весь набор математических методов оптимизации. Традиционно, определяется целевая функция параметров модели и оптимизируется с применением стандартных методов.

Оценка параметров Специфика оценивания параметров сложных сетей заключается в том, что зачастую эмпирические данные ограничиваются единственным графом. Популярным подходом является оценка максимального правдоподобия, при которой вероятность $P(\Theta|G)$ максимизируется по параметрам модели Θ при заданном графе G. Согласно байесовскому подходу, $P(\Theta|G) = P(G|\Theta)\frac{P(\Theta)}{P(G)}$ и теперь, в предположении равномерного априорного распределения $P(\Theta)$, целью максимизации является $P(G|\Theta)$.

В модели SKG [44] исходная вероятностная матрица смежности $\mathbf{A}^1$ должна быть подобрана так, чтобы ее Кронекеровская степень $\mathbf{A}^k$ лучше всего описывала данный граф G. Степень k просто выбирается минимальной для получения достаточного количества узлов. Для нахождения параметров Θ — элементов матрицы A^1_{ij}, — в алгоритме KronFit [44] логарифм вероятности $\log P(G|\Theta)$ оптимизируется градиентным спуском. Основная задача здесь состоит в том, чтобы учесть все возможные $n!$ перестановок узлов для сопоставления $\mathbf{A}^k$ с матрицей смежности графа G: $P(G|\Theta) = \sum_\sigma P(G|\Theta,\sigma)P(\Theta|\sigma)$. Проблема суперэкспоненциального суммирования была эффективно преодолена путем применения сэмплирования по Метрополису для аппроксимации распределения перестановок $P(\sigma|G,\Theta)$, что требует $O(kn)$ шагов.

Альтернативным способом оценки параметров модели является метод моментов. Авторы мультифрактального генератора MFNG [45] тоже рекурсивно моделируют граф, задавая l вероятностей категорий узлов и матрицу $l\times l$ близости категорий. Подгон модели к реальному графу методом моментов решается как задача минимизации отклонения набора целевых значений характеристик от ожидаемых их значений [123]. Сильной стороной подхода MFNG является то, что любая статистика графа, которая может быть выражена через события на подмножестве ребер (число ребер, клик, звезд и т. п.), аналитически

выражается через параметры модели и, таким образом, может использоваться в методе подгонки.

Поскольку модель SKG также позволяет выражать характеристики на основе ребер через параметры модели, здесь тоже можно применить метод моментов [124].

Экспоненциальные модели случайных графов В качестве модели случайного графа целесообразно задать вероятностное пространство на графах $\mathbb{P}(G)$ таким образом, чтобы эти графы удовлетворяли набору условий на свои статистики $F(\vec{s}(G)) = 0$. В общем случае решение состоит в том, чтобы выбрать распределение, которое имеет максимальную энтропию Шеннона $S[\mathbb{P}] = -\sum_{G\in\mathcal{G}} P(G)\log P(G)$, поскольку оно не несет никакой дополнительной информации, кроме той, которая содержится в самих условиях. Максимизация энтропии с учетом условий $\sum_{G\in\mathcal{G}} P(G)\vec{s}(G) = \vec{s}(G^*)$ дает экспоненциальные решения: $P(G) = \frac{1}{Z(\vec{\theta})}e^{\vec{\theta}\vec{s}(G)}$, где параметры модели $\vec{\theta}$ определяются из уравнений этих условий, а $Z(\vec{\theta})$ — нормировочный коэффициент. Идея, лежащая в основе экспоненциальных моделей случайных графов (***Exponential random graph models***, ERGM), строится на объяснении наблюдаемого графа G^* с помощью набора статистик его топологических характеристик или атрибутов узлов. Статистики $s_1(G), s_2(G), ...$ могут представлять собой любые измеримые функции от структуры сети: количество ребер, треугольников, k-звезд, последовательность степеней вершин и атрибуты: возраст, пол пользователей и т. д.

Большинство ERGM, за исключением тривиальных примеров, не решаются аналитически, то есть точное вычисление функции $Z(\vec{\theta})$ и ее производных невозможно. Поэтому для оценки параметров были разработаны приближенные решения и методы максимального правдоподобия или псевдо-правдоподобия [125]. Часто применяется сэмплирование Монте-Карло на марковской цепи (***Markov chain Monte-Carlo***, MCMC), где строится цепь Маркова конфигураций графа с целевым стационарным распределением.

Простейшими случаями ERGM являются модель ER ($s(G) = m$) и конфигурационная модель ($\vec{s}(G) = \{d_i\}$). Более интересный известный пример ERGM — Стохастическая Блочная Модель (***Stochastic Block Model***, SBM). Узлы графа принадлежат одной из Q групп (сообществ) с априорными вероятностями p_i и связаны в соответствии с матрицей близости $\mathbf{P}$ размера $Q \times Q$,

определяющей вероятности связей между группами. Для реальной сети параметры модели могут быть оценены с помощью алгоритма максимизации ожидания (EM-алгоритма) [126].

Сильной стороной ERGM является их математическая строгость, поскольку определяемое в модели распределение является лучшим выбором при заданных ограничениях $\vec{s}(G)$ в статистическом смысле. Это делает ERGM привлекательной нулевой моделью, что активно используется для анализа социальных и биологических сетей.

Однако, довольно скоро появляются серьезные препятствия, если графы становятся больше ($n > 10^4$), а налагаемые условия сложнее, чем линейные функции от матрицы смежности A_{ij}. Вычислительная сложность, а также чувствительность параметров являются основными трудностями; на практике встает вопрос компромисса между точностью модели и ее сложностью [127].

Редактирование графа

Альтернативой построению с нуля случайного графа с желаемыми признаками является рандомизация уже существующего графа при условии сохранения некоторых его свойств. Простейшие операции редактирования графа включают добавление/удаление узлов/ребер; многие методы основаны на их комбинации. Вторичной целью рандомизации графа является привнесение изменчивости в модель.

Переключение ребер Классической процедурой рандомизации графа является переключение ребер (***edge switching*** или ***edge rewiring***), которое многократно применяется для модификации графа G_C таким образом, чтобы некоторый набор условий C оставался выполненным. Наиболее распространенная операция — переключение пары ребер, которое сохраняет степени всех узлов неизменными: пара ребер $i \to j, k \to l$ заменяется на $i \to k, j \to l$.

Важный результат, лежащий в основе многих подходов, использующих переключение ребер, состоит в следующем. Цепь Маркова, начинающаяся с исходного графа G^0, в которой на каждом шаге перехода пара ребер для

переключения выбирается случайным образом, имеет равномерное стационарное распределение по всем графам с одной и той же последовательностью степеней. Более того, такая цепь неприводима, то есть любая возможная конфигурация ребер достижима из любой другой. Эти свойства позволяют легко осуществить равномерное сэмплирование случайных графов с заданной последовательностью степеней [128]. На практике для генерации графов цепь прогоняют некоторое количество шагов, линейное по числу ребер m, пока цепь достигает стационарного режима — эмпирически, $100m$ оказывается достаточным [129].

В случае более сложных условий C применяются стандартные методы сэмплирования Монте-Карло для получения цепи Маркова с желаемым стационарным распределением, соответствующим C. Например, Ин Сяовэй и У Синтао (Ying Xiaowei, Wu Xintao) [55] используют алгоритм Метрополиса-Гастингса для генерации графов с целевым распределением характеристик $g(S)$. А именно, на шаге t потенциальное переключение пары ребер принимается с вероятностью $P_{G^{t-1}\to G^t} = \min\left(1, \dfrac{g(S(G^t))}{g(S(G^{t-1}))} \cdot \dfrac{f(S(G^{t-1}))}{f(S(G^t))}\right)$, где $f(S)$ — распределение характеристики S по всем графам с одинаковой последовательностью степеней. Конкретным примером такого подхода является алгоритм ClustRNet [130], где, кроме распределения степеней, единственными ограничениями являются коэффициент кластеризации и связность графа, поэтому вероятность перехода равна просто 1, только если коэффициент кластеризации графа G^t выше некоторого порога и граф G^t связный, и равна 0 в противном случае. Точно так же можно генерировать случайные dK-графы, то есть с заданным совместным распредлением степеней на подграфах размера d (dK-распределениями) [54].

Подход MCMC при сложных условиях C подвержен двум проблемам: во-первых, не все состояния, удовлетворяющие условиям, могут быть достижимы друг из друга с помощью разрешенных переключений (нарушение эргодичности), а во-вторых, возрастает время сходимости цепи к стационарному распределению.

Чтобы сделать пространство состояний более связным, Л. Табурье, К. Рот и Ж.-Ф. Куанте (L. Tabourier, C. Roth J.-Ph. Cointet) [131] предлагают использовать k-реберные переключения, определяемые для k ребер $\{a_i \to b_i\}_{i=1..k}$, возможно, смежных между собой. Концы ребер $\{b_i\}$ случайным образом пере-

ставляются, в результате чего получается новый набор ребер $\{a_i \to \sigma(b_i)\}_{i=1..k}$, где σ является одной из $k!$ возможных перестановок.

Процедура переключения пары ребер часто используется в качестве дополнительного шага рандомизации в некоторых генераторах графов. Алгоритм ReCoN [56] генерирует большие графы, копируя исходный граф с отмеченными сообществами k раз и переключая ребра внутри новых копий сообществ и между ними. Хотя такие переключения сохраняют степени всех узлов, они потенциально нарушают другие признаки[3]. В генераторе LFR эталонных графов [52] переключения ребер используются для подгона топологических характеристик, а именно, для уменьшения количества ребер внутри сообщества, сохраняя фиксированные степени узлов.

Другие редактирования Вместо модификации самого графа G можно редактировать некоторое его представление $R(G)$, при условии что оно корректно отражает его признаки. При таком подходе основной задачей становится найти подходящее представление графа и удобные операции преобразования G в $R(G)$ и обратно.

Авторы Мультимасштабного генератора сетей (***Multiscale Network Generation***, MUSKETEER) [132] предлагают применять серии особых операций сжатия-расширения (***coarsening-uncoarsening***) к матрице Лапласа $\mathbf{L}$ графа совместно с редактированием сжатых версий графа. Начиная с исходного графа G серия последовательно сжатых графов $\{G^i\}_{i=1}^k$ получается путем преобразования $\mathbf{L}^{i+1} = (\mathbf{P}^i)^T \mathbf{L}^i \mathbf{P}^i$. Матрица $\mathbf{P}^i$ кодирует связь между узлами графа G^i и узлами его сжатой версии G^{i+1}. Кратко говоря, некоторые узлы объединены в один. Узлы-сиды (центры объединения) графа G^i выбираются на основании их степени, а затем по наличию у них соседей-сидов. Оставшиеся узлы (не сиды) объединяются со своими ближайшими сидами. Пара узлов-объединений в G^{i+1} соединяется ребром, если хотя бы одна пара входящих в них узлов была связана в G^i.

После сжатия до G^k начинается обратный процесс расширения. На каждом шаге текущий граф $\bar{G}^i$ редактируется: удаляются случайно выбранные

[3]Это следствие упомянутого ранее факта эргодичности Марковской цепи. Переключение ребер позволяет получить любой из всевозможных графов с теми же степенями, а значит, остальные характеристики графа могут принимать произвольные допустимые значения. Далее встает вопрос о том, насколько заданная последовательность степеней вершин определяет другие графовые признаки.

ребра, затем добавляется несколько новых; то же самое для узлов. Добавляемый узел имитирует один из существующих, то есть копирует структуру, которая агрегирована в нем в результате объединений. Новое ребро (u, v) добавляется путем случайного выбора начального узла u и выбора конечного v таким образом, чтобы расстояние между ними равнялось d. Число d сэмплируется из эмпирического распределения таких расстояний в G^i: для каждого ребра графа G^i измеряется длина кратчайшего пути, кроме самого ребра, между его концами. Количество новых ребер выбирается так, чтобы приблизительно сохранить распределение степеней. Отредактированная версия графа $\tilde{G}^i$ затем расширяется в $\bar{G}^{i-1}$. Темпы редактирования на каждом уровне являются свободными параметрами модели, контролирующими степень рандомизации и, коэффициент масштабирования графа — если темп добавления элементов превышает темп удаления. Эксперименты показали, что MUSKETEER способен воспроизводить признаки на основе степени вершины — средняя степень, ассортативность, — и расстояний — средние эксцентриситет, расстояние, гармоническое расстояние и центральность по посредничеству (***betweenness centrality***).

1.3.3 Класс предметно-специфичных подходов

Обычно модели случайных графов разрабатываются для простых графов, часто для ориентированных. Некоторые подходы для простых графов адаптируются для случая направленных ребер, реже возникает вопрос поддержки мультиграфов, петель и т.п. В некоторых моделях они возникают в качестве побочного эффекта, как, например, в SKG, где на входе предполагается граф без кратных ребер, а в процессе генерации образуются кратные ребра, которые затем просто заменяются простыми. Иные подходы в принципе не предполагают такие свойства, например, если вероятность ребра зависит от геометрического расстояния между узлами, условие появления петель требует введения дополнительных схем.

Однако на практике тип графа, отличный от простого, может иметь принципиальное значение: двудольные графы аффилированности и авторства; графы с атрибутами, где узлы и ребра могут иметь свои метки и так далее. К

классу предметно-специфичных подходов относятся все методы моделирования предметно-специфических особенностей графов: кратные ребра, петли, ребра с метками, вершины разных типов, вершины, имеющие атрибуты, и т.д.

Здесь рассмотрим только два направления: графы с сообществами, очень популярные в социальной области, и случай взвешенных ребер, также широко распространенный во многих областях [133].

Со структурой сообществ

Сложные сети часто имеют группы более плотно связанных узлов, называемые сообществами. Сообщества возникают не только в социальных сетях, где пользователи объединяются в группы явно, но и в других графовых доменах. В сетях взаимодействия белков сообщества соответствуют белкам со схожей функциональностью, в графах цитирования узлы группируются по тематикам исследований. Структура сообществ отражает строение сети на среднем масштабе и имеет свои собственные паттерны. В последние два десятилетия возник интерес к автоматическим методам поиска сообществ и, как следствие, к разработке соответствующих моделей случайных графов со структурой сообществ.

В этом разделе рассмотрим методы к генерации структуры сообществ в графе в явном виде. Основной подход состоит в том, чтобы каждому узлу графа задать метку, показывающую каким сообществам он принадлежит. Далее опишем модели, использующие такой подход. Хотя эта концепция представлена как одна категория в таксономии, подходы моделирования графов на ее основе легко разделяются на генеративные и управляемые признаками[4], в полном соответствии с соответствующими классами.

Генеративные подходы Первым шагом является задание меток сообществ для узлов графа. Затем следует стандартная генеративная схема, где вероятность ребра p_{ij} теперь зависит от меток сообществ c_i и c_j.

Простейшие подходы основаны на генерации группы ER-графов с различными вероятностями появления ребра и связывании их между собой —

[4]Это не две подкатегории категории ‘структура сообществ’, потому что она касается методов *моделирования сообществ*, в то время как разделение на ‘генеративные’ и ‘управляемые признаками’ относится к подходам *генерации графа*.

такова модель Гирвана-Ньюмана (Girvan-Newman) [134], а также с пересечением групп [135] и с наличием вложенности [136]. В модели BTER [137] к группе ER-графов добавляется свойство произвольного распределения степеней: после соединения узлов внутри блоков в соответствии с моделью ER, “избыточные” степени узла (равная желаемой степени d_i минус реальная степень в пределах его блока, если разница положительна) используются для соединения между блоками с использованием модели ожидаемой степени (Чунг-Лу).

Для более интеллектуального распределения узлов по сообществам, в модели Графа принадлежности сообществам (***community-affiliation graph model***, AGM) [33] рассматривается двудольный граф $B(N, C, M)$, ребра которого M задают, к каким сообществам из C принадлежит каждый из узлов N. Определенные для каждого сообщества $c \in C$ вероятности $\{p_c\}$ задают вероятности ребер $p_{ij} = 1 - \prod_{c \in Z_{ij}} (1 - p_c)$, где Z_{ij} — множество общих сообществ для узлов i и j. Эта модель обеспечивает важные свойства реальных сообществ: вероятность ребра p_{ij} увеличивается с ростом Z_{ij}; плотность ребер выше в пересечении сообществ; количество ребер в сообществе растет суперлинейно с его размером; хабы сообщества, скорее всего, расположены на пересечениях сообществ. На практике граф принадлежности строится с использованием конфигурационной модели, после задания последовательности членства узлов (кто в скольких сообществах состоит) и последовательности размеров сообществ.

Бóльшую гибкость предоставляет LFR бенчмарк [52]. Авторы вводят параметр топологического смешивания μ, который контролирует относительную плотность ребер внутри сообществ: внутренняя степень узла d_i^{in} определяется как число его соседей, имеющих с ним хотя бы одно общее сообщество, и ожидается, что она будет равна определенной доли от его обычной степени $d_i^{in} = (1 - \mu)d_i$. Для достижения этих условий после формирования связей внутри сообществ применяется переключение ребер.

Управляемые признаками подходы В управляемых признаками подходах метки сообществ, наоборот, определяются на основе заданного графа.

Метод SBM [126] может быть использован для подгона к некоторой реальной сети без известной структуры сообществ. Число групп Q, априорные вероятности групп p_q и матрица вероятностей связи между группами $\mathbf{P}$ оцениваются с помощью EM-алгоритма в рамках ERGM [126]. Как альтернатива, модель Кластера со скрытой позицией (***latent position cluster model***) [106]

использует идею ненаблюдаемого “социального пространства”. Сообщества узлов представлены смесью многомерных нормальных распределений векторов в этом пространстве, вероятность ребра зависит от евклидова расстояния $p_{ij} \sim e^{-\beta dist(i,j)}$. Затем параметры оцениваются с помощью максимизации правдоподобия или MCMC сэмплирования.

Другой способ — найти сообщества во входном графе с помощью какого-нибудь метода поиска сообществ, а затем имитировать их в случайном графе. Метод ReCoN [56] просто копирует найденные сообщества вместе со всем графом и применяет переключение ребер внутри них и между ними, чтобы соединить реплики. Поэтому при масштабировании графа в k раз количество сообществ растет пропорционально, а распределение их размеров остается постоянным.

С весами на ребрах

Веса ребер естественным образом возникают в сложных сетях: они могут выражать силу связи в социальной сети, величину потока в метаболической реакции, меру коэкспрессии генов и т. д. Кратное ребро в графе можно представить как ребро с целочисленным весом. Многие характеристики и понятия обобщаются на взвешенные графы, включая длину кратчайшего пути, коэффициент кластеризации, модулярность. Взвешенность графов приносит новые аспекты в существующие задачи на сетях, такие как поиск сообществ [138].

Один из способов смоделировать взвешенные ребро — это сгенерировать кратные ребра и интерпретировать их как взвешенные. В RTG [47], основанном на случайной последовательности символов, каждое следующее повторение той же пары слов увеличивает вес соответствующего ребра. Это приводит к степенному закону, связывающему мощность узла с его степенью: $s_i \sim d_i^{\beta}$. RTG также обеспечивает степенной закон суммарного веса $W(t) \sim m(t)$ и самоподобие увеличения веса.

В LFR бенчмарке [52] веса ребер назначаются так, что ожидаемая мощность узла равна $s_i \sim d_i^{\beta}$, а внутренняя мощность s_i^{in} (мощность, учитывающая только соседей в том же сообществе) контролируется параметром смешивания μ: $s_i^{in} = (1 - \mu)s_i$. Это достигается с помощью жадного алгоритма, который

итеративно изменяет вес ребер w_{ij}, чтобы минимизировать квадратичную дисперсию всех s_i, s_i^{in} и $s_i - s_i^{in}$, суммарно по всем узлам.

1.4 Обсуждение

В этой части обсудим представленную в предыдущей части таксономию и опишем, как рассмотренные подходы работают в различных приложениях моделей случайных графов.

1.4.1 Комментарии к таксономии

Отношение между моделями и подходами При попытке построить таксономию *моделей* случайных графов, получилось бы гораздо более ветвящееся дерево, где похожие подходы повторялись бы много раз. Более того, трудно классифицировать сами модели, поскольку они часто смешивают в себе различные подходы. По этой же причине некоторые модели появляются одновременно в нескольких категориях таксономии.

Таксономия представляет и классифицирует основные походы, используемые в моделях, рассмотрим теперь как именно модели комбинируют описанные подходы. Чтобы ответить на вопрос в таблице 3 сравниваются известные модели на основе двух или более подходов. Можно заметить, что в моделях используется до трех-четырех подходов в различных сочетаниях. Например, в модели Forest Fire [31] используются сразу три генеративных механизма: копирование входящих и исходящих ребер выбранного узла, локальные правила при обработке еще не посещенных соседей и принцип рекурсии при повторении одной и той же процедуры на каждом узле.

Несмотря на небольшой размер таблицы, были вычислены корреляции между столбцами, чтобы выяснить, какие подходы хорошо совместимы друг с другом (высокая корреляция), а какие нет (низкая корреляция). На рисунке 1.2 представлена матрица корреляций категорий с наибольшими и наименьшими корреляциями. Видно, что принцип копирования хорошо сочетается с ‘другими

Таблица 3 — Модели, комбинирующие несколько подходов (отметка '✓' в соответствующей клетке): генеративные (синий), управляемые признаками (зеленый), предметно-специфичные (красный) подходы. Разработанный метод ERGG-dwc (см. главу 2) также включен в таблицу для полноты сравнения.

	Классические подходы	Предпочтит. присоед. (PA)	Копирование	Другие локальные правила	Рекурсивные процедуры	Геометрические подходы	Метки узлов	Аналитические подходы	Оценка паарметров	Экспоненциальные (ERGM)	Переключение ребер	Другое редактирование	Со структурой сообществ	С весами на ребрах
R-MAT [42]					✓		✓							
SKG [44]					✓		✓		✓					
MFNG [45]					✓		✓		✓					
RTG [47]					✓		✓							✓
Forest Fire [31]			✓	✓	✓									
GScaler [49]			✓	✓										
MUSKETEER [132]			✓	✓	✓							✓		
LPC [106]						✓	✓		✓				✓	
Girvan-Newman [134]	✓						✓						✓	
BTER [137]	✓						✓	✓					✓	
AGM [33]							✓	✓					✓	
LFR [52]								✓			✓		✓	✓
ReCoN [56]			✓								✓		✓	
SBM [126]	✓								✓	✓			✓	
ERGG [6]			✓			✓			✓			✓	✓	✓
GIRG [111]						✓		✓						
Dot-product [46]						✓	✓							
BRITE [107]		✓				✓								

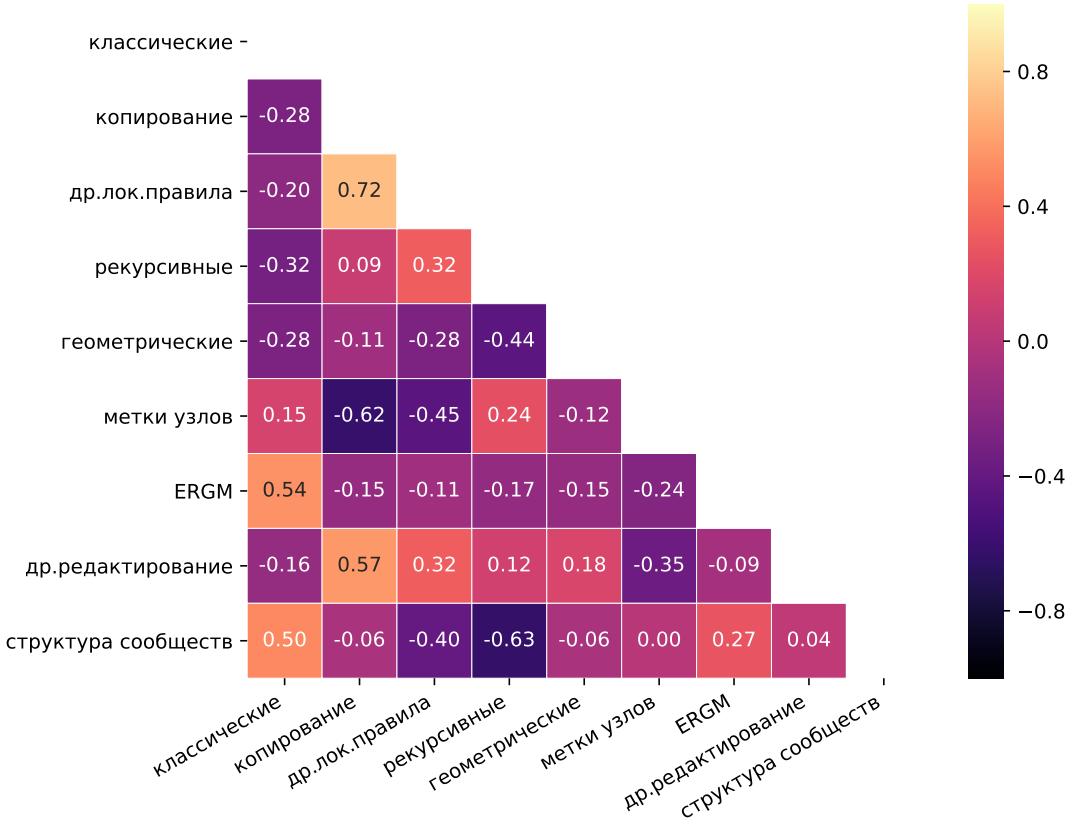

Рисунок 1.2 — Матрица корреляций подходов из разных категорий таксономии. Рассчитано на основе таблицы 3. Категории с корреляциями около 0 исключены.

локальными правилами' (0,72) и 'другим редактированием' (0,57), что может соответствовать эволюционному принципу копирования с мутациями. 'Классические подходы' часто встречаются вместе с 'ERGM' (0.54) и подходов 'со структурой сообществ' (0.50).

Находясь на противоположном конце списка, самые низкие корреляции могут указывать на то, что подходы несовместимы или по какой-то причине не использовались вместе. 'Структура сообществ' плохо сочетается с 'рекурсивными' подходами (-0,63), что кажется странным ввиду известной связи между фрактальной (рекурсивной) структурой графа и наличием иерархической структуры сообществ. Однако, объяснением может быть то, что существующие рекурсивные методы (SKG, RTG, Forest Fire) не способны сгенерировать явную структуры сообществ с желаемыми свойствами. Возможная причина этому в сложности настройки модели, поскольку необходимо подгонять и графовые характеристики, и характеристики сообществ. Процедуры подгона параметров в SKG и MFNG достаточно сложны даже без учета свойств сообществ. Следующие пары с самой низкой корреляцией — 'метки узлов' с 'копированием' и

‘другими локальными правилами’. Объяснить здесь несочетаемость подходов сложнее; возможно, такие модели были упущены при обзоре, либо еще ждут своего часа.

Выбор категорий и полнота таксономии Количество классов в таксономии выбрано несколько произвольно. В то время как классы генеративных и управляемых признаками подходов предназначены для охвата всех моделей для простых и ориентированных графов, категории низших уровней по-прежнему содержат многие подходы и могут быть классифицированы дальше. Например, рекурсивные подходы могут быть детерминированными или стохастическими, геометрические подходы могут содержать подкатегорию ‘гиперболическая геометрия’ из-за большого количества работ, посвященных гиперболическим случайным графам. С другой стороны, графовые модели часто используют множество различных эвристик, которые трудно классифицировать и которые могут сформировать свои собственные подкатегории. Примером служит раздел ‘другие локальные правила’, где можно различить правила выбора соседей или триадного формирования. Поэтому были выбраны три уровня вложенности в качестве компромисса.

Несколько категорий, а именно ‘другие локальные правила’, ‘другое редактирование’ и ‘метки узлов’ являются контейнерами для подходов, которых нет в других категориях общего класса. Например, ‘метки узлов’ соответствует методам, основанным на атрибутах узла, которые не являются геометрическими, — таких подходов было обнаружено не так много.

Говоря о полноте таксономии, описанные три класса задуманы для охвата всех основных направлений моделирования случайных графов. Классы генеративных и управляемых признаками подходов описаны подробно и хорошо структурированы. Предполагается, что они отражают текущую картину в моделировании простых и ориентированных графов. В третий класс предметно-специфичных подходов вошло только два самых популярных случая, в то время как все остальное разнообразие типов графов осталось за рамками рассмотрения. По этой причине состав категорий третьего класса нельзя считать исчерпывающим.

Новые подходы, а также пропущенные в обзоре, должны попасть в одну из существующих (под-)категорий или образовать новую в одном из трех классов. Что касается появления новых концепций моделирования, то кажется,

что класс генеративных подходов исчерпан. Основные механизмы формирования сети уже изобретены и описаны в литературе, дальнейший прогресс ожидается от управляемых признаками подходов. Основные проблемы моделирования графов касаются лучшего подгона модели к заданному графу, создания быстрых и простых процедур сэмплирования графов. Перспективными направлениями здесь могут быть: методы редактирования графа на основе обучения представления графа; разработка методов генерации очень больших графов с миллиардами ребер.

Есть надежда, что проведенные классификация и анализ подходов, доказавших свою работоспособность при моделировании графов, будет способствовать разработке новых моделей. Однако, вспомним, что создание новой модели это не только комбинирование известных концепций. Обычно модель разрабатывается для решения некоторой практической задачи. В следующем разделе рассмотрим, какие из рассмотренных подходов являются успешными в известных приложениях.

1.4.2 Приложения моделей случайных графов

В этом разделе выделим и рассмотрим подробно шесть направлений, в которых модели случайных графов находят свое применение: понимание сетей, анализ сетей, экстраполяция, тестирование, нулевые модели и рандомизация. Далее покажем, как идеи, описанные в предыдущем разделе, применяются для решения задач, возникающих в этих областях.

Понимание сетей

Открытие новых топологических паттернов в реальных сетях вызвало попытки объяснить их возникновение. Если предполагаемый генеративный механизм порождает графы с нужными характеристиками, он может быть основой реальных процессов. Поэтому все идеи класса генеративных подходов являются потенциальными объяснениями формирования сетей. Правило

предпочтительного присоединения, которое в сочетании с добавлением узлов порождает безмасштабную топологию, имеет интуитивные интерпретации. Новый человек, присоединяющийся к социальной сети, с большей вероятностью устанавливает связь с тем, у кого уже много контактов, а новая веб-страница с большей вероятностью будет ссылаться на страницу со многими ссылками, то же справедливо и для цитирования научных работ. Другой показательный пример — принцип копирования. Копирование ребер узла соответствует переиспользованию ссылок цитирования в графах цитирования и веб-графе, дублированию генов в биологических сетях и так далее. Присоединение на основе атрибутов узла согласуется с принципом связыванием пользователей на основе схожести интересов. Рекурсивная процедура тесно связана с самоподобием, и ее работоспособность может указывать на то, что одни и те же простые законы определяют формирование сетей на разных масштабах. Работы по получению топологий из задач оптимизации свидетельствуют о том, что определенные паттерны сетевых структур являются оптимальными при некоторых условиях.

Управляемые признаками подходы, такие как ERGM, также способствуют пониманию сложных сетей по следующим причинам. Сама идея ERGM строится на проверке того, как различные графовые статистики могут объяснить наблюдаемую структуру. Та модель, которая лучше всего подходит (в статистическом смысле) к реальной сети, показывает, какие признаки можно считать наиболее важными для объяснения сетевой архитектуры [139]. Кроме того, стохастическая модель может отражать не только закономерности в графе, но и вариабельность его свойств. При подгонке к реальному графу, модель дает картину распределения возможных наблюдаемых реализаций [140]. Наконец, некоторые признаки могут иметь несколько объяснений, например, высокий коэффициент кластеризации вызван гомофилией, а также мог возникнуть в результате самоорганизации структурных эффектов. Модель, объединяющая оба эффекта, помогает количественно оценить вклад обеих альтернатив [140].

Анализ сетей

Простые модели, такие как ER, с вероятностью ребра, зависящей от размера графа $p(n)$, были детально изучены на предмет эволюции их поведения, то есть когда n стремится к бесконечности. Были обнаружены различные виды фазовых переходов, например, появление гигантской связной компоненты, появление треугольников [66]. Это побудило изучать надежность сетей, таких как Интернет, сети коммуникаций и т. д., с точки зрения устойчивости к атакам, таким как случайное или намеренное удаление узлов или ребер [141].

Для анализа процессов, происходящих в сетях — распространение информации или эпидемий в социальных сетях, потоки в транспортных сетях и Интернете, экономические транзакции и т. д. — проводится имитационное моделирование. Поскольку топология сети взаимодействий имеет решающее значение для динамики процессов, необходимы реалистичные модели графов [142]. Согласно обзору публикаций в JASSS (1998 – 2015), большинство работ на самом деле используют очень простые модели: регулярные решетки, случайные графы (ER), сети малого мира или безмасштабные сети [72]. Фредерик Амблард и соавторы (Frédéric Amblard et al) предлагают три перспективы для моделей социальных сетей:

- Абстрактные модели, такие как Forest Fire [31], воспроизводящие множество известных свойств. Они соответствуют описанному классу генеративных подходов. Такие бенчмарки, как LFR [52], выглядят обещающе с использованием структуры сообществ для моделирования популяций.
- Модели с возможностью подгона к данному графу, например, ERGM и SKG [44]. Эти идеи как раз представлены в категории ‘оптимизация функции’ 1.3.2.
- Подходы, основанные на правилах, где генерируется популяция с использованием байесовских правил, а затем строится граф путем определения узлов в соответствии с правилами, как, например, в работе С. Тириота и Ж.-Д. Канта (Thiriot S., Kant J.-D.) [143]. Это направление отражено в категориях ‘скрытые атрибуты’ и ‘локальные правила’.

Экстраполяция

Интернет, социальные и другие сети со временем увеличиваются в размерах, поднимая такие вопросы, как: будет ли интернет-протокол работать через 5 лет, или как будет выглядеть граф Facebook в будущем? Отсюда возникает необходимость в алгоритмах для масштабирования существующего графа до большего размера. Более того, названные графы уже сами большие, порядка миллиарда узлов, что накладывает жесткие ограничения на сложность алгоритмов и размер используемой ими памяти. В общем случае можно выделить два подхода: генеративный процесс, реализующий нужные признаки и создающий графы произвольного размера, или метод масштабирования заданного графа.

Из класса генеративных подходов рекурсивные подходы, методы скрытых атрибутов и идеи копирования показали здесь наибольший успех. Рекурсивное умножение матриц (SKG [44], MFNG [45], TrillionG [57]) является интересным решением, однако требует наличия быстрого алгоритма для построения графа, так как время генерации $O(n^2)$ может быть неприемлемым в случае очень больших графов. Кроме того, непрактичной является сильная дискретность размеров получающихся графов: $n = n_0^k$. Принцип копирования применяется на разных уровнях: от репликации отдельных узлов с половинами ребер (GScaler [49]) до целых сообществ и самого графа (ReCoN [56]). Для воспроизведения графовых признаков обычно требуется либо подгонка параметров модели (SKG, MFNG), либо специальные эвристики для контроля процесса генерации на соответствие графа требуемым свойствам (GScaler). Тем не менее, другие подходы из категории ‘скрытые атрибуты’, такие как гиперболические или графы скалярного произведения, являются многообещающими кандидатами, которым, возможно, лишь не хватает процедур подгона.

Альтернативный способ масштабирования графа предполагает его модификацию, приводящую к увеличению размера. Например, серия сжатий, а затем более длинная серия расширений приводит к увеличенной версии исходного графа (MUSKETEER [132]). В случае более простого алгоритма (ReCoN), граф с сообществами мультиплицируется, а ребра затем перемешиваются. Заметим, что оба приведенных примера в конечном счете основаны на идее копирования.

Тестирование

Существует широкий спектр инструментов и алгоритмов майнинга графов, начиная от расчета характеристик (диаметр, модулярность) и заканчивая выводом топологии (предсказание ребра) [9; 17; 63], активно развивающихся в различных сетевых доменах. Надежное тестирование этих методов затруднено или даже невозможно из-за отсутствия подходящих данных. Например, для проверки статистической значимости требуется репрезентативный набор графов, обеспечивающий достаточную вариабельность по их характеристикам, в то время как для тестирования масштабируемости требуется серия похожих графов разного размера [6]. Решением могут служить эталонные случайные графы, адаптированные для конкретных случаев (LFR).

Репрезентативность выборки имеет первостепенное значение для проверки статистической значимости. Идеальным решением могут быть ERGM, поскольку они определяют вероятностное пространство графа с теоретически любыми условиями на его статистики. К сожалению, на практике современные подходы имеют серьезные вычислительные ограничения. Управляемое переключение ребер в рамках MCMC сэмплирования позволяет получить графы с требуемыми диапазонами значений характеристик и даже с заданными ограничениями на распределения характеристик [55], хотя и при неизменной последовательности степеней. Менее строгие в смысле воспроизведения признаков, но обеспечивающие управляемую вариабельность сгенерированных графов, дают рекурсивные модели, например, SKG с зависимым сэмплированием ребер (***Mixed Kroneker product graph model***) [99].

Для тестирования масштабируемости необходим генератор графов контролируемого размера. Если к тому же требуется сходство с заданным графом, задача сводится к задаче экстраполяции, возможно, с коэффициентом масштабирования меньше 1. Если сходство с заданным графов не важно, то подойдет любой алгоритм из класса генеративных подходов, способный воспроизводить нужные признаки. Проблемы вычислительной сложности и потребления памяти возникают здесь, когда размер графов становится очень большим: алгоритмы квадратичного времени работы нецелесообразны при размерах более миллиона узлов. Приемлемой будет сложность порядка $O(n \log n)$ или линейная по количеству ребер $O(m)$, что означает, что некоторые алгоритмы,

основанные, например, на попарном переборе узлов, исключаются из рассмотрения. Существующие решения адаптируют имеющиеся известные алгоритмы (ROLL [144] — ускорение PA), разрабатываются их аппроксимации и распределенные версии [73]. Распределенные алгоритмы позволяют генерировать графы с миллиардами узлов (CKB [53]) и триллионами ребер, как например Darwini [145].

Большой интерес представляют случайные графы с сообществами, особенно для эмуляции социальных сетей. Аналитический подход используется для реализации богатого набора параметров, описывающих распределение степеней и свойства сообществ. Конфигурационная модель для реализации распределения степеней и переключение ребер для рандомизации используется в LFR бенчмарке [52]. Распределенные алгоритмы [53; 145] здесь также актуальны.

Нулевые модели

Нулевые модели служат для аккуратной проверки гипотез, предоставляя выборку графов с хорошо контролируемыми свойствами. В контексте сложных сетей с помощью нулевой модели можно выяснить, какие структурные паттерны характерны для заданной сети по сравнению с некоторым средним распределением. Нулевая модель позволяет количественно определить, насколько сеть отличается от случайной (нулевая гипотеза), например, оценить статистическую значимость доли двунаправленных ребер или наблюдаемого количества общих соседей для двух узлов [146].

Наиболее популярной нулевой моделью графа является конфигурационная модель, которая определяет равномерное распределение по всем графам с одной и той же последовательностью степеней. Она широко используется для изучения сетевых паттернов в социологии, экологии, системной биологии и других областях (подробнее можно найти в обзоре [120]). Например, обнаружение графовых мотивов как статистически значимых подграфов [147] или в качестве нулевой модели, относительно которой определяется модулярность сообществ [148]. Равномерное сэмплирование графов с фиксированной степенной последовательностью эффективно выполняется с помощью методов MCMC сэмплинга, основанных на переключении пар ребер [120].

Альтернативной нулевой моделью может быть модель Чунг-Лу, которая задает ожидаемые степени узла d_i вместо фиксированных, определяя вероятность ребра как $p_{ij} \sim d_i d_j$. Ее использование связано с методом поиска сообществ на основе SBM алгоритмов [149].

Рандомизация

Предоставление графовых данных в общий доступ может быть проблематичным, если эти данные содержат приватную информацию. Поэтому необходимо рандомизировать граф таким образом, чтобы избежать утечки конфиденциальной информации и в то же время сохранить его важные структурные свойства. Было показано, что простое перенумерование идентификаторов узлов не гарантирует безопасности данных [4]. Возможные атаки включают активные атаки, — например, предварительное введение группы узлов (***sybil attack***) с определенной узнаваемой конфигурацией ребер в исходный граф и последующее ее обнаружение после рандомизации, — и пассивные атаки, когда пользователь может деанонимизировать себя, используя знание близлежащей топологии сети или вспомогательную информацию, например, агрегированные социальные сети [150].

Как правило, модели случайных графов могут помочь анонимизации двумя способами: генерация нового графа, тщательно повторяющего признаки исходного, или рандомизация исходного с помощью методов редактирования графа. Генеративные алгоритмы сталкиваются с проблемой подгонки, но они потенциально более надежны, поскольку изначальное отображение между исходными узлами и узлами сгенерированного графа отсутствует. Анонимизация соответствует задаче экстраполяции с коэффициентом масштабирования 1, поэтому здесь могут быть использованы те же методы.

На практике все же более распространенным методом является рандомизация через редактирование графов. Переключение ребер — относительно быстрый и простой в реализации метод, хотя при непосредственном применении нарушает все признаки, кроме последовательности степеней. Тем не менее, переключение ребер при некоторых условиях, например, сохранении выбранных спектральных характеристик в сочетании со случайным добавлением и удале-

нием ребер, как утверждается, защищает конфиденциальность связей [151]. Для моделирования паттернов также используются dK-графы. Найденные dK-распределения затем возмущаются таким образом, чтобы обеспечить свойства дифференциальной приватности (***differential privacy***) [152], и, наконец, генерируется новый граф в соответствии с новыми dK-распределениями [81].

Согласно обзору Шоулинь Цзи и соавторов (Shouling Ji et al) [153] другие подходы рандомизации включают k-анонимность (где каждый узел в публикуемом графе имеет $k-1$ двойников), анонимность на основе кластера (структура графа сохраняется на уровне кластеров, игнорируя их внутреннюю конфигурацию), анонимизацию на основе случайного блуждания (ребро (u,v) заменяется на (u,w), где w — конечная точка случайного блуждания из u).

1.5 Генераторы графов, похожих на данный

В работе решается задача генерации графа, похожего на данный. Такие методы относятся к классу управляемых признаками подходов, поскольку требуют процедуры подгона параметров модели. Поэтому в этом разделе рассмотрим подробнее наиболее релевантные модели из упомянутых ранее: способные автоматически обучаться на данном графе и позволяющие генерировать графы контролируемого размера.

Ю. Лисковец и соавторы [44] предлагают рекурсивную генеративную модель SKG, основанную на Кронекеровском произведении графов, а также алгоритм подгонки параметров этой модели к реальному графу. Основная идея состоит в том, чтобы рекурсивно задать структуру графа, по аналогии с известным свойством самоподобия реальных безмасштабных сетей [154]. Вероятностная матрица смежности графа представляет собой k-ю степень Кронекера (т.е. тензорное произведение $\otimes$) некоторой малой матрицы-инициатора A^1_{ij}: $A^k_{ij} = (A^1_{ij})^{\otimes k}$. В этом случае граф имеет $n = n_0^k$ узлов и $m = m_0^k$ ребер. Аналитически можно показать, что такие графы имеют набор свойств, присущих реальным сетям: мультиномиальные распределение степеней, распределения собственных значений и собственных векторов, степенной закон уплотнения, малый диаметр.

Чтобы согласовать параметры модели $\Theta = \{A_{ij}^1\}$ с реальным графом G, логарифм вероятности $\log P(G|\Theta)$ оптимизируется с помощью градиентного спуска. Основная задача здесь состоит в том, чтобы учесть все возможные $n!$ перестановок узлов при сопоставлении A_{ij}^k с матрицей смежности G. Было предложено эффективное решение с применением сэмплирования по Метрополису перестановок вершин и аппроксимации функции градиента правдоподобия, используя рекурсивное определение матрицы A^k.

Другой подход, использующий идею рекурсивной структуры графа, был предложен Г. Палла и соавторами (G. Palla et al) [45]. Как и в SKG, в генераторе мультифрактальных сетей (MFNG) вероятность ребра задается рекурсивно определенной матрицей, но, кроме того, узлы задаются с помощью рекурсивно определенных категорий. А именно, отрезок [0,1] делится на m различных подинтервалов (определяемых дополнительными $m-1$ параметрами), каждый из которых снова итеративно разделяется с одинаковыми соотношениями k раз, таким образом устанавливая категории. Затем узлы графа равномерно выбираются в [0,1]. Подгонка к реальному графу может быть выполнена методом моментов как задача минимизации отклонения набора целевых характеристик от их ожидаемых значений [123]. Сильной стороной этой модели является то, что любая статистика, которая может быть выражена как функция на подмножестве ребер (число ребер, клик, звезд и т. п., включая распределения подграфов), аналитически выражается через параметры модели и, таким образом, может использоваться для подгонки.

SKG и MFNG, два схожих подхода на основе рекурсивной процедуры, позволяют генерировать графы разного размера. При этом, если MFNG требует явно указывать признаки графа для подгона, SKG имеет специальную процедуру Kronfit, принимающую на вход сам граф.

В серии работ К. Л. Стаут и др. (Christian L Staudt et al) [56] представлен метод ReCoN, который копирует входной граф с сообществами и рандомизирует его с помощью переключения ребер. Создается k копий исходного графа, переключение ребер применяется внутри каждого сообщества, а затем среди оставшихся ребер, избегая образования новых ребер внутри сообществ. Показана воспроизводимость значений следующего набора характеристик: средняя и максимальная степень, коэффициент Джини распределения степеней вершин, средний коэффициент кластеризации, диаметр, количество компонент связности и количество сообществ.

В методе Gscaler [49] входной граф $G = (N,E)$ разбивается на маленькие фрагменты, которые реплицируются и затем соединяются обратно в новый граф $G' = (N',E')$. Процесс масштабирования и соединения выполняется в соответствии с функциями корреляций входящих/исходящих степеней узлов и ребер. При декомпозиции каждый узел i порождает два фрагмента: узел с входящими в него ребрами и тот же узел с исходящими ребрами. Процесс масштабирования включает в себя создание копий всех кусков таким образом, чтобы общее число узлов стало равно n', число ребер стало m' — двумя параметрами масштабирования, указанными пользователем. На третьем этапе пары масштабированных кусков соединяются в узлы так, чтобы сохранить распределение бистепеней узлов $f_{bi}(d_{in}, d_{out})$, которое определяет корреляцию входящих/исходящих степеней узлов. Последний шаг состоит в соединении концов ребер с узлами, сохраняя функцию корреляции ребер $f_{corr}(\alpha_1, \alpha_2)$, где α_i — бистепень узла (d_i^{in}, d_i^{out}).

Эксперименты показали, что Gscaler хорошо воспроизводит распределения входящих/исходящих степеней и бистепеней, а также похожесть генерирумых графов по эффективному диаметру, коэффициенту кластеризации, отношению размеров наибольших сильно связных компонент и среднему минимальному пути.

1.5.1 Случайные графы скалярного произведения

В дополнение рассмотрим одну модель случайных графов, которая не была разработана для имитации реальных графов, но тесно связана с разработанным подходом ERGG (глава 2). В модели графов скалярного произведения [155] вероятность ребра между узлами i и j определяется некоторой функцией от скалярного произведения ассоциированных им векторов: $p_{ij} = f(\vec{u}_i \cdot \vec{u}_j)$.

В простой генеративной модели n векторов $\vec{u}_i$ сэмплируются из $\mathcal{U}^{\alpha}[0,1]$ (степень α равномерного распределения), а соответствующие узлы связываются с вероятностью $p_{ij} = \vec{u}_i \cdot \vec{u}_j$. Такая модель позволяет генерировать графы со степенным распределением степеней, малым диаметром и высоким коэффициентом кластеризации. Имеется естественное обобщение на направленный

случай [156], где $\vec{u}$ и $\vec{v}$ выбирается независимо из вероятностных распределений $\mathcal{U}$, $\mathcal{V}$ соответственно.

В работе также была упомянута идея оценить распределение $\mathcal{U}$ для соответствия некоторому реальному (взвешенному неориентированному) графу G методом ядерной оценки плотности (***Kernel Density Estimators***) с целью генерации графов, подобных G. Учитывая, что задача нахождения представления данного графа как модели скалярного произведения уже рассматривалась в [157], это делает ее тесно связанной с подходом ERGG.

1.6 Выводы к первой главе

В этой главе был представлен новый взгляд на подходы к моделированию случайных графов. Были собраны и проанализированы работы в этой области, извлечены основные принципы, использованные в них, и построена иерархическая таксономия подходов моделирования графов (рисунок 1.1). Первые два из трех классов верхнего уровня соответствуют двум принципам моделирования. Класс генеративных подходов описывает изобретенные процедуры генерации графов, качественно показывающие известные графовые признаки. Класс управляемых признаками подходов касается второй стороны задачи моделирования, а именно, количественной подгонки модели графа под заданные условия. Третий класс, предметно-специфичных подходов, относится к способам моделирования различных типов графов, отличных от простых и направленных, в частности, графов со структурой сообществ и взвешенными ребрами. Каждый класс содержит категории и подкатегории, отражающие более детальную классификацию идей.

Многие модели графов часто комбинируют в себе несколько подходов, что отражено в таблице 3. Выделение отдельных используемых подходов позволяет лучше ориентироваться во множестве моделей графов, а также должно помочь при разработке новых моделей в будущем. Также было проанализировано использование описанных подходов в известных приложениях случайных графов.

На момент разработки предлагаемого в работе подхода единственными подходящими методами, которые автоматически выполняют подгонку к графу,

являлись SKG со специальной функцией Kronfit, ReCoN и Gscaler. Однако они обладают рядом недостатков.

Во многих исследованиях у SKG обнаружено много недостатков в генерируемых графах, хотя и частично решенных позднее в его расширениях. Например, появление осцилляций в распределении степеней, высокая доля изолированных вершин (что влияет, например, на ожидаемую среднюю степень), маленькие ядерные числа [98], слишком низкий коэффициент кластеризации [37]; наконец, размер сгенерированного графа может быть только степенью целого числа. Модель SKG также не может обеспечить достаточную вариабельность в сгенерированных графах [158]. Кроме того, она не способна воспроизвести структуру сообществ [56], и генерировать взвешенные ребра.

Метод ReCoN использует переключение ребер для рандомизации, которое, как было сказано в главе 1, нарушает важные признаки, в частности, 3-GP (это следует из определения графового мотива, см. [20]). Кроме того, он не позволяет использовать произвольный коэффициент масштабирования $x \notin \mathbb{N}$.

Главный недостаток метода Gscaler в отсутствии поддержки структуры сообществ и взвешенных ребер. Кроме того, авторы проводили эксперименты только на двух реальных графах из социального домена, что не дает достаточной надежности их результатов.

Таким образом, существующие модели ориентированных случайных графов, похожих на данный, с автоматическим обучением не удовлетворяют остальным поставленным в работе требованиям (см. 2.1), а именно:

– произвольный размер генерируемых графов,
– поддержка структуры сообществ и взвешенных ребер.

В следующей главе представлен новый подход к генерации случайных графов, основанный на идее вложения (эмбеддинга) графа в векторное пространство. Предложенный в рамках подхода метод ERGG-dwc позволяет генерировать направленные случайные графы контролируемого размера, похожие на данный, и поддерживает структуру сообществ и взвешенные ребра.

Глава 2. Генерация графов, похожих на данный. Подход на основе вложения графа и алгоритм ERGG-dwc

Надежная оценка качества инструментов интеллектуального анализа графов подразумевает проверку статистической значимости и масштабируемости используемых методов. Обычно это достигается путем выбора нескольких графов разного размера из разных доменов. Однако, свойства графа и, как следствие, результаты оценки могут существенно различаться от одного домена к другому. Отсюда вытекает необходимость агрегирования результатов по множеству графов в каждом домене.

В этой главе представлен подход, позволяющий автоматически извлекать признаки ориентированного графа из любого домена и генерировать похожие графы, масштабируя размер исходного графа на вещественный коэффициент. Создание нескольких графов одинакового размера позволяет оценивать статистическую значимость, а контролируемый размер графа делает возможным оценку масштабируемости алгоритмов. Предложенный подход основан на вложении исходного графа в низкоразмерное пространство, тем самым кодируя признаки графа в наборе векторов узлов — Embedding based Random Graph Generation, ERGG.

В рамках подхода ERGG предложен метод генерации случайных графов контролируемого размера, похожих на данный. Метод поддерживает направленные взвешенные графы со структурой сообществ — ERGG-dwc (от англ. ***directed, weighted, communities***). Метод ориентирован на обеспечение изменчивости синтетических графов, сохраняя распределение степеней и распределение подграфов размера 3 близкими к исходному графу.

Основные результаты работы опубликованы в статье [6].

2.1 Описание и постановка задачи

Адекватная модель случайного графа должна отражать основные свойства моделируемой реальной сети. Согласно опыту изучения литературы,

распространенная схема моделирования и генерации случайных графов включает в себя следующие шаги:

1. Изучить статистические особенности реальных графов: распределения, зависимости, диапазоны параметров;
2. Выбрать признаки для моделирования, например, распределение степеней вершин, коэффициент кластеризации, диаметр, распределение подграфов и т. д.;
3. Определить вероятностное пространство по всем возможным графам с выбранными характеристиками, что обычно достигается путем определения параметризованного процесса генерации графов;
4. Сэмплировать случайные графы из заданного пространства.

Однако, этот подход сталкивается с двумя серьезными проблемами. Первая проблема заключается в том, что конкретные признаки отличаются от графа к графу и от домена к домену. Например, многие социальные сети демонстрируют степенной закон в распределении степени узла, в то время как его показатель степени различается для разных сетей [159]. Такие параметры приходится специально извлекать из данной сети и использовать для настройки модели.

Вторая проблема — неопределенность в точном списке свойств, которые нужно воспроизвести. Заранее неизвестно, какие свойства важны для моделирования, а какие нет. Этот факт стимулирует разработку все большего количества моделей графов для конкретных приложений, которые не подходят для других задач, и, таким образом, усложняет выбор подходящей модели для конкретной задачи.

Однако, указанные препятствия можно преодолеть путем автоматического извлечения признаков из реальных данных. То есть, шаги 1 – 3 вышеупомянутой схемы заменяются этапом автоматического обучения модели для данного графа.

Многие графовые домены ориентированы естественным образом — пищевые цепочки, графы цитирований, звонков и т. д, то есть, ориентация ребер имеет принципиальное значение. Кроме того, как было показано в предыдущей главе, дополнительную важную информацию о моделируемом объекте несут веса на ребрах графа и структура сообществ. При этом, указанные свойства нельзя рассматривать отдельно друг от друга, так как, например, вес ребра, выражающий силу связи двух вершин, должен быть согласован с их

принадлежностью к сообществам — по определению, сообщества представляют группы узлов, сильнее связанных между собой, чем с узлами других сообществ. Поэтому необходим подход, совмещающий в себе поддержку направленных взвешенных ребер одновременно со структурой сообществ.

Формально, каждому направленному ребру (упорядоченной паре (i,j)) соответствует вещественный вес $w_{ij} > 0$; для каждой вершины i задается метка принадлежности сообществам $\mathcal{C}_i$. Сообществом может быть любое подмножество вершин графа $N_c \subseteq N$, при этом каждая вершина может состоять в любом числе сообществ графа.

Общепринятого критерия похожести графов не существует, на практике используется ряд известных характеристик графов, по которым оценивается близость графов. В данной работе используются следующие характеристики:

- числовые: средняя степень вершины, взаимность ребер, ассортативность степеней, средний коэффициент кластеризации, эффективный диаметр гигантской компоненты, спектральный радиус;
- распределения: распределение степеней, кумулятивный коэффициент кластеризации, коэффициент кластеризации от степени вершины, распределение подграфов размера 3, достижимость вершин.

Под *похожестью* графов в данной работе понимается близость их числовых характеристик, а также некоторых распределений, для которых определена мера их сравнения, например, косинусная близость векторов-распределений подграфов в графе.

Как уже было отмечено, для практических приложений недостаточно генерировать лишь похожие графы, а необходимо привнести в них различия, например, для обеспечения достаточной анонимизации. Поэтому модель графа должна обеспечить баланс между похожестью и вариабельностью в смысле графовых свойств.

Вариабельностью множества графов будем называть дисперсию их числовых характеристикв этом множестве.

Таким образом, задача формулируется следующим образом. Необходимо разработать подход для генерации случайных графов, удовлетворяющий следующим **требованиям**:

- автоматическое обучение на заданном графе;
- возможность генерировать графы контролируемого размера;

– одновременная поддержка трех особенностей графа: направленные ребра, взвешенные ребра и структура сообществ;
– похожесть генерируемых графов на исходный: отклонение по каждой характеристике не выше соответствующих отклонений у альтернативных методов;
– вариабельность генерируемых графов: разброс значений числовых характеристик близок к таковому у реальных графов из одного домена.

2.2 Подход на основе вложения графа — ERGG

Самым популярным методом автоматического извлечения признаков из графов является обучение представления. В последние годы эта область привлекает большое внимание в связи с недавними успехами в области векторного представления слов (***word embedding***) [160] и адаптации этих идей к области графов [161; 162].

Вложение графа (***graph embedding***) — это представление его вершин точками векторного пространства. Интерес представляет задача сохранения при этом некоторых его свойств. Например, такое вложение в евклидово пространство, что любое расстояние между вершинами в графе близко к евклидову расстоянию между соответствующими векторами [163]. Тесно связанной проблемой является *обучение представления* (***representation learning***), где узлы графа должны быть отображены в векторы, такие что они являются полезными признаками в различных приложениях, таких как предсказание ребер и многоклассовая классификация (***multi-label classificcation***) [161].

Опишем теперь подход ERGG, предлагаемый для решения поставленной в работе задачи.

Сначала происходит обучение представления вершин исходного графа векторами небольшой размерности. Затем из некоторого вероятностного распределения, которое аппроксимирует распределение выученных векторов узлов, сэмплируются новые векторы, соответствующие будущим узлам. Наконец, полученные новые узлы связываются ребрами, завершая построение графа.

Более формально, на вход ERGG подается граф $G = (N,E)$ и коэффициент масштабирования $x > 0$ ($x \in \mathbb{R}$). На выходе получается новый случайный граф $G' = (N',E')$ с $|N'| \approx \lfloor xn \rfloor$ узлами. Алгоритм имеет следующие шаги:

1. Получить вложение графа $G = (N,E)$ в низкоразмерное пространство, так что его узлы $i \in N$ отображаются в вещественные векторы $\{\vec{r}_i\}_{i=1}^{n}$.
2. Аппроксимировать эмпирическое распределение векторов $\{\vec{r}_i\}_{i=1}^{n}$ и сэмплировать набор из $\lfloor xn \rfloor$ новых случайных векторов $\{\vec{q}_i\}_{i=1}^{\lfloor xn \rfloor}$ из того же вероятностного распределения. Эти векторы будут соответствовать узлам нового графа $(N',\cdot)$.
3. Соединить узлы графа $(N',\cdot)$ ребрами, используя модель вложения из шага 1, получая в результате граф $G' = (N',E')$.

Предполагается, что на шаге 1 можно использовать любой метод получения вложения, предусматривающий для пары узлов (i,j) функцию оценки $s_{ij} = s(\vec{r}_i, \vec{r}_j)$, характеризующую ребро (i,j). Эта функция используется далее в процессе создания ребер на основе векторного представления вершин.

Получив вложение заданного графа G и аппроксимировав распределение векторов его узлов $\{\vec{r}_i\}_{i=1}^{n}$ некоторой моделью распределения $\mathcal{R}$, получаем генеративную модель, задающую распределение вероятностей по графам, похожим на G. Для генерации такого случайного графа необходимо сэмплировать $\lfloor xn \rfloor$ векторов из $\mathcal{R}$ и, используя выученную функцию s_{ij}, соединить ребрами соответствующие пары новых узлов.

Обратим внимание, что приведенная схема подходит как для ориентированных, так и для неориентированных графов, а также допускает расширение на графы с взвешенными ребрами и/или сообществами — что будет продемонстрировано далее в разделе 2.4. Перед этим приведем обзор существующих подходов, релевантных ERGG: генераторов графов, похожих на данный и методов обучения представления.

2.3 Краткий обзор методов вложения направленных графов

Предлагаемый подход решает задачу генерации графа, похожего на данный, и основан на идее вложения графа. Поэтому далее кратко рассмотрены существующие релевантные методы вложения ориентированных графов.

В контексте работы интерес представляют такие методы обучения представления ориентированного графа, которые: 1) кодируют его важные свойства, и 2) позволяют реконструировать графы различного размера с такими свойствами, основываясь на представлении исходного графа. Для удобства будем называть вложением графа саму задачу нахождения представления. Многие методы вложения графов были либо разработаны для неориентированных графов — например, вложение на основе сил (***force-directed embedding***) [164], либо не предполагают интуитивного способа восстановления ориентированного графа — например, спектральные вложения [165]. Кроме того, недавние методы вложения, основанные на машинном обучении, успешно себя показали в приложениях, поэтому рассмотрим их далее.

DeepWalk Впервые такой подход для вложения графов был предложен Брайаном Перощи, Рами Аль-Рфу и Стивеном Скиеной (Perozzi, Bryan and Al-Rfou, Rami and Skiena, Steven) [161]. Основная идея заключается в адаптации языковой модели word2vec к графовым данным. Это делается путем запуска из каждой вершины графа серии случайных блужданий (***random walks***) по узлам вдоль ребер, а затем обучения модели word2vec на полученных последовательностях узлов графа, как на предложениях из слов. Цель word2vec — максимизировать вероятности совместного появления слов, которые встречаются в текстах рядом (в контекстном окне фиксированного размера). Формально, для обучающей последовательности слов $w_1, w_2, ..., w_T$ и размера контекста c, целевая функция[1] — это сумма логарифмов вероятностей, выраженная через параметры модели Θ:

$$J_\Theta = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leqslant j\leqslant c, j\neq 0}\log p(w_{t+j}|w_t) \to \max_\Theta, \tag{2.1}$$

где вероятность определяется как софтмакс функция (***softmax***) по всему словарю:

$$p(w_O|w_I) = \frac{\exp\left(\vec{u}_{w_I}\cdot\vec{v}_{w_O}\right)}{\sum_{w'_O}\exp\left(\vec{u}_{w_I}\cdot\vec{v}_{w'_O}\right)} \tag{2.2}$$

[1]Это цель модели skip-gram, которая используется в [161]. В альтернативной word2vec модели CBOW вместо $p(w_{t+j}|w_t)$ (прогнозирование контекста по целевому слову) оптимизируется $p(w_t|w_{t+j})$ (прогнозирование целевого слова по контексту).

Параметры модели $\Theta = \{\vec{u}_i, \vec{v}_i\}_{i=1}^n$ являются скрытыми векторными представлениями слов, векторы $\vec{u}_i$ выдаются после обучения как представления слов. В случае графа каждый $\vec{u}_i$ является вектором представления соответствующего узла.

Поскольку наиболее вычислительно дорогой частью является суммирование по всему множеству слов в знаменателе, в последние годы было предложено несколько способов аппроксимации функции софтмакс. Алгоритм DeepWalk использует иерархический софтмакс [166], который, говоря приближенно, заменяет софтмакс-слой (2.2) на двоичное дерево со словами в листьях. Поэтому нормализация для каждого слова вычисляется уже не за $O(n)$, а за $O(\log n)$, благодаря иерархической структуре.

BLM Билинейная реберная модель (***bilinear link model***, BLM), разработанная специально для ориентированных графов, была предложена Олегом Ивановым и Сергеем Бартуновым [167]:

$$p(j|i) = \frac{\exp\left(\vec{u}_i \cdot \vec{v}_j\right)}{\sum_k \exp\left(\vec{u}_i \cdot \vec{v}_k\right)} \tag{2.3}$$

Здесь параметры $\Theta = \{\vec{u}_i, \vec{v}_i\}_{i=1}^n$ ассоциированы с входным и выходным представлением каждого узла. Учитывая совместную вероятность $p(i,j) = p(i)p(j|i)$, целевой функцией является логарифм вероятности для всего графа:

$$J_\Theta = \sum_{(i,j)\in m} \log p(i,j) \to \max_\Theta \tag{2.4}$$

Для аппроксимации софтмакс авторы реализуют другую технику, контрастное оценивание (***Noise contrastive estimation***, NCE) [168]. Этот метод был разработан для оценки ненормализованных вероятностных моделей, в котором нормировочный коэффициент рассматривается как дополнительный параметр. Основная идея состоит в том, чтобы свести задачу обучения плотности вероятности к задаче бинарной классификации, а именно, отличить настоящее распределение данных $p_d(x)$ от некоторого шумового распределения $p_n(x)$. Применительно к языковым моделям [169], а также к задаче вложения графа [167] контрастное оценивание принимает следующую форму. В предположении, что шумовые элементы встречаются в выборке в k раз чаще, чем настоящие данных, наблюдаемое распределение смеси: $\frac{1}{k+1}p_d(x) + \frac{k}{k+1}p_n(x)$.

Тогда апостериорная вероятность того, что элемент x из настоящего распределения, равна $P(y=1|x) = \dfrac{p_d(x)}{p_d(x)+kp_n(x)}$, а $P(y=0|x) = \dfrac{kp_n(x)}{p_d(x)+kp_n(x)}$. Если модель $p_\theta(x)$ с набором параметров θ нацелена на подгон к настоящему распределению данных, то апостериорная вероятность становится функцией θ:

$$\begin{aligned} P(y=1|x,\theta) &= \frac{p_\theta(x)}{p_\theta(x)+kp_n(x)}, \\ P(y=0|x,\theta) &= \frac{kp_n(x)}{p_\theta(x)+kp_n(x)} \end{aligned} \tag{2.5}$$

Затем логистическая регрессия используется для оптимизации логарифма правдоподобия данных против шума:

$$J_\theta^{NCE} = \mathbb{E}_{x\sim p_d}\log P(y=1|x,\theta) + k\mathbb{E}_{x\sim p_n}\log P(y=0|x,\theta) \to \max_\theta \tag{2.6}$$

что корректно аппроксимирует $p_d(x)$ без требования нормировки для $p_\theta(x)$, как доказано в работе [168].

В случае BLM коэффициент нормализации становится новым параметром $Z_i = \log\sum_k \exp(\vec{u}_i\cdot\vec{v}_k))$, приводя к новому набору параметров $\Theta' = \{\vec{u}_i, \vec{v}_i, Z_i\}_{i=1}^n$ и вероятностной модели:

$$p_{\Theta'}(j|i) = \exp((\vec{u}_i\cdot\vec{v}_j) - Z_i) \tag{2.7}$$

Принимая во внимание, что p_d отражает действительные ребра графа, и выбирая p_n как $p_n(i,j) = p(i)p_n(j)$, новая целевая функция выглядит следующим образом:

$$J_\Theta^{NCE} = \frac{1}{m}\left(\sum_{(i,j)\in E}\log\frac{p_{\Theta'}(j|i)}{p_{\Theta'}(j|i)+kp_n(j)} + \sum_{(i,j)\sim p_n}^{kE}\log\frac{kp_n(j)}{p_{\Theta'}(j|i)+kp_n(j)}\right) \tag{2.8}$$

Другими словами, исходная цель (2.4) в BLM заменяется на NCE-цель (2.8), которая может быть эффективно оптимизирована.

LINE Аналогичный подход для вложения ориентированного взвешенного графа был предложен Цзянь Тан и др. (Jian Tang et al) [162]. Определяются так называемые близость первого и второго порядка между двумя узлами. Близость первого порядка узлов i и j выражается весом ребра w_{ij} и характеризует силу связи узлов:

$$p_1(i,j) = \frac{1}{1 + \exp\left(-\vec{u}_i \cdot \vec{u}_j\right)}, \tag{2.9}$$

в то время как близость второго порядка характеризует связь узла j с контекстом i:

$$p_2(j|i) = \frac{\exp\left(\vec{u}_i \cdot \vec{v}_j\right)}{\sum_k \exp\left(\vec{u}_i \cdot \vec{v}_k\right)}, \tag{2.10}$$

которое фактически совпадает с софтмакс в DeepWalk (2.2) и BLM (2.3).

Авторы предлагают оптимизировать две соответствующие задачи по отдельности:

$$\begin{aligned} J_\Theta^1 &= -\sum_{(i,j)\in E} w_{ij} \log p_1(i,j) \\ J_\Theta^2 &= -\sum_{(i,j)\in E} w_{ij} \log p_2(j|i) \end{aligned} \tag{2.11}$$

а затем просто конкатенировать векторы вложения $\vec{u}_i$ из обеих моделей.

Чтобы обойти долгое суммирование в знаменателе, используется другой подход, популярный в моделировании языка, — негативное сэмплирование (***Negative sampling***, NEG) [160]. NEG — это упрощение NCE, которое в действительности не аппроксимирует софтмакс, но сохраняет качество векторов представления. Это достигается путем замены $kp_n(x)$ в (2.5) на 1 и игнорирования нормировочного коэффициента:

$$\begin{aligned} P(y=1|x,\theta) &= \frac{p_\theta(x)}{p_\theta(x)+1} = \sigma(\vec{u}_i \cdot \vec{v}_j), \\ P(y=0|x,\theta) &= \frac{1}{p_\theta(x)+1} = \sigma(-\vec{u}_i \cdot \vec{v}_j), \end{aligned} \tag{2.12}$$

где функция $\sigma(x) = \dfrac{1}{1+e^{-x}}$ — сигмоида.

Подставляя это в (2.6), получаем целевую функцию NEG. Для одного ребра в LINE она имеет вид:

$$J_\Theta^{(i,j)} = \log \sigma(\vec{u}_i \cdot \vec{v}_j) + \sum_{j' \sim p_n(j')}^{k} \log \sigma(-\vec{u}_i \cdot \vec{v}_{j'}) \tag{2.13}$$

node2vec Другая модификация DeepWalk была предложена А. Гровером и Ю. Лисковцом (Grover Aditya, Leskovec Jure) [170]. Авторы рассматривают случайные блуждания 2-го порядка для более гибкого определения окрестности

узла. А именно, вводятся 2 параметра блуждания для управления следующими двумя случаями: 1) порядок обхода $i \to j \to k$ более вероятен, если узлы i и k соединены; 2) порядок обхода $i \to j \to i$ менее вероятен.

Здесь, в отличие от DeepWalk, целевая функция оптимизирована с использованием негативного сэмплирования.

2.3.1 Анализ и выводы

Поскольку основная идея ERGG предлагает и кодировать граф векторами, и восстанавливать из множества векторов граф, это накладывает некоторые ограничения на используемый метода вложения графа. Одним из требований является возможность восстановить тот же граф из его вложения, максимально сохранив его основные свойства.

DeepWalk и, следовательно, node2vec не позволяют восстановить направления ребер из векторов представления из-за симметричного контекста, используемого в word2vec. Другими словами, если инвертировать все ребра в графе, векторы представления будут точно такими же (при условии, использования одних и тех же начальных инициализаций).

Определение близости первого порядка в LINE (2.9) вообще не учитывает направления ребер и поэтому имеет смысл только в сочетании с моделью софтмакс второго порядка (2.10), которая обучается отдельно от нее и совпадает с софтмакс моделью BLM (2.3).

Поэтому в качестве методов вложения предварительно были выбраны методы BLM и LINE.

2.4 Метод генерации случайных графов на основе вложения графа — ERGG-dwc

В этом разделе предлагается конкретная реализация подхода ERGG — алгоритм ERGG-dwc, способный обрабатывать графы с весами на ребрах и сообществами. Веса интерпретируются как метки на ребрах, а сообщества — как

метки узлов. Положим, что входные данные представляют собой ориентированный взвешенный граф $G = (N,E)$ со структурой сообществ, заданной как метки узлов $\{\mathcal{C}_i\}_{i=1}^n$, и положительный коэффициент масштабирования $x \in \mathbb{R}$. Дополнительным (опциональным) параметром является величина шума ε. Порядок выполнения алгоритма следующий:

1. **Вложение**. Обучить представление графа $G = (N,E)$ с помощью модифицированного метода вложения (см. 2.4.1) в виде векторов $\{\vec{r}_i\}_{i=1}^n$; и найти порог t_G, отделяющий первые m пар узлов (i,j) с наибольшими значениями функции оценки $s(\vec{r}_i, \vec{r}_j)$ от остальных пар узлов $(i,j) \in N \times N$. Порог t_G будет использован на этапе генерации ребер.
2. **Аппроксимация + сэмплирование**. Выбрать случайно (с повторениями) $n' = \lfloor xn \rfloor$ векторов $\{\vec{q}_i\}_{i=1}^{n'}$ из множества $\{\vec{r}_i\}_{i=1}^n$ с добавлением малого гауссовского шума $\vec{g} \sim \mathcal{N}(0, diag(\varepsilon, ..., \varepsilon))$. Таким образом определяется отображение φ узлов исходного графа в узлы нового графа: $N \overset{\varphi}{\mapsto} N'$.
3. **Соединение**. Соединить ребрами те пары узлов k,l из N', для которых $s(\vec{q}_k, \vec{q}_l) > t_G$. Удалить висячие узлы при их наличии, получив граф $G' = (N'',E')$.

   Атрибуты.

   а) Каждому узлу $k \in N'$ назначить метку сообществ $\mathcal{C}'_k = \mathcal{C}_i$, где $i = \varphi^{-1}(k)$.

   б) Каждому ребру $(k,l) \in E'$ назначить вес $w'_{kl} = w_{ij}$, где $i = \varphi^{-1}(k)$, $j = \varphi^{-1}(l)$. Если $(i,j) \notin E$, присвоить вес по умолчанию $w_0 = \min_{(i,j)\in E} w_{ij}$.

Опишем теперь подробно все шаги алгоритма ERGG-dwc. Вместо решения задачи ERGG-dwc целиком, она была разделена на последовательность более простых подзадач, для которых было проведено исследование их возможных решений, оптимизируя каждую из них по отдельности. А именно, такими подзадачами являются: вложение + восстановление (2.4.1), аппроксимация + сэмплирование (2.4.2) и определение атрибутов (2.4.3). В конце раздела исследуется вычислительная сложность алгоритма (2.4.4).

2.4.1 Вложение + восстановление

Первая задача — представить узлы данного графа векторами, сохраняя максимальное количество информации. Для достижения этой цели, ребра исходного графа восстанавливаются обратно из векторов, максимизируя F_1 меру восстановленных ребер по отношению к исходным ребрам. Далее будем называть эту оценку F_1-мерой восстановления. Таким образом, шаг вложения тесно связан с шагом соединения.

Восстановление графа

Как восстановить граф $G = (N,E)$ из его вложения? В рамках подхода ERGG предполагается, что метод вложения предусматривает функцию оценки $s_{ij} = s(\vec{r}_i, \vec{r}_j)$ на парах узлов, которая в результате обучения оценивает ребра графа выше, чем пары узлов, не являющиеся ребрами. В BLM и LINE эту роль играет логарифм вероятности ребра $s_{ij} = \log\left(p(i)p_\theta(j|i)\right)$. После обучения s_{ij} можно использовать двумя способами для построения графа: как вероятность ребра (i,j) или с порогом (все пары с $s_{ij} > t_G$ становятся ребрами). Экспериментальным путем второй способ был найден более подходящим, в частности из-за того, что s_{ij} принимает значения, сильно отличные от 1, после сэмплирования новых векторов $\vec{r}_i$ и ее простая нормировка не приводит к удовлетворительному качеству. Самый простой способ выбора порога состоит в том, чтобы отобрать топ m пар узлов, ранжированных по оценке s_{ij}, в качестве ребер, устанавливая t_G равным оценке s_{ij} для пары с рангом $m + 1$.

Выбор метода вложения

Поскольку в методе ERGG-dwc максимизируется F_1-мера для восстановленных ребер графа, это накладывает некоторые ограничения на использование метода вложения.

Как уже было отмечено, модель первого порядка в LINE (2.9) полностью игнорирует направления ребер и поэтому имеет смысл только в сочетании с моделью софтмакс второго порядка, которая в любом случае обучается отдельно и совпадает с моделью софтмакс BLM (2.3). С другой стороны, поскольку LINE использует другие методы оптимизации и никогда не сравнивался с BLM, было принято решение скомбинировать идеи обоих методов.

Поскольку направленная модель софтмакс $p(j|i) = \frac{\exp(\vec{u}_i \cdot \vec{v}_j)}{\sum_k \exp(\vec{u}_i \cdot \vec{v}_k)}$, как в LINE, так и в BLM, включает в себя $\vec{u}$-векторы, а также $\vec{v}$-векторы для определения вероятности ребер, естественно использовать и те и другие при восстановлении графа. Проблема здесь в том, что из-за особенностей алгоритма для узлов с $d_i^{out} = 0$ их векторы $\vec{u}_i$ остаются неизменными с момента (случайной) инициализации в отличие от $\vec{v}_i$ векторов. По этой причине авторы BLM предлагают использовать $\{\vec{v}_i\}_{i=1}^n$ в качестве результирующих векторов представления, однако проблема со случайными $\vec{u}_i$ все еще может повлиять на восстановление графа.

Модифицированный метод вложения COMBO

Метод вложения COMBO был разработан на основе алгоритмов BLM и LINE и оптимизирован с точки зрения F_1 для тестового набора графов (подробнее см. 3.2.1). А именно, в качестве оптимизации целевой функции используется негативное сэмплирование:

$$J_\theta = \frac{1}{m} \sum_{(i,j)\in E} \left(\log \sigma(s_{ij}) + \sum_{j' \sim p_n(j')}^{\nu} \log \sigma(-s_{ij'}) \right), \tag{2.14}$$

где билинейная модель используется как функция оценки:

$$s_{ij} = \vec{u}_i \cdot \vec{v}_j - Z_i \tag{2.15}$$

Вектора вершин инициализируются как $\vec{u}_i, \vec{v}_i \sim \mathcal{U}[-\frac{1}{2\sqrt{d}}, \frac{1}{2\sqrt{d}}]$ (где d — размерность пространства вложения) и $Z_i = \log n$; шумовые ребра фильтруются так, что только $(i,j') \notin E$ отбираются в качестве негативных примеров (шума); шумовое распределение выбрано $p_n(j) \propto d_j^{3/4}$ [162].

Таким образом, множество параметров алгоритма: $\theta = \{\vec{u}_i, \vec{v}_i, Z_i\}_{i=1}^{n}$; вектор представления для узла $i \in N$ есть: $\vec{r}_i = \begin{bmatrix}\vec{u}_i & \vec{v}_i & Z_i\end{bmatrix}^T$.

Оптимизация выполняется с помощью асинхронного стохастического градиентного спуска (как в LINE и BLM). На каждом шаге градиент вычисляется по одному ребру (i,j):

$$\frac{\partial J_\theta^{(i,j)}}{\partial \theta} = \frac{\partial}{\partial \theta}\log \sigma(s_{ij}) + \sum_{j' \sim p_n(j')}^{\nu} \frac{\partial}{\partial \theta}\log \sigma(-s_{ij'}) = \sigma(-s_{ij})\frac{\partial s_{ij}}{\partial \theta} + \sum_{j' \sim p_n(j')}^{\nu} \sigma(s_{ij'})\frac{\partial s_{ij'}}{\partial \theta}, \tag{2.16}$$

где $\sigma(x) = \dfrac{1}{1+e^{-x}}$ — сигмоида, а производная от s_{ij}:

$$\frac{\partial s_{ij}}{\partial \theta} = \begin{bmatrix} -\vec{v}_j \\ -\vec{u}_i \\ 1 \end{bmatrix} \tag{2.17}$$

Регуляризация была устранена, поскольку уменьшала качество восстановления ребер.

После обучения параметров Θ модели определяется порог t_G. Для этого все пары узлов сортируются по убыванию оценки s_{ij}, и t_G выбирается равным оценке s_{ij} для пары с рангом $m+1$.

Вложение графа считалось успешным, если его ребра можно восстановить с $F_1 \geqslant 0.99$. Это означает, что полученное представление объясняет более 99% ребер графа в рамках модели, в то время как оставшийся 1% может быть выбросами.

Отметим, что размерность пространства вложения d является существенным параметром. Минимальное d, такое, что F_1 достигает 0,99 для конкретного графа, можно рассматривать как “сложность” этого графа в рамках модели вложения. Поскольку заранее такое значение d для графа неизвестно, оно определяется путем бинарного поиска.

Обратим внимание, что предложенный модифицированный метод вложения специально адаптирован для отличения ребер от не ребер и, таким образом, не претендует на полезность в обычных приложениях представления графа. Во всяком случае, таких экспериментов не проводилось.

2.4.2 Аппроксимация распределения + сэмплирование

На этом этапе имеется векторное представление узлов входного графа $\{\vec{r}_i\}_{i=1}^n$, такое что граф можно восстановить из него с качеством $F_1 \geqslant 0.99$. Следующая задача состоит в том, чтобы смоделировать распределение векторов узлов $\vec{r}_i \sim \mathcal{R}$ таким образом, чтобы новые векторы узлов, выбранные из $\mathcal{R}$, порождали (используя аналогичную процедуру восстановления из раздела 2.4.1) графы с похожими характеристиками. На этом шаге происходит создание модели генерации графа и обеспечивается рандомизация и контроль размера генерируемых графов. Для экспериментального сравнения графов разного размера используется косинусное сходство их 3-GP векторов и сравнение "на глаз" формы их распределений степеней.

После обучения представления исходного графа, вектора его узлов кодируют информацию о структуре графа. Основное предположение заключается в том, что ключевые статистические свойства графа отражены в *распределении* векторов $\{\vec{r}_i\}_{i=1}^n$, а не в отдельных векторах. Поэтому необходима модель распределения $\mathcal{R}$ такая, что $\vec{r}_i \sim \mathcal{R}$ также будут отражать эти свойства. Далее, насэмплировав набор новых векторов узлов из $\mathcal{R}$ и построив новый граф согласно описанной выше процедуре, ожидается что он будет иметь схожие характеристики. Эта идея хорошо обосновывается на примере случайных графов скалярного произведения, где распределение векторов узлов аналитически определяет свойства графа [155]. Еще одним преимуществом такого подхода является то, что число выбранных векторов и, следовательно, размер генерируемого графа может быть задан любым.

В качестве модели распределения $\mathcal{R}$ были рассмотрены несколько подходов.

Модель гауссовых смесей Первый метод подгоняет параметры модели гауссовых смесей (***Gaussian Mixture Model***, GMM) к набору векторов $\{\vec{r}_i\}_{i=1}^n$ с помощью алгоритма максимизации ожидания. Затем новые векторы сэмплируются из этой GMM. Количество компонентов n_{comp} для использования в смеси является гиперпараметром. Поскольку GMM обучается на всем множестве $\{\vec{r}_i\}_{i=1}^n$, ее параметры кодируют информацию об исходном графе.

Гауссовский шум Второй метод состоит в простом запоминании всего набора векторов $\{\vec{r}_i\}_{i=1}^n$ и добавления гауссовского шума (***Gaussian noise***, GN) величиной ε. Чтобы сэмплировать из $\mathcal{R}_\varepsilon$, случайным образом выбирается $i \in \{1..n\}$ и возвращается $\vec{r}_i + \vec{g}$, где $\vec{g} \sim \mathcal{N}(0, diag(\varepsilon, ..., \varepsilon))$. По сути это напоминает метод ядерной оценки плотности с нормальным ядром. В случае больших графов хранение в модели всех n векторов может быть избыточным, поэтому имеет смысл ограничиться некоторым их подмножеством, однако это направление не было исследовано.

Существуют также и другие варианты решения задачи аппроксимации эмпирического распределения. Для генерации векторов из того же распределения были опробованы Генеративно-состязательные сети (***Generative adversarial networks***) [171], автоэнкодеры (в частности, Variational autoencoder и Sequence-to-sequence autoencoder[2]). Современные подходы на основе нейросетей выглядят перспективно для моделирования неизвестных распределений, однако в экспериментах нам не удалось добиться похожести сгенерированных графов, близкой к той, что давали первые два метода.

По результатам экспериментов (см. 3.2) для ERGG-dwc был выбран метод GN.

Выбор амплитуды шума

Что происходит, если в GN не использовать шум? Следующая проблема может возникнуть, когда коэффициент масштабирования x достаточно велик, напимер, $x = 10$. Рассмотрим узлы i и j в исходном графе G. После сэмплирования они будут отображены в 2 группы новых узлов (размером около 10) с одинаковыми векторами, равными $\vec{r}_i$ и $\vec{r}_j$ соответственно. В зависимости от наличия ребра (i,j) в E, либо каждый узел из 1-й группы будет соединен с каждым узлом 2-й, либо не будет соединен ни один. Это приводит к наличию ступенчатости в распределении степеней сгенерированного графа[3], что во-пер-

[2] https://blog.keras.io/building-autoencoders-in-keras.html

[3] Отметим, что эта проблема похожа на “эффект лестницы” в детерминированных Кронекеровских графах, как было отмечено в [44].

вых, нереалистично и во-вторых, сужает множество возможных генерируемых графов.

Напротив, слишком высокая амплитуда шума ε исказит свойства распределения, поэтому следует найти некоторые компромиссные значения. В экспериментах ε увеличивался, пока косинусное сходство 3-GP восстановленных графов не начинало снижаться. Полученное значение ε может быть рекомендовано для использования по умолчанию.

Размер сгенерированных графов

Если масштабировать граф с n_0 узлами и m_0 ребрами с коэффициентом x, у него должно быть $n_x \approx xn_0$ узлов. Какое количество ребер m_x он должен иметь, или, другими словами, какая ожидаемая зависимость $m(n)$? В существующих моделях графов нет согласия: авторы ReCoN [56] сообщают о линейном законе $m(n)$ со ссылками на другие работы, в то же время эмпирически обнаружен степенной закон уплотнения [31] $m \sim n^a$, где показатель $a \in [1; 2]$ зависит от конкретного графа. Генераторы случайных графов также дают разные ответы: в модели Барабаши-Альберт [76] $m \sim n$, в графах скалярного произведения [156] $m \sim n^2$, в Кронекеровских графах [44] $m \sim n^a$, где $a = \log_{n_0} m_0$ определяется в результате обучения на исходном графе.

В подходе ERGG $m \sim n^2$ справедливо для графов, генерируемых одной моделью, независимо от метода аппроксимации, что отражено в следующей теореме.

Теорема 1. *Пусть $\mathcal{R}$ вероятностное распределение в пространстве $\mathbb{R}^d$, функция $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Если из распределения $\mathcal{R}$ получен набор случайных векторов $\{\vec{r}_i\}_{i=1}^n$, соответствующих n вершинам графа, и наличие ребра (i,j) в графе G задается условием $s(\vec{r}_i, \vec{r}_j) > t_G$, то число ребер m графа будет расти как $m \propto n^2$.*

Доказательство. Поскольку $\vec{r}_i$ и $\vec{r}_j$ из одного распределения, то вероятность генерации ребра между двумя произвольными вершинами $p_{ij} = P(s_{ij} > t_G) = p$ константа и зависит только от $\mathcal{R}$. Доля пар вершин, соединенных ребром, не

зависит от числа сэмплированных векторов вершин, т. е. $\mathbb{E}(m/n^2) = p = const$. Отсюда $m \propto n^2$ (без удаления висячих узлов). □

Отсюда следует, что при увеличении коэффициента масштабирования x генерируемые графы будут быстрее становится плотнее, чем это наблюдается при эволюции реальных графов.

Поскольку реальные графы могут демонстрировать разные законы роста, можно трактовать ERGG-dwc имитацию реального графа не как его будущее состояние, а как его масштабированную версию.

2.4.3 Атрибуты: метки сообществ и веса ребер

Теперь возможно генерировать ориентированные графы контролируемого размера, похожие по структуре на заданный. Предположим, что исходный граф взвешенный и имеет структуру сообществ. Последняя задача заключается в корректной обработке и назначении меток сообществ и весов ребер в сгенерированном графе. При этом их согласованность с топологией графа должна быть сохранена.

Структура сообществ

Структура сообществ графа рассматривается как совокупность меток его узлов: каждый узел i имеет (возможно, пустой) набор меток сообществ $\mathcal{C}_i$, которым он принадлежит. В алгоритме ERGG-dwc предлагаетя наследовать эти метки в сгенерированном графе в процессе сэмплирования в методе GN: если вектор узла k нового графа был сэмплирован из вектора узла i, он имеет те же метки $\mathcal{C}'_k = \mathcal{C}_i$. Таким образом, при равномерном сэмплировании узлов, сообщества в новом графе становятся пропорционально масштабированными образами исходных сообществ.

Нужно отметить, что, подобно случаю распределения степеней при нулевом шуме, проблема “ступенчатости” может возникнуть и здесь, когда

коэффициент масштабирования достаточно велик. Такое поведение нереалистично, поскольку известно, что распределение размеров сообществ в реальных графах следует степенному закону [25], а в данном случае степенной хвост из сообществ малых размеров будет отсутствовать. Некий аналог шума может быть введен для меток сообществ, что может быть вариантом будущих исследований.

Веса ребер

Веса в графе — это числовые атрибуты его ребер, и они часто имеют топологический смысл, который зависит от приложения. Например, это может быть количество совместных появлений двух слов или длина маршрута между двумя городами. Что означает большой вес ребра между двумя узлами для топологии графа? В первом примере два слова сильно связаны, а во втором города располагаются далеко друг от друга. Поэтому трактовка веса ребра лишь как атрибута имеет наиболее общее применение.

Чтобы назначить веса ребер в сгенерированном графе, в алгоритме ERGG-dwc наследуются веса ребер исходного графа при соответствующих узлах, определенных в процессе сэмплирования GN. Для ребра (k,l) нового графа, если вектора узлов k, l были сэмплированы из векторов узлов i, j исходного графа, ребру присваивается вес соответствующего ребра $w'_{kl} = w_{ij}$.

Остается вопрос: что, если исходный граф $G(N,E)$ не имеет ребра (i,j)? Одной из причин того является неправильное вложение ребра (i,j). В целом, поскольку доля таких ребер $(i,j) \notin E$ после успешного вложения составляет менее 1%, их веса не сильно повлияют на общую картину. В этом случае можно предложить несколько стратегий, например, равномерно выбрать случайный вес: $w_0 \sim \mathcal{U}(\{w_{ij}\}_{(i,j)\in E})$.

Вторая причина — шум, добавленный к выученным векторам узлов. Предположим, что вложение является идеальным ($F_1 = 1$) и используется небольшой шум $\varepsilon > 0$. Условие $(i,j) \notin E$ означает, что векторы $\vec{r}_i, \vec{r}_j$ удовлетворяют $s(\vec{r}_i, \vec{r}_j) \leqslant t_G$, тогда как сэмплированные вектора с шумом $\vec{q}_k, \vec{q}_l$, соответствующие $\vec{r}_i, \vec{r}_j$, дают $s(\vec{q}_k, \vec{q}_l) > t_G$. Поскольку величина шума мала, $s(\vec{q}_k, \vec{q}_l)$, скорее всего, несильно отличается от t_G. Поэтому можно считать ребро (k,l) слабой связью. Если веса ребер отражают силу связей в смоделирован-

ном графе, целесообразно присвоить такому ребру минимально возможный вес (k, l): $w_0 = \min_{(i,j)\in E} w_{ij}$.

2.4.4 Вычислительная сложность алгоритма ERGG-dwc

Описанный алгоритм ERGG-dwc состоит из этапа обучения представления, который выполняется один раз для исходного графа, и этапа генерации графа, который может быть выполнен произвольное количество раз — по количеству требуемых графов. Рассмотрим вычислительную сложность каждой части в отдельности.

Лемма 1. *Вычислительная сложность этапа обучения алгоритма ERGG-dwc составляет* $O((\frac{m^2}{n} + n^2)d)$.

Доказательство. Основное отличие с точки зрения сложности алгоритма вложения COMBO от BLM заключается в фильтрации шумовых ребер. Негативными сэмплами в NEG являются шумовые ребра: $(i,j') \notin E$. В случае COMBO случайно выбираются узлы $j' \in N$ пока не найдется такой, что $(i,j') \notin E$; прямая проверка на наличие (i,j') в E выполняется за d_i^{out} операций. Поскольку реальные графы разрежены, то $d_i^{out} \ll n$ (по крайней мере, асимптотически), поэтому вероятностью повторного выбора можно пренебречь. В результате фильтрация ребер дает в d_i^{out} раз больше шагов, что в среднем есть m/n. На каждой эпохе обучения количество шагов равно $m(1+\nu\frac{m}{n})(2d+1)$, где ν — количество негативных сэмплов для каждого ребра, d — размерность пространства вложения. После исключения независимых от графа параметров сложность алгоритма вложения равна $O(\frac{m^2}{n}d)$.

После вложения графа G порог t_G определяется путем вычисления оценки s_{ij} для каждой пары узлов, их сортировки и выбора $(m+1)$-й. На это требуется $O(n^2d) + O(n \log n) = O(n^2d)$ операций. □

Лемма 2. *Вычислительная сложность этапа генерации графа алгоритма ERGG-dwc составляет* $O(x^2n^2d)$.

Доказательство. При заданном коэффициенте масштабирования x генерация $n' = \lfloor xn \rfloor$ новых векторов узлов методом GN требует $O(xnd)$ шагов, что

незначительно по сравнению со сложностью генерации ребер. Для генерации ребер используется наивный подход: вычисляется s_{ij} для каждой пары $(i,j) \in N' \times N'$ и сравнивается с порогом t_G, — что дает квадратичную сложность $O(x^2n^2d)$. Присвоение меток сообществ и весов ребер не меняет вышеуказанную сложность, поскольку метки и веса копируют соответствующие значения. Определение минимального веса ребра требует $O(m)$ операций, что не увеличивает оценку сложности, поскольку $m < n^2$. □

К сожалению, на практике квадратичная сложность сильно затрудняет генерацию графов с большим n' (более миллиона вершин), поэтому в будущем требуется разработка более быстрого метода генерации ребер.

Общая сложность алгоритма складывается из этапа обучения и этапа генерации и выражается следующей теоремой.

Теорема 2. *Общая вычислительная сложность алгоритма ERGG-dwc составляет* $O((\frac{m^2}{n} + x^2n^2)d)$.

Доказательство. Следует из лемм 1 и 2. □

2.5 Выводы ко второй главе

В предложенном подходе ERGG впервые используется вложение графа для генерации случайных графов. Результатом вложения графа является отображение его вершин в низкоразмерные вектора, которые в совокупности кодируют некоторые статистические характеристики исходного графа. Вложение считается успешным, если из выученных векторов могут быть восстановлены почти все (не менее 99%) ребра исходного графа при минимальной размерности этих векторов.

Главным преимуществом использования вложения в контексте генерации графов является удобство представления графа в виде распределения векторов узлов. Это позволяет использовать единую схему сэмплирования случайных графов, независимо от домена и свойств исходного графа. Кроме того, для синтетического графа можно задать произвольное число векторов узлов, что позволяет достаточно точно контролировать размер результирующего графа.

В рамках предложенного подхода разработан алгоритм ERGG-dwc, который поддерживает графы со структурой сообществ и взвешенными ребрами. Однако его недостатками являются негибкий закон масштабирования генерируемых графов $m \propto n^2$, и квадратичная от числа вершин сложность генерации ребер.

В следующей главе будет подробно описана программная реализация метода ERGG-dwc и приведены результаты его экспериментальных исследований.

Глава 3. Программная реализация алгоритма ERGG-dwc и экспериментальные исследования

Данная глава посвящена программной реализации описанного алгоритма ERGG-dwc и ее экспериментальным исследованиям.

Вначале представлена программная система, реализующая фреймворк для анализа и тестирования ERGG-dwc и альтернативных моделей случайных графов. Далее идет описание экспериментальных исследований, проведенных в процессе разработки ERGG-dwc и касающихся выбора того или иного подхода и определения оптимальных параметров методов. Последняя часть посвящена сравнению реализации ERGG-dwc с наиболее релевантными существующими алгоритмами.

3.1 Описание программной системы

Для целей разработки и исследования ERGG-dwc и других моделей случайных графов была реализована программная система (преимущественно на языке `python`), которая состоит из следующих частей:

1. Менеджер графовых данных.
2. Обучение представления графа.
3. Работа с представлением графа.
4. Генерация графа.
5. Подсчет характеристик.
6. Фреймворк для тестирования.

Менеджер графовых данных обеспечивает хранение реальных графов в определенном формате и доступ к ним. Графы хранятся в виде списка ребер, представленных парами вершин или тройками $\langle i, j, w_{ij} \rangle$. Структура сообществ графа хранится как список множеств вершин, входящих в одно сообщество. Такой формат совместим с большинством алгоритмов генерации графов и поиска сообществ, а также поддерживается многими графовыми коллекциями. Для работы с графами использовалась библиотека `networkx`. Основными источниками реальных данных была коллекция сетей KONECT (Koblenz Network

Collection) [172] и датасет, собранный исследовательской группой SNAP из Стэнфордского университета [173].

Модуль *обучения представления графа* реализует обобщенный метод вложения COMBO на основе BLM и LINE. Оригинальные версии BLM и LINE[1] написаны на `C++`, рабочий вариант COMBO — на `cython`. Выбор инструмента `cython` мотивирован тем, что он совмещает удобство интерфейсов `python` с преимуществами производительности типов данных языка `C++`. Реализация COMBO поддерживает параллельное исполнение потоков и по производительности не уступает оригинальному BLM. Модуль также позволяет запускать другие методы вложения графа, в частности node2vec.

Работа с представлением графа состоит в хранении представлений для графа и различных методах его аппроксимации и сэмплирования. Реализованы описанные в 2.4.2 методы GN, GMM (использовалась библиотека `scikit-learn`), также есть возможность запускать генеративно-состязательные сети[2] и автоэнкодеры[3].

Модуль генерации отвечает за этап соединения вершин графа ребрами по заданной схеме; в случае ERGG-dwc это отсечение по порогу на основе функции оценки s_{ij}. Кроме того, этот модуль позволяет запускать сторонние генераторы, в частности, Gscaler[4] и SKG (используя библиотеку `snap`[5]).

В *модуле подсчета характеристик* доступны все упомянутые графовые характеристики для анализа графов, есть возможность строить графики. Для подсчета большинства характеристик используются инструменты библиотеки `snap`, для остальных библиотека для научных вычислений `scipy`.

Наконец, *фреймворк* для проведения анализа на всех этапах разработки и сравнительного тестирования моделей, с помощью которого построены все графики в работе. Описание проведенных экспериментов представлено в следующих разделах.

Кроме того, была разработана веб-демонстрация алгоритма ERGG-dwc, доступная по адресу `http://ergg.at.ispras.ru`.

[1] Общедоступная реализация на гитхаб `https://github.com/tangjianpku/LINE`.
[2] Исходный код `http://www.github.com/goodfeli/adversarial`
[3] `https://blog.keras.io/building-autoencoders-in-keras.html`
[4] Реализация на Java `https://github.com/jayCool/GscalerSource`
[5] `https://github.com/snap-stanford/snap`

3.2 Экспериментальное исследование ERGG-dwc

В процессе разработки алгоритма ERGG-dwc проводились экспериментальные исследования возможных решений каждой из поставленных подзадач. Для этого использовался набор ориентированных графов малого и среднего размера из разных доменов, а также несколько синтетических графов. Описание и параметры графов приведены в таблице 4. Для исследования поведения атрибутов были взяты реальные взвешенные графы, сообщества в которых найдены алгоритмом поиска сообществ OSLOM [174], а также синтетический граф, полученный с помощью генератора LFR (таблица 6).

Таблица 4 — Датасет направленных графов. d — размерность вложения.

Описание графа	**имя**	**n**	**m**	**d**
Каратэ-клуб Zachary[6]	Karate	34	78	3
Транскрипция генов дрожжей[7]	Yeast	688	1079	9
Мобильные звонки[8]	VAST	400	1562	12
Пищевые цепочки Флорида Бэй[9]	Foods	128	2106	8
Эго-сеть из Твиттер [173]	TW	146	1309	12
Синтетический Кронекеровский граф[10] [175]	Kron	2187	11675	24
Смежность слов в японских текстах[7]	Words	2704	8300	17
Синтетический ER-граф[11] [175]	ER	800	8000	26
Эго-сеть из Google-plus [173]	G+	1243	106485	62

Опишем теперь результаты исследований ERGG-dwc в соответствии с его подзадачами.

[6] https://en.wikipedia.org/wiki/Zachary%27s_karate_club ребра трактованы как напраленные.

[7] http://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks

[8] http://hcil2.cs.umd.edu/newvarepository/VAST%20Challenge%202008/challenges/MC3%20-%20Cell%20Phone%20Calls/

[9] http://vlado.fmf.uni-lj.si/pub/networks/data/bio/foodweb/foodweb.htm

[10] Использован генератор библиотеки `snap` с параметрами `"krongen -m:'0 0.783, 0.003, 0.733; 0.147, 0.636, 0.772; 0.028, 0.700, 0.009' -i:9"`

[11] Использован генератор библиотеки `snap` с параметрами `"graphgen -g:e -n:800 -m:8000"`

Таблица 5 — Сравнение компонентов алгоритмов BLM, LINE и предложенной модификации (COMBO).

Компонент	**LINE**	**BLM**	**COMBO**
аппроксимация цели J	NEG	NCE	NEG
функция оценки s_{ij}	$(\vec{u}_i,\vec{v}_j)$	$(\vec{u}_i,\vec{v}_j) - Z_i$	$(\vec{u}_i,\vec{v}_j) - Z_i$
инициализация $\lvert\vec{u}_i\rvert$; $\lvert\vec{v}_i\rvert$	$\mathcal{N}(0, diag(\frac{1}{d}, ..., \frac{1}{d}))$; 0	$\mathcal{U}[-\frac{1}{2d}, \frac{1}{2d}]$; $\mathcal{U}[-\frac{1}{2d}, \frac{1}{2d}]$	$\mathcal{U}[-\frac{1}{2\sqrt{d}}, \frac{1}{2\sqrt{d}}]$; $\mathcal{U}[-\frac{1}{2\sqrt{d}}, \frac{1}{2\sqrt{d}}]$
шумовые распределения $p_n(i)$; $p_n(j)$	i с вер-тью 1; $p_n(j) \propto d_j^{3/4}$	i с вер-тью 1; $p_n(j) \propto d_j$	i с вер-тью 1; $p_n(j) \propto d_j^{3/4}$
нормировочный параметр Z_i (для восстановления ребер)	прямой подсчет	из модели / прямой подсчет	из модели / прямой подсчет
фильтр шумовых ребер	НЕТ	НЕТ	ДА
регуляризация	снижение скорости обучения до 0	L^2	НЕТ

3.2.1 Параметры метода вложения

Алгоритмы BLM и LINE были детально проработаны и рассмотрены следующие их компоненты: способ оптимизации целевой функции J, функция оценки s_{ij}, инициализация векторов узлов, распределение шумовых данных, фильтрация шумовых ребер и метод регуляризации (таблица 5).

Для разработки модифицированного метода вложения COMBO, был реализован алгоритм, который обобщает и параметризует все перечисленные алгоритмические компоненты BLM и LINE. Для справедливого сравнения были зафиксированы общие параметры и изменялись остальные. Число эпох обучения (количество итераций по всем ребрам) было установлено равным 200, размерность векторов $d = 30$, количество шумовых сэмплов $\nu = 25$, скорость обучения $\eta = 0{,}01$ (эти значения параметров взяты из статьи BLM [167]) и выключена регуляризация.

С помощью жадного алгоритма была найдена комбинация компонентов, работающая значительно лучше, чем оригинальные методы вложения. Резуль-

таты качества восстановления ребер приведены на рисунке 3.1. Параметры найденной комбинации представлены в таблице 5 в графе COMBO и используется в дальнейшем.

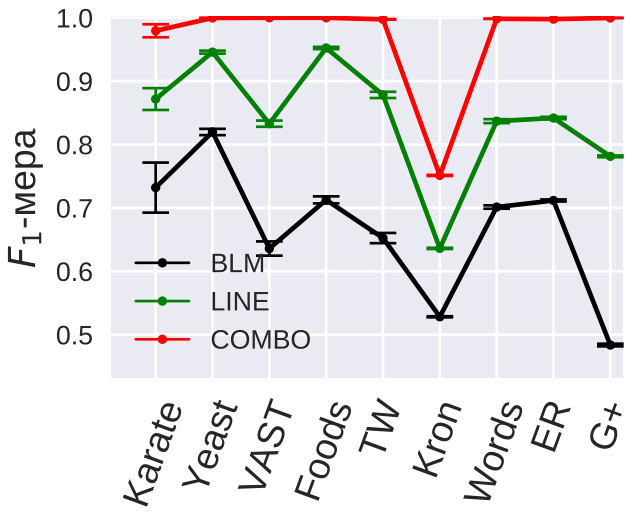

Рисунок 3.1 — F_1-мера восстановления ребер графа при использовании оригинальных методов вложения BLM и LINE, и разработанной модификации COMBO. Разброс значений оценен по 5 запускам.

Было исследовано, как F_1 зависит от размерности пространства вложения d. Все графы из датасета успешно достигали $F_1 = 0.99$ при некотором d, отражающую их "сложность". Минимальное значение d, соответствующее $F_1 > 0.99$ для конкретного графа, определялось с помощью бинарного поиска. Это оправданно, поскольку функцию $F_1(d)$ можно считать монотонно возрастающей, что подтверждается эмпирически (см. рисунок 3.2). При этом F_1-мера для представления графа при разных d оценивалась приблизительно — не по всем парам узлов, а лишь по фиксированному числу случайно выбранных пар, чтобы избежать квадратичной сложности восстановления графа.

3.2.2 Метод аппроксимации распределения

Для исследования метода аппроксимации распределения векторов вершин, фиксировалось вложение для каждого тестового графа так, что $F_1 > 0.99$ при минимальном d. Тестировались два метода аппроксимации распределения: 1) модель гауссовых смесей GMM с параметром количества компонентов n_{comp}; 2) добавление Гауссовского шума (GN), с параметризованной амплитудой ε.

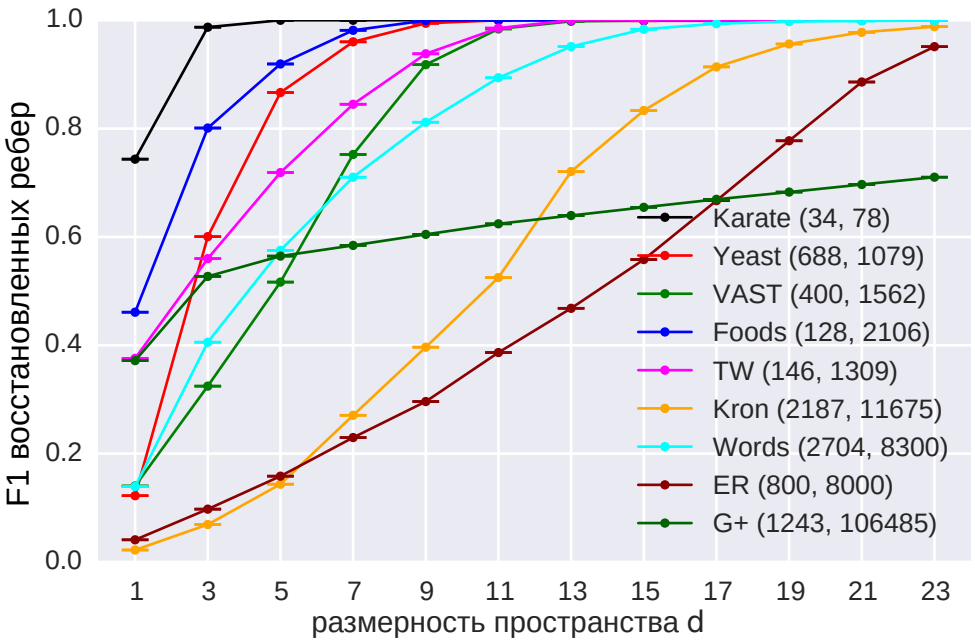

Рисунок 3.2 — F_1-мера восстановления ребер различных графов в зависимости от размерности пространства вложения d. В скобках после названия графа указаны его число вершин n и ребер m. Разброс значений оценен по 5 запускам.

Входом является набор векторов $\vec{r}_i = \begin{bmatrix}\vec{u}_i & \vec{v}_i & Z_i\end{bmatrix}^T$ (длины $2d+1$) для $i = 1..n$ и порог t_G. Метод аппроксимации применяется для заданного набора $\{\vec{r}_i\}_{i=1}^n$, а затем полученная модель используется для сэмплирования новых $n' = \lfloor xn \rfloor$ векторов. Соответствующие узлы нового графа соединяются ребрами по условию $s_{ij} > t_G$, висячие узлы удаляются, образуя граф $G' = (N'', E')$.

Для сравнения качества воспроизведения графов измерялось косинусная близость между 3-GP сгенерированного графа и исходного, количество узлов и количество ребер относительно их ожидаемых значений — n'/xn и m'/x^2m — при различных коэффициентах масштабирования x и параметрах метода аппроксимации.

Для GMM увеличение числа компонент n_{comp} с 20 до 320 увеличивает близость по 3-GP, хотя оно становится сопоставимым с GN только при весьма высоких значениях n_{comp}. Для GN при увеличении ε более 0.2 начинает уменьшаться близость 3-GP, в то время как значения около $0.1 - 0.2$ являются приемлемыми. Соответствующие графики приведены в приложении А.1.

Если коэффициент масштабирования $x = 1$ и висячие узлы не удаляются, то ожидается, что $n' = n$ и $m' \approx m$. Из-за случайности выбора векторов некоторые узлы нового графа оказываются ни с чем не соединенными (даже

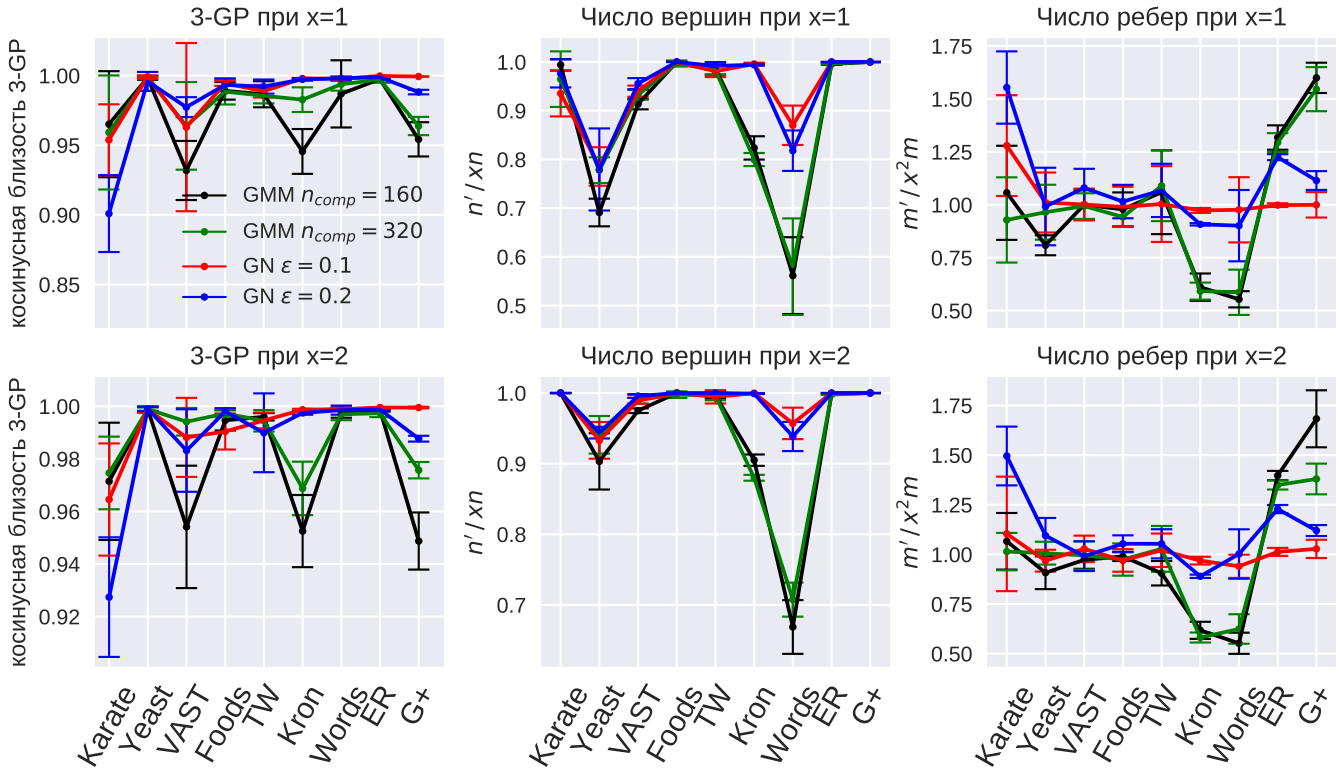

Рисунок 3.3 — Сравнение методов аппроксимации распределения GMM и GN с различными параметрами с точки зрения косинусной близости 3-GP и отношений количества вершин и ребер к их ожидаемым количествам. Разброс значений оценен по 5 запускам.

при $\varepsilon = 0$) и, следовательно, удаляются, поэтому всегда $n' \leqslant n$. На рисунке 3.3 (верхний ряд) показаны относительные количества узлов n'/n и ребер m'/m сгенерированного графа. Можно видеть, что для некоторых графов до 40% узлов становятся висящими при использовании GMM и до 20% для GN. Ту же тенденцию показывает отклонение от 1 по количеству ребер.

В случае $x \neq 1$ имеем $n' \leqslant xn$, $m' \approx x^2 m$. На рисунке 3.3 (нижний ряд, $x = 2$) показаны n'/xn и m'/x^2m по сравнению с 1 для методов GMM и GN. Поведение то же, что и для случая $x = 1$. Интересно, что более высокая величина шума ε дает большее отклонение m' от ожидаемого x^2m, тогда как n'/xn наоборот приближается к 1. Таким образом, было определено, что GN с $\varepsilon \in [0.1; 0.2]$ работает лучше всего в смысле рассмотренных показателей (рисунок 3.3).

Рисунок 3.4 демонстрирует ступенчатый эффект распределения степеней для графа эго-сети Twitter при $x = 16$. Обратим внимание, что для $\varepsilon = 0.0$ и $\varepsilon = 0.2$ размеры графов и их 3-GP почти идентичны, а распределение степеней сильно различаются.

[12] http://moreno.ss.uci.edu/data.html#oz

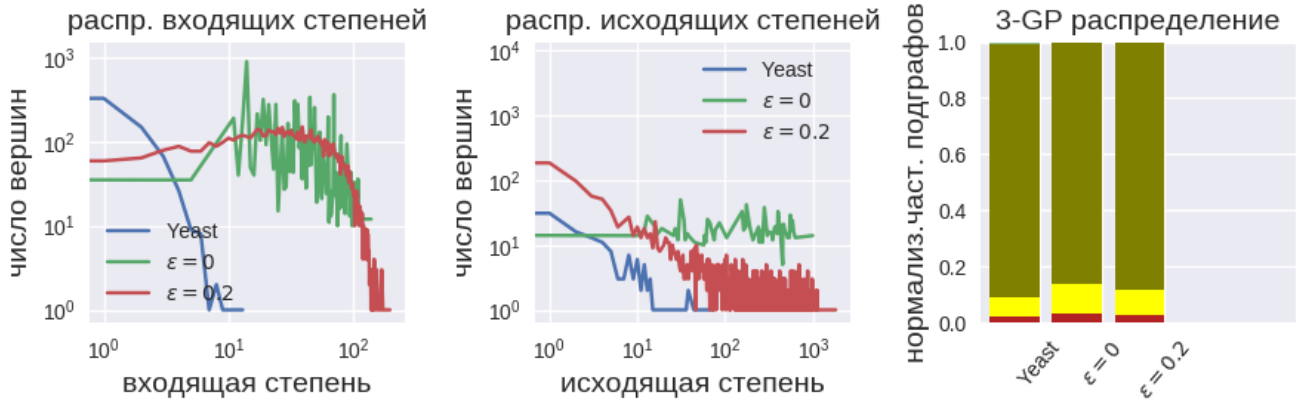

Рисунок 3.4 — Ступенчатый эффект в распределении степеней при $\varepsilon = 0.0$ и большом коэффициенте масштабирования ($x = 16$). При увеличении шума до $\varepsilon = 0.2$ график распределения степеней становится более гладким, что больше соответствует реальности. Исходный граф Yeast размера $n = 688$.

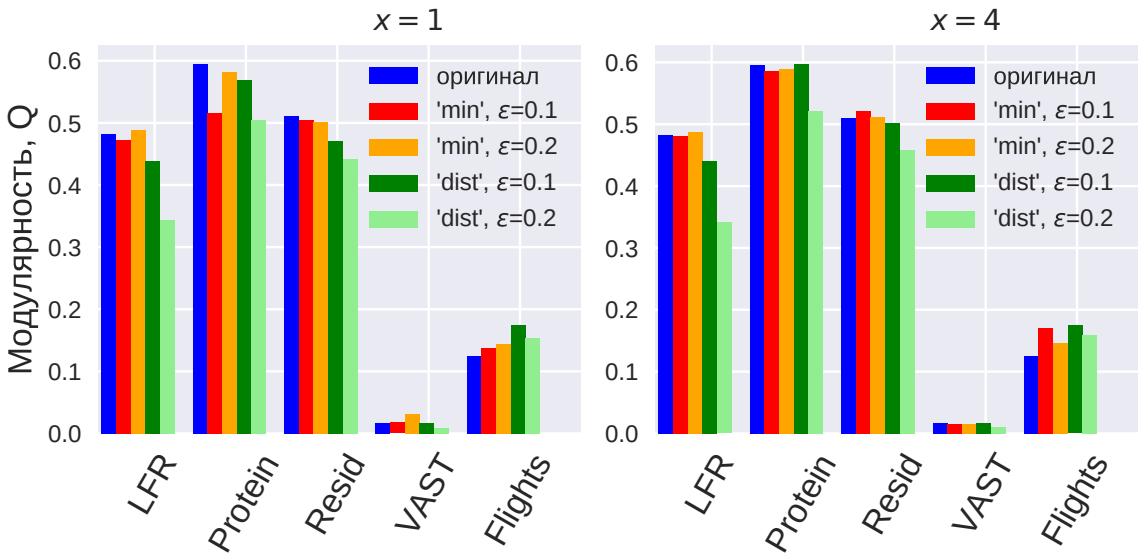

Рисунок 3.5 — Модулярность сообществ сгенерированных графов для двух схем выбора веса по умолчанию (’min’ и ’dist’) и разных амплитудах шума ε. Коэффициент масштабирования $x = 1$ (слева) и $x = 4$ (справа). Разброс значений оценен по 5 запускам.

[13] https://toreopsahl.com/datasets/#usairports

[14] Запуск с параметрами `"-N 1000 -om 3 -on 0.5 -maxk 150 -t1 2.4 -t2 1.6"`

Таблица 6 — Датасет направленных взвешенных графов. Q — модулярность [5] найденных сообществ, d — размерность вложения.

Описание графа	**имя**	**n**	**m**	**Q**	**d**
Структурная смежность иммуноглобулина[7]	Protein	95	213	0.6630	7
Отношения дружбы в группе общежития. Вес — уровень дружбы [12]	Resid	217	2672	0.5106	14
Мобильные звонки; вес ребра — количество звонков[8]	VAST	400	1562	0.5743	12
Перелеты между аэропортами США[13]	Flights	1574	28236	0.1247	31
Синтетический граф LFR[14] [52]	LFR	1000	14396	0.7209	26

3.2.3 Корректность присвоения атрибутов

Чтобы проверить корректность назначения весов метода ERGG-dwc, взвешенные графы с сообществами масштабировались, и использовалась мера модулярности, — обобщение для случая ориентированных взвешенных графов с сообществами [5]. Высокое значение этой меры является свидетельством того, что сообщества более плотно связаны внутри, чем между собой, с учетом направлений и весов ребер. Получение в результате масштабирования значений модулярности столь же высоких, как в исходном графе, поддерживает гипотезу о присвоении весов (раздел 2.4.3).

Для экспериментов использовался набор ориентированных взвешенных графов из разных доменов с найденными сообществами, а также один синтетический граф, сгенерированный LFR (таблица 6). Зафиксировав вложения с $F_1 > 0.99$ и минимальным d, применялся метод GN для аппроксимации и назначения меток, согласно описанию в 2.4.3. Сравнивалась модулярность синтетических сообществ в сгенерированных графах для двух методов выбора веса по умолчанию: минимального веса ('min') и случайного веса из распределения ('dist'). Варьировалась величина шума ε и коэффициент масштабирования x.

Сравнение схем выбора веса по умолчанию показало, что обе они работают почти одинаково при $\varepsilon \in [0; 0.2]$ с точки зрения модулярности на разных графах, как для $x = 1$, так и для $x = 4$ (рисунок 3.5). Интервал $\varepsilon \in [0.1; 0.2]$

снова оказался оптимальным, поэтому он представляет рекомендуемые значения величины шума.

При более высоком значении $\varepsilon = 0.3$ схема выбора 'dist' приводит к уменьшению модулярности (см. рисунок A.2), что подтверждает аргументы выбора минимального веса исходя из слабости связи (2.4.3).

3.2.4 Производительность

Было измерено время выполнения двух основных частей алгоритма ERGG-dwc: обучения вложения и генерации ребер при различных коэффициентах масштабирования x (рисунок 3.6). Метод вложения COMBO реализован на языке `C++` с использованием библиотеки `pthreads` для распараллеливания потоков. Для генерации графов использовалась ранняя (неоптимизированная) реализация на `python` для демонстрации соотношения времен исполнения двух частей. Поскольку сложность генерации графа равна $O(n'^2) = O(x^2 n^2)$, увеличение коэффициента масштабирования в $\sqrt{2}$ удваивает время генерации. Между тем, сложность вложения зависит от m, поэтому не коррелирует со временем генерации.

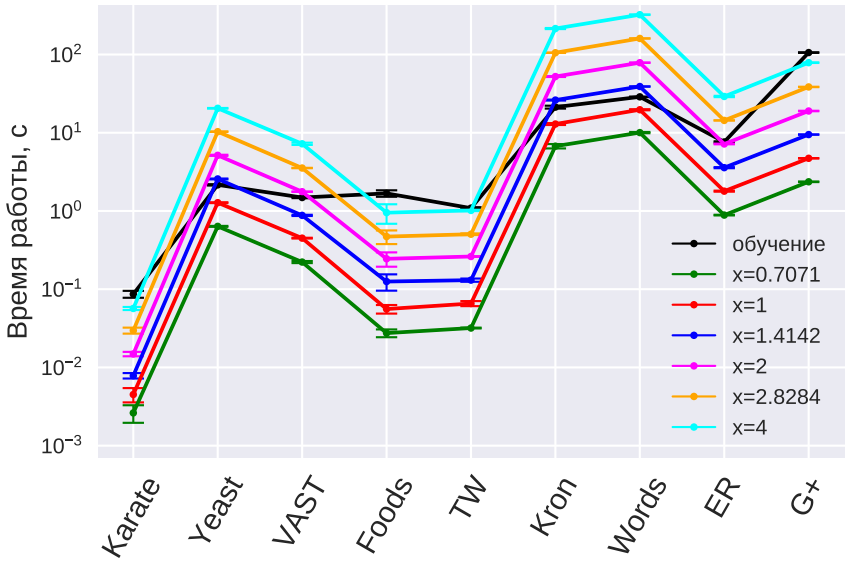

Рисунок 3.6 — Время обучения представления и генерации графа при разных коэффициентах масштабирования x. Разброс значений оценен по 5 запускам.

3.3 Экспериментальное сравнение ERGG-dwc с другими методами

В этом разделе описано экспериментальное сравнение разработанного алгоритма с наиболее близкими подходами на графах из различных доменов с применением ряда графовых характеристик.

3.3.1 Методология

Цель экспериментов — проанализировать способность ERGG-dwc и других моделей случайных графов имитировать графы из различных графовых доменов. С одной стороны, для адекватного моделирования необходимо точное воспроизведение графовых свойств. При этом ясно, что изменение лишь нескольких ребер или точное повторение последовательности степеней узлов не обеспечит достаточного разнообразия в синтетических графах для проведения надежной оценки качества графовых алгоритмов. Поэтому хорошая модель графа должна удовлетворять двум требованиям: во-первых, сгенерированные графы должны быть *похожи* на исходный граф по совокупности графовых характеристик, а во-вторых, они должны имитировать *вариабельность* реальных сетей в одном домене.

Для этой цели были выбраны известные графовые характеристики разных типов (см. 1.1.3) и проанализировано как они воспроизводятся в сгенерированных случайных графах. Далее описаны выбор характеристик, моделей и данных для тестирования.

Выбор характеристик

Поскольку не существует универсальной информативной меры похожести графов, похожесть сгенерированных графов на исходный оценивалась по набору известных характеристик. Для числовых характеристик сравнивалось отклонение среднего значения характеристики сгенерированного графа

от оригинального. Для характеристик-распределений, в отличие от числовых, возможны разные способы сравнения, поэтому были проведены качественные оценки сходства их форм.

Для оценки вариабельности сгенерированных графов сравнивался разброс по нескольким графовым характеристикам с соответствующими разбросами в реальных графах из одного домена. Когда схожесть с оригиналом по характеристике есть, разброс ее значений в идеале должен быть достаточно широким, чтобы охватить реальную вариабельность.

Топологические характеристики можно разделить на четыре группы в соответствии с классификацией в разделе 1.1.3: распределение и ассортативность для степеней вершин; распределение 3-GP, кумулятивное распределение коэффициента кластеризации и зависимость коэффициента кластеризации от степени вершины; достижимость вершин, радиус, диаметр и эффективный диаметр гигантской компоненты; спектральный радиус.

Выбор моделей

Для экспериментального исследования были отобраны две наиболее релевантные модели случайных графов помимо ERGG-dwc: Кронекеровские графы (SKG) [44] и Gscaler [49].

Что касается других моделей, проблема их использования заключается в том, что многие алгоритмы не дают представления о том, как выбирать значения своих параметров для получения графа, *наиболее похожего* на заданный, за исключением тривиальных параметров, таких как число вершин. Примерами этих алгоритмов являются: модель Forest Fire [176] (вероятности прямого и обратного распространения и еще 3 параметра), Random Walk [80] (вероятность продолжения обхода, вероятность присоединения узла), Nearest Neighbor [80] (вероятность добавления узла, количество пар узлов для подключения — введено как расширение в [37]).

Теоретически, можно выполнить поиск в пространстве параметров, руководствуясь некоторой мерой подобия, как это было сделано в сравнительной работе Алессандра Сала и др. (Alessandra Sala et al) [37], однако полный поиск в пространстве параметров может быть очень затратным, а выбор подходящей

меры близости представляет собой отдельную проблему [177]. Другое возможное решение — использовать параметры по умолчанию (как это было сделано в работе [49]), но это заведомо снижает гибкость моделей, поскольку делает графы близкого размера из разных доменов неразличимыми, в то время как известно, что графовые домены имеют очень разные свойства, в частности могут быть различены на основе 3-GP характеристик [147].

Модели графов, потенциально подходящие для имитации заданного графа, но исключенные по разным причинам: dK-графы [54] — подходят только для неориентированных графов, UpSizeR [178] — алгоритм, разработанный для масштабирования реляционных баз данных и уступающий методу Gscaler [49] для графов.

Датасет

Для тестирования были выбраны 8 направленных графов размером от тысячи до сотни тысяч вершин из различных доменов, предложенных в коллекции графов KONECT [172]. Параметры графов представлены в таблице 7.

Таблица 7 — Датасет направленных графов.

Описание графа	**домен**	**имя**	**n**	**m**
Белок-белковые взаимодействия[15]	биологический	PPI	2 239	6 452
Сеть доверия Epinion[16]	социальный	Epinion	49 288	487 183
Цитирования статей по физике высоких энергий[17]	информационный	CitHepTh	27 770	352 807
Смежность слов в англоязычных текстах[18]	информационный	Words	7 381	46 281
Мобильные звонки[19]	социальный	WU	72 146	100 974
Перелеты между аэропортами США[20]	технологический	Flights	1 574	28 236
Зависимости между классами софта JDK 1.6.0.7[21]	технологический	JDK	6 434	53 892
E-мейлы сотрудников Enron[22]	социальный	Enron	87 273	321 918

3.3.2 Измерение похожести

Все упомянутые характеристики были посчитаны для каждой из трех моделей на всех графах датасета. Далее приведены только наиболее значимые результаты; полный набор графиков содержится в приложении Б.

В таблице 8 показаны результаты измерения числовых характеристик. Для каждой модели измерялось отклонение значения (усредненного по 5 запускам) характеристики в сгенерированном графе от оригинального. В каждой ячейке присутствует символ соответствующего генератора, если это значение менее 10%. Более полные данные по разбросу значений каждой характеристики приведены в приложении Б.

В силу своего алгоритма, Gscaler почти идеально воспроизводит распределение степеней и как следствие, коэффициент Джини; n и m являются параметрами, поэтому число ребер и средняя степень задаются точно. Из сравнения остальных характеристик видно, что в точности их воспроизведения ERGG-dwc почти не уступает Gscaler: 28 и 33 вхождений соответственно. При этом по всем характеристикам оба алгоритма значительно превосходят SKG: 37 для ERGG-dwc, 49 для Gscaler и 8 вхождений для SKG.

Значение 10% было выбрано как некоторый порог по точности, чтобы показать что ERGG-dwc работает незначительно хуже Gscaler, заметно превосходя SKG. Однако, меньшая в среднем точность не является недостатком: как будет показано в следующем разделе, ERGG-dwc показывает реалистичный разброс значений характеристик, в то время как Gscaler воспроизводит их “слишком” точно.

Далее рассмотрим результаты измерений характеристик-распределений с разбором каждой из них.

[15] http://konect.uni-koblenz.de/networks/maayan-figeys

[16] http://konect.uni-koblenz.de/networks/epinions

[17] http://konect.uni-koblenz.de/networks/cit-HepTh

[18] http://www.weizmann.ac.il/mcb/UriAlon/sites/mcb.UriAlon/files/uploads/CollectionsOfComplexNetwroks/darwinbookinter_st.txt

[19] http://www.pnas.org/content/suppl/2010/10/15/1013140107.DCSupplemental/SD02.txt

[20] http://konect.uni-koblenz.de/networks/opsahl-usairport

[21] http://konect.uni-koblenz.de/networks/subelj_jdk

[22] http://konect.uni-koblenz.de/networks/enron

Таблица 8 — Числовые характеристики. Для каждой модели измерялось отклонение значения (усредненного по 5 запускам) характеристики в сгенерированном графе от оригинального. В каждой ячейке присутствует символ соответствующего генератора ('E' – ERGG-dwc, 'G' – Gscaler, 'S' – SKG), если это отклонение меньше 10%, или прочерк.

Характеристика	PPI	Epinions	CitHepTh	Words	WU	Flights	JDK	Enron
число ребер	-GS	EG-	-G-	EGS	EG-	EG-	EG-	-GS
средняя степень	EG-	EG-	-GS	-GS	-G-	EG-	EG-	-GS
ассортативность степени	EG-	—	-G-	EG-	-G-	E–	EG-	-G-
взаимность ребер	—	—	—	-G-	—	—	—	—
коэф. Джини распределения степеней	EG-	EG-	EG-	EG-	EG-	EG-	EG-	EG-
средний коэф. кластеризации	—	E–	—	-G-	—	—	—	—
косинусная близость 3-GP	EG-	EG-	EG-	EGS	—	—	EG-	EG-
эфф. диаметр	EG-	-GS	E–	EG-	—	—	—	—
спектральный радиус	-G-	EG-	EG-	EG-	-G-	EG-	EG-	EG-

Степень вершины

При измерении распределения входящих и исходящих степеней вершин было обнаружено, что Gscaler повторяет их почти идеально, в то время как ERGG-dwc воспроизводит форму распределения степеней (в логарифмической шкале) не точно, но намного ближе к оригиналу, чем SKG (рисунок 3.7). Кривые распределения степеней на графиках для SKG часто имеют осцилляции, что отражено в литературе [179].

ERGG-dwc имеет проблему с узлами малой степени: число узлов со степенью 1 меньше оригинала почти во всех доменах.

Ассортативность степеней вершин хорошо улавливается как ERGG-dwc, так и Gscaler, тогда как для SKG графики ассортативности больше похожи друг

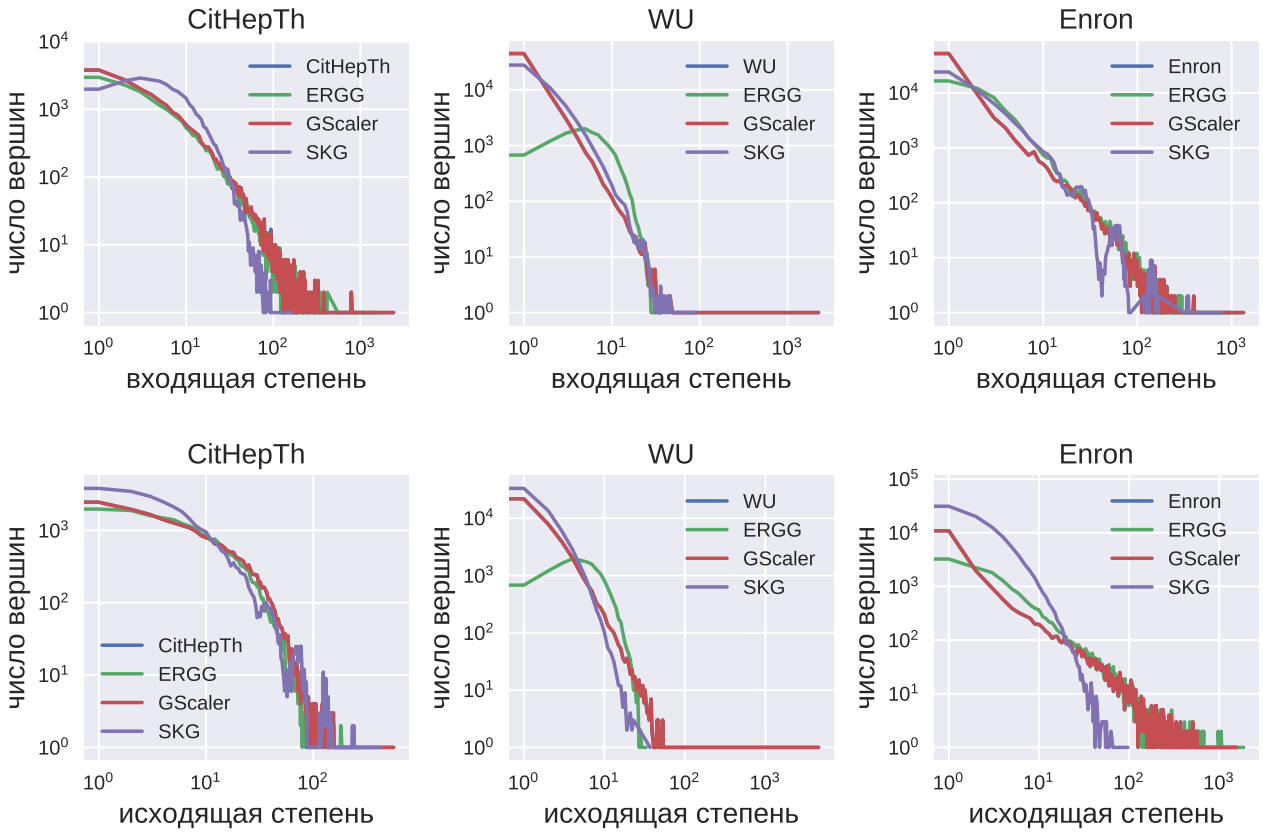

Рисунок 3.7 — Gscaler практически идеально повторяет распределение входящих и исходящих степеней. Для ERGG-dwc распределение степеней воспроизводится значительно ближе к оригиналу чем SKG.

на друга, чем на оригинальные (рисунок 3.8). То же самое можно сказать и об ассортативности входящих и исходящих степеней.

Подграфы

Коэффициент кластеризации в виде кумулятивного распределения показывает, сколько в графе существует узлов с определенным коэффициентом кластеризации. На рисунке 3.9 (верхние 2 ряда) можно видеть, что SKG не в состоянии смоделировать высокий коэффициент кластеризации, в то время как результаты Gscaler и ERGG-dwc сильно различаются по доменам (рисунок 3.9, вверху). ERGG-dwc демонстрирует хорошее соответствие распределению коэффициента кластеризации на графах Epinions и CitHepTh, Gscaler — на графе

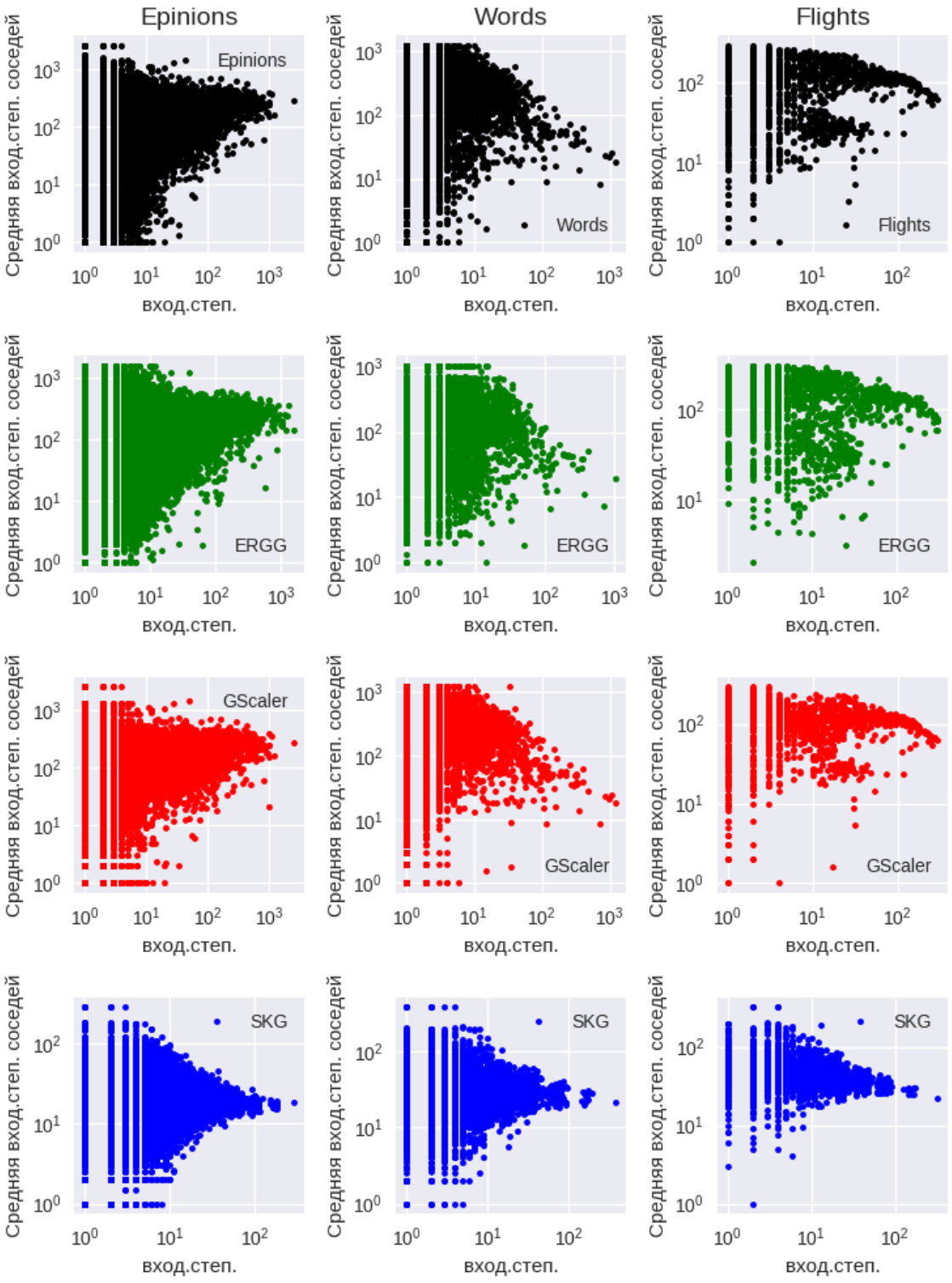

Рисунок 3.8 — Ассортативность входящих степеней хорошо улавливается ERGG-dwc и Gscaler, но не SKG.

Words. Кроме того, есть много графов (WU, Flights, JDK, Enron), чьи коэффициенты кластеризации не были хорошо воспроизведены ни одной моделью.

Аналогичная картина возникает при построении зависимости коэффициента кластеризации от степени узла (рисунок 3.9, нижние 2 ряда). Качество воспроизведения этой характеристики для ERGG-dwc и Gscaler коррелирует с предыдущим случаем. Gscaler часто не улавливает монотонный степенной закон зависимости коэффициента кластеризации от степени узла.

3-GP воспроизводится ERGG-dwc и Gscaler довольно хорошо практически для всех графов, кроме графов Flights (есть неточности) и WU (не воспроизводится).

Связность

Сходство графиков достижимости вершин в сгенерированных графах с оригиналами также отличается в разных доменах. ERGG-dwc и Gscaler показывают наиболее близкие к оригиналам результаты по достижимости для различных графов: Epinions и CitHepTh для ERGG-dwc, Words for Gscaler. Однако для JDK-графа Gscaler почти повторяет оригинальный график достижимости вершин и коэффициент кластеризации от степени вершины, но плохо воспроизводит кумулятивное распределение коэффициента кластеризации. SKG обычно работает не лучше двух других генераторов, но все же показывает относительно хороший результат для PPI, Epinions и Enron.

На рисунке 3.12 показано как воспроизводится радиус, диаметр и эффективный диаметр гигантской компоненты графа. Результаты трудно интерпретировать в пользу какого-либо из генераторов.

3.3.3 Измерение вариабельности

Вариабельность так же оценивалась по числовым характеристикам и некоторым характеристикам-распределениям. Следуя подходу С. Морено и соавторов (S. Moreno et al) [180], были взяты распределение степеней, коэффици-

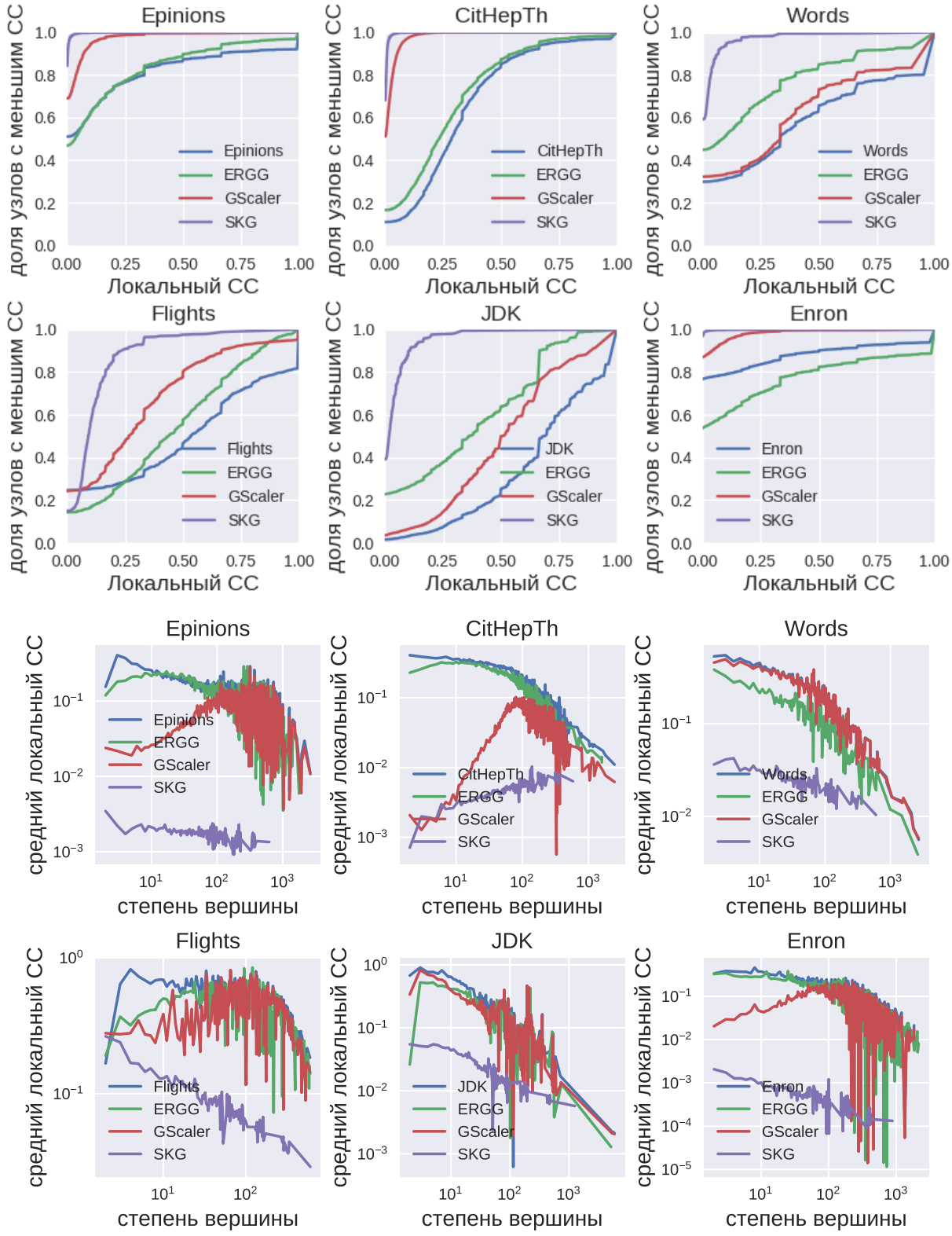

Рисунок 3.9 — Кумулятивное распределение коэффициент кластеризации (верхние 2 ряда), коэффициент кластеризации от степени узла (нижние 2 ряда). SKG не воспроизводит коэффициент кластеризации, Gscaler и ERGG-dwc показывают разные результаты.

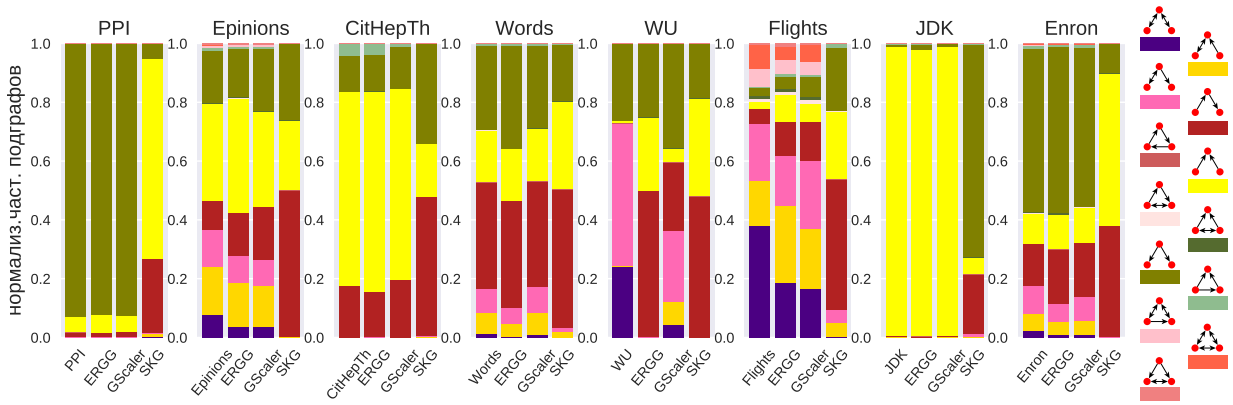

Рисунок 3.10 — 3-GP воспроизводится ERGG-dwc и Gscaler для 7 из 8 тестовых графов, кроме WU — возможно, из-за его низкой плотности (средняя степень 1.4). SKG не улавливает 3-GP.

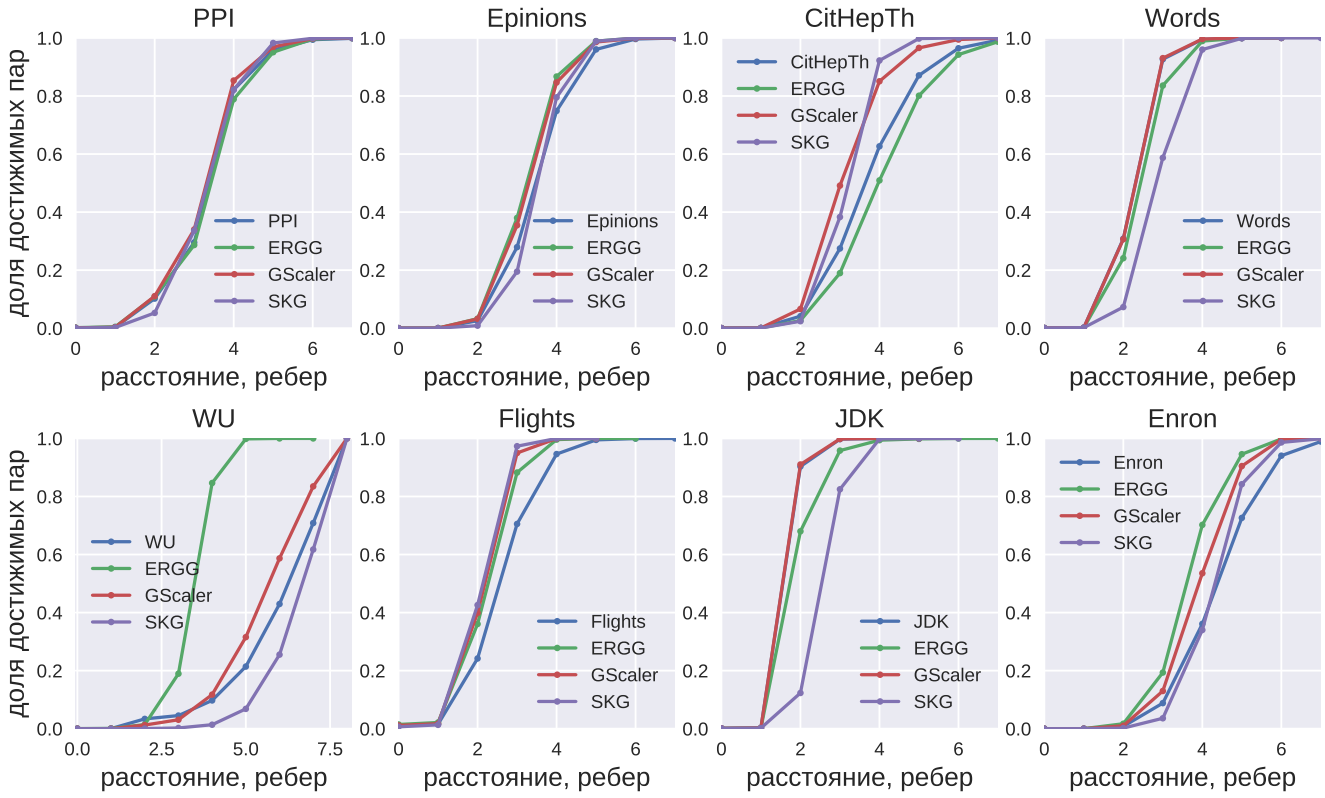

Рисунок 3.11 — Достижимость вершин. Точность воспроизведения варьируется для разных доменов. Gscaler чаще ближе к оригиналу, чем ERGG-dwc (в 5 из 8 случаев).

ент кластеризации и достижимости вершин. Дополнительно было использовано 3-GP ввиду его домен-специфичности [19].

В реальных графовых доменах довольно трудно найти выборку одинаковых по размеру графов для оценки разброса их характеристик. Поэтому в

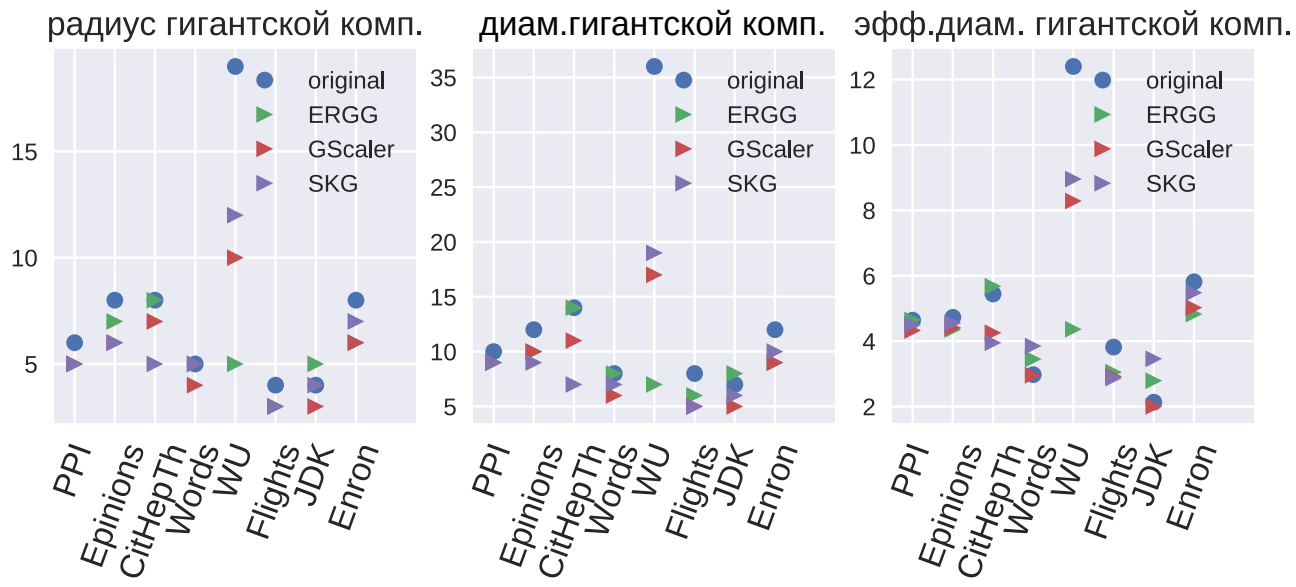

Рисунок 3.12 — Радиус, диаметр и 90%-эффективный диаметр гигантской компоненты. Результаты неоднозначные.

качестве датасета использовалась коллекция эго-сетей twitter[23], откуда было выбрано 15 графов, близких по числу узлов ($n \in [170; 180]$) и числу ребер ($m \in [2000; 3000]$). Набор сгенерированных данных был построен путем подгонки модели к одному случайному графу из датасета, и запуска генератора 15 раз. Результат по числовым характеристикам показан на рисунке 3.13, по распределениям на рисунке Б.21.

Диаграммы разброса числовых характеристик показывают, что хотя Gscaler и показывает высокую точность по некоторым характеристикам, вариабельность этих результатов нулевая или существенно меньше, чем вариабельность исходного датасета. В то же время ERGG-dwc показывает вариабельность, гораздо более близкую к исходной. В терминах дисперсии (см. таблицу 9), ERGG-dwc по 6 из 8 характеристикам имеет дисперсию не менее 40% от оригинальной, в то время как у двух других методов дисперсия не превосходит 8%.

На рисунке Б.21 можно видеть, что ERGG-dwc (второй ряд сверху) имитирует аналогичную вариабельность распределений входящих и исходящих степеней, графика достижимости, кумулятивного распределения коэффициента кластеризации и более низкий разброс в 3-GP. Gscaler (третий ряд) демонстрирует почти нулевую изменчивость в распределении степеней, 3-GP

[23] http://snap.stanford.edu/data/egonets-Twitter.html

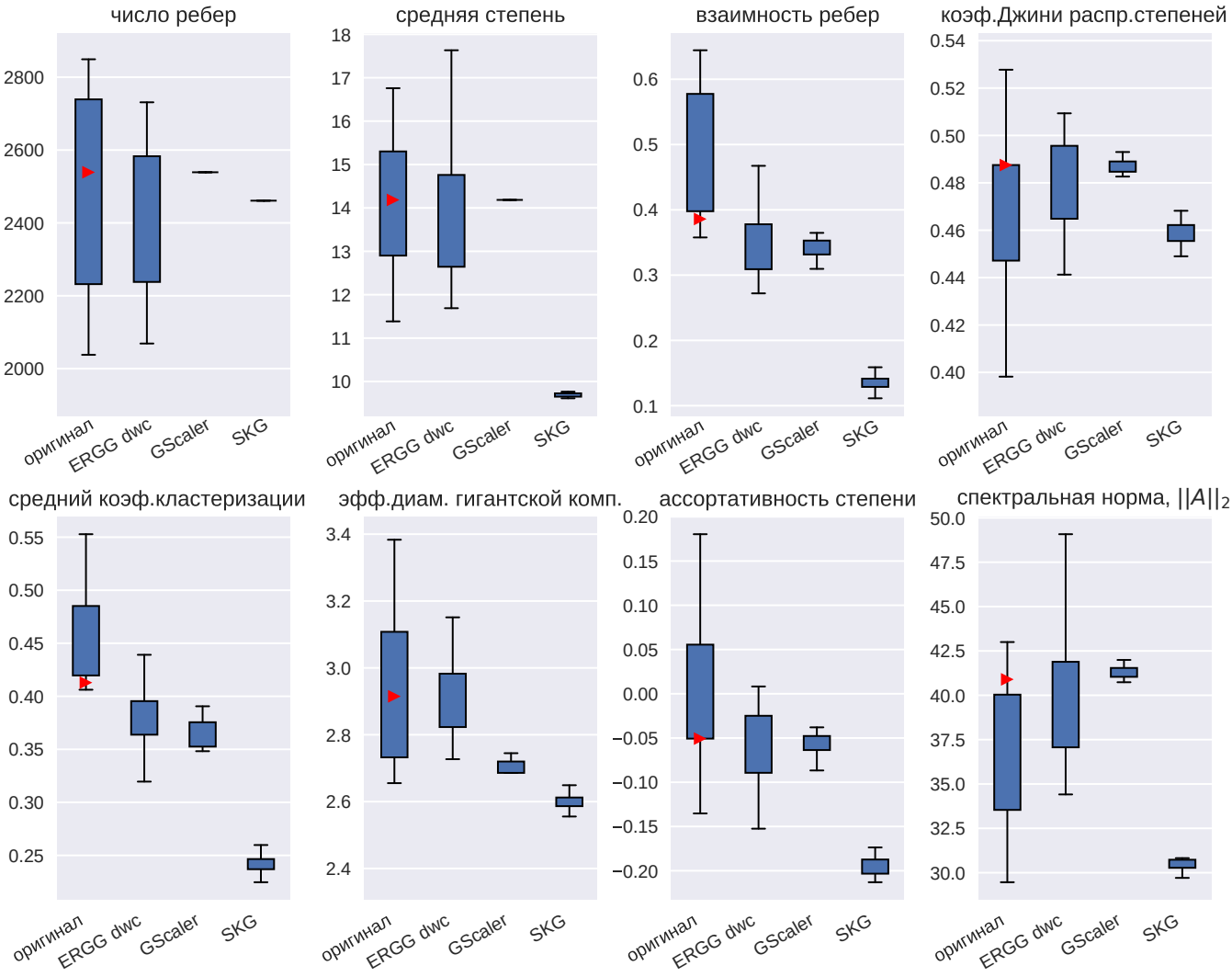

Рисунок 3.13 — Сравнение вариабельности числовых характеристик при имитации домена. Оригинальный датасет: 15 эго-сетей twitter с $n \in [170; 180]$ и $m \in [2000; 3000]$; ERGG-dwc; Gscaler; SKG. Красной стрелкой отмечено значение характеристики для графа, на котором происходило обучение.

и графике достижимости. Все его сгенерированные графы идентичны по количеству узлов и ребер в силу своего алгоритма и слишком похожи друг на друга.

Таблица 9 — Сравнение дисперсий вариабельностей числовых характеристик при имитации домена. На 6 из 8 характеристик дисперсия ERGG-dwc не менее 40% от исходной, у Gscaler и SKG дисперсия не более 8%.

Характеристика	**ориг. дисперсия**	**ERGG-dwc**	**Gscaler**	**SKG**
число ребер	78222	**105022 (134%)**	0 (0.0%)	0 (0.0%)
средняя степень	2.636	**3.1767 (120%)**	0.0000 (0.0%)	0.0011 (0.0%)
ассортативность степени	0.009	**0.0041 (46.0%)**	0.0001 (0.8%)	0.0002 (2.1%)
взаимность ребер	$7.988{\cdot}10^{-3}$	$1.652{\cdot}10^{-3}$ (20.7%)	$3.595{\cdot}10^{-4}$ (4.5%)	$3.202{\cdot}10^{-5}$ (0.4%)
коэф. Джини распределения степеней	$9.962{\cdot}10^{-4}$	$\mathbf{4.058}{\cdot}10^{-4}$ **(40.7%)**	$7.948{\cdot}10^{-6}$ (0.8%)	$4.313{\cdot}10^{-5}$ (4.3%)
средний коэф. кластеризации	0.002	**0.0009 (51.4%)**	0.0001 (7.7%)	0.0000 (1.8%)
эфф. диаметр	0.107	0.0043 (4.0%)	0.0015 (1.4%)	0.0001 (0.1%)
спектральный радиус	15.217	**20.6935 (136%)**	0.0963 (0.6%)	0.0839 (0.6%)

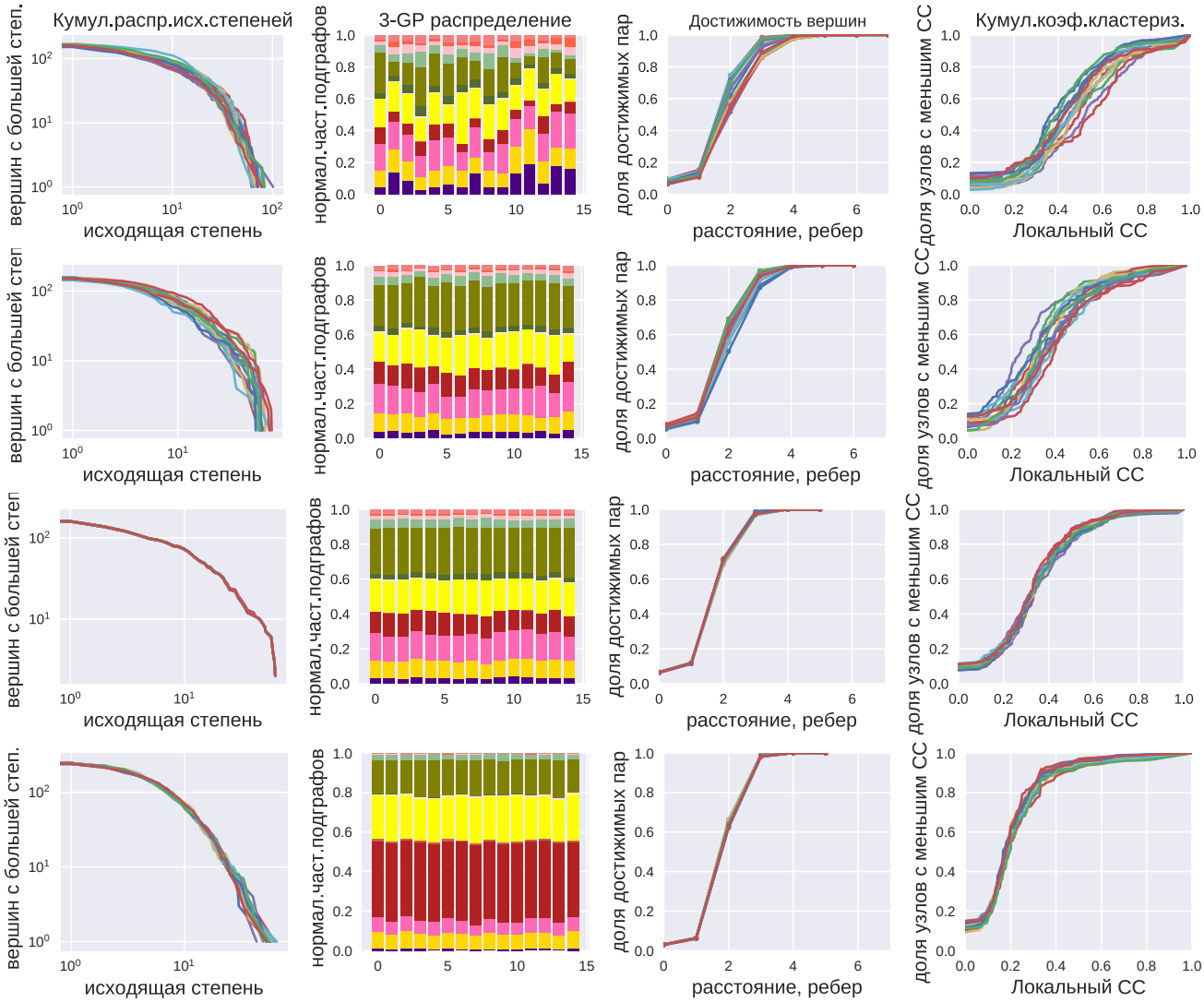

Рисунок 3.14 — Сравнение вариабельности характеристик-распределений при имитации домена. Сверху вниз: оригинальный датасет (15 эго-сетей twitter с $n \in [170; 180]$ и $m \in [2000; 3000]$); ERGG-dwc; Gscaler; SKG.

3.4 Пример использования: тестирование качества работы алгоритмов

В этом разделе показан пример использования метода ERGG-dwc для надежной проверки стабильности работы алгоритмов интеллектуального анализа графов. Практическая проблема заключается в отсутствии достаточного количества графовых данных для надежного тестирования алгоритмов: при наличии малого количества результатов работы алгоритма, сделать надежный вывод о качестве его работы затруднительно. Поскольку метод ERGG-dwc позволяет генерировать искусственные графы, похожие на данный и обладающие достаточной вариабельностью, то есть, репрезентативную выборку, то его можно использовать для тестирования стабильности работы алгоритмов.

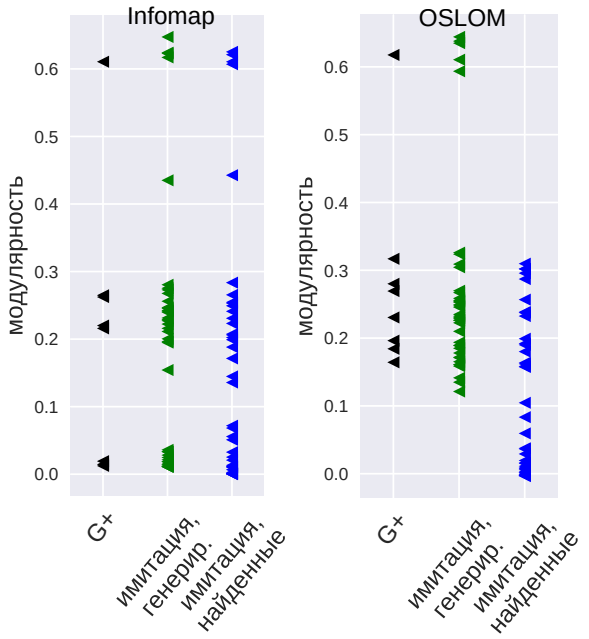

Рисунок 3.15 — Иллюстрация проверки стабильности работы алгоритма поиска сообществ. Стабильность означает сохранение высокого качества (модулярности) на выборке искусственных графов. На выборке, сгенерированной ERGG-dwc для метода Infomap модулярность сообществ, найденных в сгенерированных графах остается на том же уровне (есть стабильность), для метода OSLOM модулярность найденных сообществ сильно ниже (нет стабильности).

Для этого из набора эго-сетей Google-plus было выбрано 8 близких по количеству узлов графов ($n \in [450; 500]$). В качестве алгоритмов анализа графов

были взяты два метода поиска сообществ: OSLOM и Infomap[24]. Для оценки качества покрытия сообществами использовалась модулярность. Вначале алгоритмы запускаются на исходных графах (рисунок 3.15, черные треугольники). Затем для каждого графа генерируется 5 имитаций с помощью ERGG-dwc и проверяется что сгенерированные сообщества имеют близкие значения модулярности (зеленые треугольники). После этого запускается поиск сообществ на сгенерированных графах и снова измеряется модулярность (синие треугольники).

Ожидается, что на тестовой выборке графов хороший алгоритм должен показывать стабильные результаты. Сравнивая полученные значения модулярности с результатами на исходных графах, можно заключить что Infomap показывает стабильный результат в смысле модулярности, в то время как результаты OSLOM не стабильны: модулярность сообществ, найденных в сгенерированных графах заметно меньше чем в оригинальных, несмотря на присутствие в этих графах сообществ с такой же высокой модулярностью. Это означает что качество найденных сообществ методом OSLOM сильно падает на похожих графах.

3.5 Выводы к третьей главе

Экспериментально показано, что ориентированные графы из разных доменов могут быть успешно вложены в низкоразмерное пространство с помощью модифицированного метода COMBO. Качество вложения подразумевает, что ребра графа могут быть восстановлены с высокой мерой $F_1 > 0.99$ при некоторой минимальной размерности пространства d, которая зависит от графа.

Метод аппроксимации распределения векторов представления GN с добавлением шума позволяет генерировать графы с близкими к исходным распределениями степеней и 3-GP. Корректность способа присвоения меток сообществ и весов ребер в генерируемых графах подтверждается сохранением исходных высоких значений модулярности получающихся сообществ.

Затем была проанализирована способность ERGG-dwc воспроизводить известные графовые признаки в сравнении с другими современными моделями

[24] http://www.mapequation.org/code.html

случайных графов — Gscaler и SKG. Сравнивалась похожесть и вариабельность сгенерированных графов с точки зрения различных графовых характеристик.

Было обнаружено, что Gscaler достаточно точно воспроизводит распределение степеней и корреляции степеней вершин. Gscaler и ERGG-dwc воспроизводят большинство протестированных графовых характеристик, хотя их точность варьируется в разных доменах. Эти два алгоритма превосходят SKG на рассмотренных характеристиках, более того, имитации SKG для графов из разных доменов больше похожи друг на друга, чем на исходные графы.

Далее было показано, что ERGG обеспечивает вариабельность графов, близкую к естественной вариабельности графов из одного домена, в то время как Gscaler имеет тенденцию генерировать практически идентичные графы, уменьшая репрезентативность набора синтетических данных.

Таким образом была показана способность ERGG-dwc имитировать графы, похожие на данный, с двумя условиями: 1) похожесть на исходный граф по ряду известных характеристик, 2) вариабельность по ряду характеристик, отражающая вариабельность внутри одного домена. Учитывая эти две особенности, ERGG-dwc может применяться на практике для создания искусственных датасетов для надежной проверки статистической значимости алгоритмов майнинга графов. Эта возможность была продемонстрирована на примере алгоритмов поиска сообществ.

Возможность генерирования графов контролируемого размера с сохранением исходных признаков позволяет также тестировать масштабируемость алгоритмов.

К сожалению, ограничение производительности алгоритма не позволяет применять его для больших графов из-за квадратичной сложности генерации ребер графа. Один из способов его преодолеть заключается в оптимизации поиска всех пар узлов (i,j), удовлетворяющих условию $s_{ij} > t_G$. Задача похожа на проблему поиска ближайшего соседа, поэтому потенциально могут быть использованы некоторые подходы для преодоления полного перебора пар. Другая возможность состоит в том, чтобы сузить пространство поиска до пар, соответствующих ребрам исходного графа, что позволило бы находить ребра E' за $O(x^2 m)$ шагов.

Второй существенный недостаток кроется в законе масштабирования генерируемых графов, в котором количество ребер растет пропорционально квадрату вершин: $m \propto x^2 n^2$, в то время как на практике наблюдается степенной

рост $m \sim n^{\alpha}$ со степенью $1 < \alpha < 2$. Поэтому при больших коэффициентах масштабирования x генерируемые графы теряют реалистичность, что выражается, например, в искажении формы распределения степеней. Эта закономерность обусловлена способом генерации ребер на основе порога $s_{ij} > t_G$ и не зависит от способа аппроксимации распределения. Для преодоления этого ограничения в рамках подхода ERGG требуется другая модель представления графа. В контексте же его реализации в виде ERGG-dwc на практике рекомендуется использовать только небольшие коэффициенты масштабирования x (не более 10).

Заключение

Основные результаты работы заключаются в следующем.

1. Предложен новый подход ERGG к генерации случайных направленных графов, похожих на данный, основанный на вложении графа в пространство размерности, много меньшей числа его вершин.
2. В рамках подхода ERGG предложен метод ERGG-dwc, решающий задачу генерации графов, похожих на данный и удовлетворяющих требованиям: автоматическое обучение на заданном графе, контролируемый размер генерируемых графов, поддержка направленных ребер, взвешенных ребер и структуры сообществ. Соответствие метода ERGG-dwc требованиям похожести и вариабельности генерируемых графов подтверждено экспериментально, также показана корректность назначения структуры сообществ и весов ребер. Доказаны теоремы о вычислительной сложности и масштабировании. Налагаемые ими ограничения не являются существенными для применимости метода.
3. Создана программная система, в которой реализован прототип ERGG-dwc и проведено его экспериментальное сравнение с другими современными методами. Показано, что ERGG-dwc не уступает другим методам в похожести генерируемых графов, но превосходит их по вариабельности.

Благодарности

Хотелось бы выразить благодарность Турдакову Денису за научное руководство, Коршунову Антону за активное содействие в получении результатов, Козлову Илье за развитие идей а также другим коллегам отдела информационных систем ИСП РАН, принимавшим участие в обсуждениях в процессе работы.

Список сокращений и условных обозначений

Следующие обозначения используются по умолчанию, если не указано обратного.

$G = (N,E)$	граф с множеством вершин N и ребер $E \subseteq N \times N$
$n = \|N\|$	количество вершин
$m = \|E\|$	количество ребер
i, j, k, l	вершина
d_i	степень вершины i
d_i^{in}	входящая степень вершины i
d_i^{out}	исходящая степень вершины i
s_i	мощность вершины i
s_i^{in}	мощность вершины i по входящим ребрам
s_i^{out}	мощность вершины i по исходящим ребрам
$\mathcal{C}_i$	метка принадлежности вершины i сообществам графа
(i,j)	ребро между вершинами i и j (с учетом направления для ориентированного графа)
w_{ij}	вес ребра (i,j);
p_{ij}	вероятность ребра (i,j)
A_{ij}	матрица смежности
W_{ij}	взвешенная матрица смежности
3-GP	распределение подграфов размера 3: L^2-нормализованный вектор частот встречаемости всех возможных подграфов размера 3 (с точностью до изоморфизма) в данном графе [19]
DD	***degree distribution***, распределение степеней вершин
CC	***clustering coefficient***, коэффициент кластеризации
MCMC	***Markov Chain Monte Carlo***, метод Монте-Карло на цепи Маркова
ER	***Erdős-Rényi model***, модель Эрдеша-Реньи
PA	***Preferential Attachment***, предпочтительное присоединение
ReCoN	***Replicating of Complex Networks***, репликация сложных сетей
SKG	***Stochastic Kronecker Graphs***, стохастические Кронекеровские графы

MFNG	***Multi-Fractal Network Generator***, генератор мультифрактальных сетей
GIRG	***Geometric Inhomogeneous Random Graphs***, геометрические неоднородные случайные графы
ERGG	***Embedding Based random Graph Generation***, генерация случайных графов на основе вложения графа
SBM	***Stochastic blockmodel***, стохастические блочные модели
RTG	***Random typing graphs***, графы случайного набора
LPC	***Latent position cluster***, модель кластера со скрытой позицией
AGM	***Affiliation graph model***, модели графа принадлежности сообществам
LFR	***Lancichinetti-Fortunato-Radicci benchmark***, эталонные графы Ланчичинетти-Фортунато-Радиччи
ERGM	***Exponential random graph model***, экспоненциальная модель случайных графов
MUSKETEER	***Multiscale Network Generation***, мультимасштабный генератор сетей
BLM	***Bilinear Link Model***, билинейная реберная модель
LINE	***Large-scale Information Network Embedding***, метод вложения больших информационных сетей

Список литературы

1. *Newman Mark EJ.* The structure and function of complex networks // *SIAM review.* — 2003. — Vol. 45, no. 2. — Pp. 167–256.

2. *Erdős P, Rényi A.* On random graphs I // *Publ. Math. Debrecen.* — 1959. — Vol. 6. — Pp. 290–297.

3. *Barabási Albert-László, Albert Réka.* Emergence of scaling in random networks // *science.* — 1999. — Vol. 286, no. 5439. — Pp. 509–512.

4. *Backstrom Lars, Dwork Cynthia, Kleinberg Jon.* Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography // Proceedings of the 16th international conference on World Wide Web / ACM. — 2007. — Pp. 181–190.

5. *Drobyshevskiy Mikhail, Korshunov Anton, Turdakov Denis.* Parallel modularity computation for directed weighted graphs with overlapping communities // *Труды Института системного программирования РАН.* — 2016. — Vol. 28, no. 6. — Pp. 153–170.

6. *Drobyshevskiy Mikhail, Korshunov Anton, Turdakov Denis.* Learning and scaling directed networks via graph embedding // Joint European Conference on Machine Learning and Knowledge Discovery in Databases / Springer. — 2017. — Pp. 634–650.

7. *Drobyshevskiy Mikhail, Turdakov Denis, Kuznetsov Sergey.* Reproducing Network Structure: A Comparative Study of Random Graph Generators // Ivannikov ISPRAS Open Conference (ISPRAS), 2017 / IEEE. — 2017. — Pp. 83–89.

8. *Gilbert Edgar N.* Random graphs // *The Annals of Mathematical Statistics.* — 1959. — Vol. 30, no. 4. — Pp. 1141–1144.

9. *Barrat Alain, Barthèlemy Marc, Vespignani Alessandro.* Dynamical Processes on Complex Networks. — Cambridge University Press, 2008.

10. *Faragó András.* Structural properties of random graph models // Proceedings of the Fifteenth Australasian Symposium on Computing: The Australasian Theory-Volume 94 / Australian Computer Society, Inc. — 2009. — Pp. 131–138.

11. *Newman Mark.* Networks: an introduction. — Oxford university press, 2010.

12. *Coolen Ton, Annibale Alessia, Roberts Ekaterina.* Generating random networks and graphs. — Oxford University Press, 2017.

13. *Kolaczyk Eric D.* Statistical Analysis of Network Data: Methods and Models. — 1st edition. — Springer Publishing Company, Incorporated, 2009.

14. *Ebel Holger, Mielsch Lutz-Ingo, Bornholdt Stefan.* Scale-free topology of e-mail networks // *Physical review E.* — 2002. — Vol. 66, no. 3. — P. 035103.

15. Double Pareto lognormal distributions in complex networks / Zheng Fang, Jie Wang, Benyuan Liu, Weibo Gong // Handbook of Optimization in Complex Networks. — Springer, 2012. — Pp. 55–80.

16. *Newman Mark EJ.* Assortative mixing in networks // *Physical review letters.* — 2002. — Vol. 89, no. 20. — P. 208701.

17. *Newman Mark EJ, Strogatz Steven H, Watts Duncan J.* Random graphs with arbitrary degree distributions and their applications // *Physical review E.* — 2001. — Vol. 64, no. 2. — P. 026118.

18. Characterization of complex networks: A survey of measurements / L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, Paulino Ribeiro Villas Boas // *Advances in physics.* — 2007. — Vol. 56, no. 1. — Pp. 167–242.

19. Mining large networks with subgraph counting / Ilaria Bordino, Debora Donato, Aristides Gionis, Stefano Leonardi // 2008 Eighth IEEE International Conference on Data Mining / IEEE. — 2008. — Pp. 737–742.

20. Network motifs: simple building blocks of complex networks / Ron Milo, Shai Shen-Orr, Shalev Itzkovitz et al. // *Science.* — 2002. — Vol. 298, no. 5594. — Pp. 824–827.

21. *Watts Duncan J, Strogatz Steven H.* Collective dynamics of 'small-world' networks // *nature.* — 1998. — Vol. 393, no. 6684. — P. 440.

22. *Faloutsos Michalis, Faloutsos Petros, Faloutsos Christos.* On power-law relationships of the internet topology // ACM SIGCOMM computer communication review / ACM. — Vol. 29. — 1999. — Pp. 251–262.

23. *Erdos Paul, Rényi Alfréd.* On the evolution of random graphs // *Publ. Math. Inst. Hung. Acad. Sci.* — 1960. — Vol. 5, no. 1. — Pp. 17–60.

24. *Newman Mark EJ.* Modularity and community structure in networks // *Proceedings of the national academy of sciences.* — 2006. — Vol. 103, no. 23. — Pp. 8577–8582.

25. *Yang Jaewon, Leskovec Jure.* Defining and evaluating network communities based on ground-truth // *Knowledge and Information Systems.* — 2015. — Vol. 42, no. 1. — Pp. 181–213.

26. *Brouwer Andries E, Haemers Willem H.* Spectra of graphs. — Springer Science & Business Media, 2011.

27. *Van Mieghem Piet, Omic Jasmina, Kooij Robert.* Virus spread in networks // *IEEE/ACM Transactions on Networking (TON).* — 2009. — Vol. 17, no. 1. — Pp. 1–14.

28. *Pothen Alex, Simon Horst D, Liou Kang-Pu.* Partitioning sparse matrices with eigenvectors of graphs // *SIAM journal on matrix analysis and applications.* — 1990. — Vol. 11, no. 3. — Pp. 430–452.

29. *Eikmeier Nicole, Gleich David F.* Revisiting power-law distributions in spectra of real world networks // Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining / ACM. — 2017. — Pp. 817–826.

30. *Hernández Javier Martın, Van Mieghem Piet.* Classification of graph metrics // *Delft University of Technology, Tech. Rep.* — 2011. — Pp. 1–8.

31. *Leskovec Jure, Kleinberg Jon, Faloutsos Christos.* Graph evolution: Densification and shrinking diameters // *ACM Transactions on Knowledge Discovery from Data (TKDD).* — 2007. — Vol. 1, no. 1. — P. 2.

32. *McGlohon Mary, Akoglu Leman, Faloutsos Christos.* Weighted graphs and disconnected components: patterns and a generator // Proceedings of the 14th

ACM SIGKDD international conference on Knowledge discovery and data mining / ACM. — 2008. — Pp. 524–532.

33. *Yang Jaewon, Leskovec Jure.* Structure and overlaps of ground-truth communities in networks // *ACM Transactions on Intelligent Systems and Technology (TIST).* — 2014. — Vol. 5, no. 2. — P. 26.

34. *Frieze Alan, Karoński Michał.* Introduction to random graphs. — Cambridge University Press, 2015.

35. Statistical properties of community structure in large social and information networks / Jure Leskovec, Kevin J Lang, Anirban Dasgupta, Michael W Mahoney // Proceedings of the 17th international conference on World Wide Web / ACM. — 2008. — Pp. 695–704.

36. A comparative study of social network models: Network evolution models and nodal attribute models / Riitta Toivonen, Lauri Kovanen, Mikko Kivelä et al. // *Social Networks.* — 2009. — Vol. 31, no. 4. — Pp. 240–254.

37. Measurement-calibrated graph models for social network experiments / Alessandra Sala, Lili Cao, Christo Wilson et al. // Proceedings of the 19th international conference on World wide web / ACM. — 2010. — Pp. 861–870.

38. *Gilbert GN, Hamill L.* Social circles: A simple structure for agent-based social network models // *Journal of Artificial Societies and Social Simulation.* — 2009. — Vol. 12, no. 2.

39. *Bonato Anthony.* A survey of properties and models of on-line social networks // Proc. of the 5th International Conference on Mathematical and Computational Models, ICMCM / Citeseer. — 2009.

40. *Edward M. Lazzarin Raj Jain.* An overview of the analysis of online social networks // *?* — 2011.

41. *Pofsneck L, Hofmann Henning, Buettner Ricardo.* Physical theories of the evolution of online social networks: a discussion impulse // *dynamical systems.* — 2012. — Vol. 22. — P. 4.

42. *Chakrabarti Deepayan, Zhan Yiping, Faloutsos Christos.* R-MAT: A recursive model for graph mining // Proceedings of the 2004 SIAM International Conference on Data Mining / SIAM. — 2004. — Pp. 442–446.

43. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication / Jurij Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos // European Conference on Principles of Data Mining and Knowledge Discovery / Springer. — 2005. — Pp. 133–145.

44. Kronecker graphs: An approach to modeling networks / Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg et al. // *Journal of Machine Learning Research.* — 2010. — Vol. 11, no. Feb. — Pp. 985–1042.

45. *Palla Gergely, Lovász László, Vicsek Tamás.* Multifractal network generator // *Proceedings of the National Academy of Sciences.* — 2010.

46. *Nickel Christine Leigh Myers.* Random dot product graphs a model for social networks: Ph.D. thesis / The Johns Hopkins University. — 2006.

47. *Akoglu Leman, Faloutsos Christos.* RTG: a recursive realistic graph generator using random typing // Joint European Conference on Machine Learning and Knowledge Discovery in Databases / Springer. — 2009. — Pp. 13–28.

48. Hyperbolic geometry of complex networks / Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak et al. // *Physical Review E.* — 2010. — Vol. 82, no. 3. — P. 036106.

49. *Zhang JW, Tay YC.* GSCALER: Synthetically Scaling A Given Graph. // EDBT. — 2016. — Pp. 53–64.

50. Mobile call graphs: beyond power-law and lognormal distributions / Mukund Seshadri, Sridhar Machiraju, Ashwin Sridharan et al. // Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining / ACM. — 2008. — Pp. 596–604.

51. *Wegner Anatol.* Random Graphs with Motifs. — 2011.

52. *Lancichinetti Andrea, Fortunato Santo.* Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities // *Physical Review E.* — 2009. — Vol. 80, no. 1. — P. 016118.

53. Distributed generation of billion-node social graphs with overlapping community structure / Kyrylo Chykhradze, Anton Korshunov, Nazar Buzun et al. // Complex Networks V. — Springer, 2014. — Pp. 199–208.

54. Systematic topology analysis and generation using degree correlations / Priya Mahadevan, Dmitri Krioukov, Kevin Fall, Amin Vahdat // ACM SIGCOMM Computer Communication Review / ACM. — Vol. 36. — 2006. — Pp. 135–146.

55. *Ying Xiaowei, Wu Xintao.* Graph generation with prescribed feature constraints // Proceedings of the 2009 SIAM International Conference on Data Mining / SIAM. — 2009. — Pp. 966–977.

56. Generating scaled replicas of real-world complex networks / Christian L Staudt, Michael Hamann, Ilya Safro et al. // International Workshop on Complex Networks and their Applications / Springer, Cham. — 2016. — Pp. 17–28.

57. *Park Himchan, Kim Min-Soo.* TrillionG: A trillion-scale synthetic graph generator using a recursive vector model // Proceedings of the 2017 ACM International Conference on Management of Data / ACM. — 2017. — Pp. 913–928.

58. *Dorogovtsev Sergei N, Mendes José FF.* Evolution of networks: From biological nets to the Internet and WWW. — OUP Oxford, 2003.

59. *Penrose Mathew.* Random geometric graphs. No. 5. — Oxford University Press, 2003.

60. *Newman Mark, Barabasi Albert-Laszlo, Watts Duncan J.* The structure and dynamics of networks. — Princeton University Press, 2006.

61. *Durrett Richard.* Random graph dynamics. — Cambridge university press Cambridge, 2007. — Vol. 200.

62. *Caldarelli Guido.* Scale-Free Networks: Complex Webs in Nature and Technology. — Oxford University Press, 2007. — 01.

63. *Alessandro Vespignani, Guido Caldarelli.* Large scale structure and dynamics of complex networks: from information technology to finance and natural science. — World Scientific, 2007. — Vol. 2.

64. *Bonato Anthony.* A course on the web graph. — American Mathematical Soc., 2008. — Vol. 89.

65. *Lovász László.* Large networks and graph limits. — American Mathematical Soc., 2012. — Vol. 60.

66. *Raigorodsky.* Models of random graphs. — Litres, 2011.

67. *Chakrabarti Deepayan, Faloutsos Christos.* Graph Mining: Laws, Tools, and Case Studies. Synthesis Lectures on Data Mining and Knowledge Discovery. — Morgan & Claypool Publishers, 2012. — URL: http://dx.doi.org/10.2200/S00449ED1V01Y201209DMK006.

68. *Harris Jenine K.* An introduction to exponential random graph modeling. — Sage Publications, 2013. — Vol. 173.

69. *Van Der Hofstad Remco.* Random graphs and complex networks. — 2014.

70. *Chakrabarti Deepayan, Faloutsos Christos.* Graph mining: Laws, generators, and algorithms // *ACM computing surveys (CSUR).* — 2006. — Vol. 38, no. 1. — P. 2.

71. A survey of statistical network models / Anna Goldenberg, Alice X Zheng, Stephen E Fienberg et al. // *Foundations and Trends® in Machine Learning.* — 2009. — Vol. 2, no. 2. — Pp. 129–233.

72. Which models are used in social simulation to generate social networks? A review of 17 years of publications in JASSS / Frédéric Amblard, Audren Bouadjio-Boulic, Carlos Sureda Gutiérrez, Benoit Gaudou // Winter Simulation Conference (WSC), 2015 / IEEE. — 2015. — Pp. 4021–4032.

73. *Meyer Ulrich, Penschuck Manuel et al.* Large-scale Graph Generation and Big Data: An Overview on Recent Results // *Bulletin of EATCS.* — 2017. — Vol. 2, no. 122.

74. *Bernovskiy MM, Kuzyurin NN.* Random graphs, models and generators of scale-free graphs // *Proceedings of the Institute for System Programming of the RAS.* — 2012. — Vol. 22.

75. *Kunegis Jérôme.* Konect: the koblenz network collection // Proceedings of the 22nd International Conference on World Wide Web / ACM. — 2013. — Pp. 1343–1350.

76. *Albert Réka, Barabási Albert-László.* Statistical mechanics of complex networks // *Reviews of modern physics.* — 2002. — Vol. 74, no. 1. — P. 47.

77. Directed scale-free graphs / Béla Bollobás, Christian Borgs, Jennifer Chayes, Oliver Riordan // Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms / Society for Industrial and Applied Mathematics. — 2003. — Pp. 132–139.

78. *Aiello William, Chung Fan, Lu Linyuan.* A random graph model for massive graphs // Proceedings of the thirty-second annual ACM symposium on Theory of computing / Acm. — 2000. — Pp. 171–180.

79. *Janson Svante, Łuczak Tomasz, Norros Ilkka.* Large cliques in a power-law random graph // *Journal of Applied Probability.* — 2010. — Vol. 47, no. 4. — Pp. 1124–1135.

80. *Vázquez Alexei.* Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations // *Physical Review E.* — 2003. — Vol. 67, no. 5. — P. 056104.

81. *Wang Yue, Wu Xintao.* Preserving differential privacy in degree-correlation based graph generation // *Transactions on data privacy.* — 2013. — Vol. 6, no. 2. — P. 127.

82. *Bollobás Béla.* Random Graphs. No. 73. — Cambridge University Press, 2001.

83. *Granovetter Mark S.* The strength of weak ties // Social networks. — Elsevier, 1977. — Pp. 347–367.

84. *Bollobás Béla, Riordan Oliver M.* Mathematical results on scale-free random graphs // *Handbook of graphs and networks: from the genome to the internet.* — 2003. — Pp. 1–34.

85. *Kunegis Jérôme, Blattner Marcel, Moser Christine.* Preferential attachment in online networks: Measurement and explanations // Proceedings of the 5th annual ACM web science conference / ACM. — 2013. — Pp. 205–214.

86. A model for social networks / Riitta Toivonen, Jukka-Pekka Onnela, Jari Saramäki et al. // *Physica A: Statistical Mechanics and its Applications*. — 2006. — Vol. 371, no. 2. — Pp. 851–860.

87. The web as a graph: measurements, models, and methods / Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan et al. // International Computing and Combinatorics Conference / Springer. — 1999. — Pp. 1–17.

88. Stochastic models for the web graph / Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan et al. // Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on / IEEE. — 2000. — Pp. 57–65.

89. *Krapivsky Pavel L, Redner Sidney*. Network growth by copying // *Physical Review E*. — 2005. — Vol. 71, no. 3. — P. 036118.

90. *Davidsen Jörn, Ebel Holger, Bornholdt Stefan*. Emergence of a small world from local interactions: Modeling acquaintance networks // *Physical Review Letters*. — 2002. — Vol. 88, no. 12. — P. 128701.

91. *Marsili Matteo, Vega-Redondo Fernando, Slanina František*. The rise and fall of a networked society: A formal model // *Proceedings of the National Academy of Sciences*. — 2004. — Vol. 101, no. 6. — Pp. 1439–1442.

92. *Ravasz Erzsébet, Barabási Albert-László*. Hierarchical organization in complex networks // *Physical Review E*. — 2003. — Vol. 67, no. 2. — P. 026112.

93. *Barabási Albert-László, Ravasz Erzsebet, Vicsek Tamas*. Deterministic scale-free networks // *Physica A: Statistical Mechanics and its Applications*. — 2001. — Vol. 299, no. 3-4. — Pp. 559–564.

94. *Dorogovtsev Sergey N, Goltsev Alexander V, Mendes José Ferreira F*. Pseudofractal scale-free web // *Physical review E*. — 2002. — Vol. 65, no. 6. — P. 066122.

95. *Molontay Roland*. Fractal Characterization of Complex Networks: Ph.D. thesis / Department of Stochastics, Budapest University of Technology and Economics. — 2015.

96. *Rozenfeld Hernán D, Havlin Shlomo, Ben-Avraham Daniel.* Fractal and transfractal recursive scale-free nets // *New Journal of Physics.* — 2007. — Vol. 9, no. 6. — P. 175.

97. Fractality and scale-free effect of a class of self-similar networks / Lifeng Xi, Lihong Wang, Songjing Wang et al. // *Physica A: Statistical Mechanics and its Applications.* — 2017. — Vol. 478. — Pp. 31–40.

98. *Seshadhri Comandur, Pinar Ali, Kolda Tamara G.* An in-depth study of stochastic Kronecker graphs // Data Mining (ICDM), 2011 IEEE 11th International Conference on / IEEE. — 2011. — Pp. 587–596.

99. Modeling the Variance of Network Populations with Mixed Kronecker Product Graph Models / Sebastian Moreno, Jennifer Neville, Sergey Kirshner, SVN Vishwanathan.

100. *Moreno S, Robles P, Neville J.* Block kronecker product graph model // Workshop on Mining and Learning from Graphs. — 2013.

101. *McPherson Miller, Smith-Lovin Lynn, Cook James M.* Birds of a feather: Homophily in social networks // *Annual review of sociology.* — 2001. — Vol. 27, no. 1. — Pp. 415–444.

102. *Onat Furuzan Atay, Stojmenovic Ivan, Yanikomeroglu Halim.* Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks // *Pervasive and Mobile Computing.* — 2008. — Vol. 4, no. 5. — Pp. 597–615.

103. *Waxman Bernard M.* Routing of multipoint connections // *IEEE journal on selected areas in communications.* — 1988. — Vol. 6, no. 9. — Pp. 1617–1622.

104. *Yook Soon-Hyung, Jeong Hawoong, Barabási Albert-László.* Modeling the Internet's large-scale topology // *Proceedings of the National Academy of Sciences.* — 2002. — Vol. 99, no. 21. — Pp. 13382–13386.

105. *Wong Ling Heng, Pattison Philippa, Robins Garry.* A spatial model for social networks // *Physica A: Statistical Mechanics and its Applications.* — 2006. — Vol. 360, no. 1. — Pp. 99–120.

106. *Handcock Mark S, Raftery Adrian E, Tantrum Jeremy M.* Model-based clustering for social networks // *Journal of the Royal Statistical Society: Series A (Statistics in Society).* — 2007. — Vol. 170, no. 2. — Pp. 301–354.

107. *Medina Alberto, Matta Ibrahim, Byers John.* On the origin of power laws in Internet topologies // *ACM SIGCOMM computer communication review.* — 2000. — Vol. 30, no. 2. — Pp. 18–28.

108. *Gugelmann Luca, Panagiotou Konstantinos, Peter Ueli.* Random hyperbolic graphs: degree sequence and clustering // International Colloquium on Automata, Languages, and Programming / Springer. — 2012. — Pp. 573–585.

109. *Kiwi Marcos, Mitsche Dieter.* A bound for the diameter of random hyperbolic graphs // Proceedings of the Meeting on Analytic Algorithmics and Combinatorics / Society for Industrial and Applied Mathematics. — 2015. — Pp. 26–39.

110. *Friedrich Tobias, Krohmer Anton.* On the diameter of hyperbolic random graphs // *SIAM Journal on Discrete Mathematics.* — 2018. — Vol. 32, no. 2. — Pp. 1314–1334.

111. *Bringmann Karl, Keusch Ralph, Lengler Johannes.* Geometric inhomogeneous random graphs // *arXiv preprint arXiv:1511.00576.* — 2016.

112. *Keusch Ralph.* Geometric Inhomogeneous Random Graphs and Graph Coloring Games: Ph.D. thesis / ETH Zurich. — 2018.

113. *Fabrikant Alex, Koutsoupias Elias, Papadimitriou Christos H.* Heuristically optimized trade-offs: A new paradigm for power laws in the Internet // International Colloquium on Automata, Languages, and Programming / Springer. — 2002. — Pp. 110–122.

114. Competition-induced preferential attachment / Noam Berger, Christian Borgs, Jennifer T Chayes et al. // International Colloquium on Automata, Languages, and Programming / Springer. — 2004. — Pp. 208–221.

115. Spontaneous emergence of complex optimal networks through evolutionary adaptation / Venkat Venkatasubramanian, Santhoji Katare, Priyan R Patkar, Fang-ping Mu // *Computers & chemical engineering.* — 2004. — Vol. 28, no. 9. — Pp. 1789–1798.

116. *Bender Edward A, Canfield E Rodney.* The asymptotic number of labeled graphs with given degree sequences // *Journal of Combinatorial Theory, Series A.* — 1978. — Vol. 24, no. 3. — Pp. 296–307.

117. *Chung Fan, Lu Linyuan.* The average distances in random graphs with given expected degrees // *Proceedings of the National Academy of Sciences.* — 2002. — Vol. 99, no. 25. — Pp. 15879–15882.

118. *Chung Fan, Lu Linyuan.* Connected components in random graphs with given expected degree sequences // *Annals of combinatorics.* — 2002. — Vol. 6, no. 2. — Pp. 125–145.

119. *Kovalenko IN.* Theory of random graphs // *Cybernetics.* — 1971. — Vol. 7, no. 4. — Pp. 575–579.

120. Configuring random graph models with fixed degree sequences / Bailey K Fosdick, Daniel B Larremore, Joel Nishimura, Johan Ugander // *SIAM Review.* — 2018. — Vol. 60, no. 2. — Pp. 315–355.

121. *Heath Lenwood S, Parikh Nidhi.* Generating random graphs with tunable clustering coefficients // *Physica A: Statistical Mechanics and its Applications.* — 2011. — Vol. 390, no. 23-24. — Pp. 4577–4587.

122. *Lancichinetti Andrea, Fortunato Santo, Radicchi Filippo.* Benchmark graphs for testing community detection algorithms // *Physical review E.* — 2008. — Vol. 78, no. 4. — P. 046110.

123. *Benson Austin R, Riquelme Carlos, Schmit Sven.* Learning multifractal structure in large networks // Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining / ACM. — 2014. — Pp. 1326–1335.

124. *Gleich David F, Owen Art B.* Moment-based estimation of stochastic Kronecker graph parameters // *Internet Mathematics.* — 2012. — Vol. 8, no. 3. — Pp. 232–256.

125. *Van Duijn Marijtje AJ, Gile Krista J, Handcock Mark S.* A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models // *Social Networks.* — 2009. — Vol. 31, no. 1. — Pp. 52–62.

126. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications / Aurelien Decelle, Florent Krzakala, Cristopher Moore, Lenka Zdeborová // *Physical Review E*. — 2011. — Vol. 84, no. 6. — P. 066106.

127. *An Weihua*. Fitting ERGMs on big networks // *Social science research*. — 2016. — Vol. 59. — Pp. 107–119.

128. *Taylor Richard*. Constrained switchings in graphs // Combinatorial Mathematics VIII. — Springer, 1981. — Pp. 314–336.

129. On the uniform generation of random graphs with prescribed degree sequences / Ron Milo, Nadav Kashtan, Shalev Itzkovitz et al. // *arXiv preprint cond-mat/0312028*. — 2003.

130. *Bansal Shweta, Khandelwal Shashank, Meyers Lauren Ancel*. Exploring biological network structure with clustered random networks // *BMC bioinformatics*. — 2009. — Vol. 10, no. 1. — P. 405.

131. *Tabourier Lionel, Roth Camille, Cointet Jean-Philippe*. Generating constrained random graphs using multiple edge switches // *Journal of Experimental Algorithmics (JEA)*. — 2011. — Vol. 16. — Pp. 1–7.

132. *Gutfraind Alexander, Safro Ilya, Meyers Lauren Ancel*. Multiscale network generation // Information Fusion (Fusion), 2015 18th International Conference on / IEEE. — 2015. — Pp. 158–165.

133. Characterization and modeling of weighted networks / Marc Barthélemy, Alain Barrat, Romualdo Pastor-Satorras, Alessandro Vespignani // *Physica a: Statistical mechanics and its applications*. — 2005. — Vol. 346, no. 1-2. — Pp. 34–43.

134. *Girvan Michelle, Newman Mark EJ*. Community structure in social and biological networks // *Proceedings of the national academy of sciences*. — 2002. — Vol. 99, no. 12. — Pp. 7821–7826.

135. *Sawardecker Erin N, Sales-Pardo Marta, Amaral Luıs A Nunes*. Detection of node group membership in networks with group overlap // *The European Physical Journal B*. — 2009. — Vol. 67, no. 3. — Pp. 277–284.

136. *Arenas Alex, Díaz-Guilera Albert, Pérez-Vicente Conrad J.* Synchronization reveals topological scales in complex networks // *Physical review letters.* — 2006. — Vol. 96, no. 11. — P. 114102.

137. *Seshadhri Comandur, Kolda Tamara G, Pinar Ali.* Community structure and scale-free collections of Erdős-Rényi graphs // *Physical Review E.* — 2012. — Vol. 85, no. 5. — P. 056109.

138. *Newman Mark EJ.* Analysis of weighted networks // *Physical review E.* — 2004. — Vol. 70, no. 5. — P. 056131.

139. *Simpson Sean L, Hayasaka Satoru, Laurienti Paul J.* Exponential random graph modeling for complex brain networks // *PloS one.* — 2011. — Vol. 6, no. 5. — P. e20039.

140. An introduction to exponential random graph (p*) models for social networks / Garry Robins, Pip Pattison, Yuval Kalish, Dean Lusher // *Social networks.* — 2007. — Vol. 29, no. 2. — Pp. 173–191.

141. Network robustness and fragility: Percolation on random graphs / Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, Duncan J Watts // *Physical review letters.* — 2000. — Vol. 85, no. 25. — P. 5468.

142. *Castellano Claudio, Fortunato Santo, Loreto Vittorio.* Statistical physics of social dynamics // *Reviews of modern physics.* — 2009. — Vol. 81, no. 2. — P. 591.

143. *Thiriot Samuel, Kant Jean-Daniel.* Generate country-scale networks of interaction from scattered statistics // The fifth conference of the European social simulation association, Brescia, Italy. — Vol. 240. — 2008.

144. Roll: Fast in-memory generation of gigantic scale-free networks / Ali Hadian, Sadegh Nobari, Behrooz Minaei-Bidgoli, Qiang Qu // Proceedings of the 2016 International Conference on Management of Data / ACM. — 2016. — Pp. 1829–1842.

145. Darwini: Generating realistic large-scale social graphs / Sergey Edunov, Dionysios Logothetis, Cheng Wang et al. // *arXiv preprint arXiv:1610.00664.* — 2016.

146. *Zweig Katharina A.* Network analysis literacy: a practical approach to the analysis of networks. Lecture notes in social networks. — 2016.

147. Superfamilies of evolved and designed networks / Ron Milo, Shalev Itzkovitz, Nadav Kashtan et al. // *Science*. — 2004. — Vol. 303, no. 5663. — Pp. 1538–1542.

148. *Newman Mark EJ, Girvan Michelle.* Finding and evaluating community structure in networks // *Physical review E*. — 2004. — Vol. 69, no. 2. — P. 026113.

149. *Karrer Brian, Newman Mark EJ.* Stochastic blockmodels and community structure in networks // *Physical review E*. — 2011. — Vol. 83, no. 1. — P. 016107.

150. *Narayanan Arvind, Shmatikov Vitaly.* De-anonymizing social networks // Security and Privacy, 2009 30th IEEE Symposium on / IEEE. — 2009. — Pp. 173–187.

151. *Ying Xiaowei, Wu Xintao.* Randomizing social networks: a spectrum preserving approach // proceedings of the 2008 SIAM International Conference on Data Mining / SIAM. — 2008. — Pp. 739–750.

152. *Dwork Cynthia.* Differential privacy: A survey of results // International Conference on Theory and Applications of Models of Computation / Springer. — 2008. — Pp. 1–19.

153. SecGraph: A Uniform and Open-source Evaluation System for Graph Data Anonymization and De-anonymization. / Shouling Ji, Weiqing Li, Prateek Mittal et al. // USENIX Security Symposium. — 2015. — Pp. 303–318.

154. *Song Chaoming, Havlin Shlomo, Makse Hernán A.* Self-similarity of complex networks // *Nature*. — 2005. — January. — Vol. 433. — Pp. 392–395.

155. *Nickel Christine Leigh Myers.* Random dot product graphs: A model for social networks. — 2007. — Vol. 68.

156. *Young Stephen J, Scheinerman Edward R.* Random dot product graph models for social networks // International Workshop on Algorithms and Models for the Web-Graph / Springer. — 2007. — Pp. 138–149.

157. *Scheinerman Edward R, Tucker Kimberly.* Modeling graphs using dot product representations // *Computational Statistics.* — 2010. — Vol. 25, no. 1. — Pp. 1–16.

158. *Mahapatra Suchismit, Chandola Varun.* Modeling graphs using a mixture of Kronecker models // Big Data (Big Data), 2015 IEEE International Conference on / IEEE. — 2015. — Pp. 727–736.

159. *Newman Mark EJ.* Power laws, Pareto distributions and Zipf's law // *Contemporary physics.* — 2005. — Vol. 46, no. 5. — Pp. 323–351.

160. Distributed representations of words and phrases and their compositionality / Tomas Mikolov, Ilya Sutskever, Kai Chen et al. // Advances in neural information processing systems. — 2013. — Pp. 3111–3119.

161. *Perozzi Bryan, Al-Rfou Rami, Skiena Steven.* Deepwalk: Online learning of social representations // Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining / ACM. — 2014. — Pp. 701–710.

162. Line: Large-scale information network embedding / Jian Tang, Meng Qu, Mingzhe Wang et al. // Proceedings of the 24th International Conference on World Wide Web / ACM. — 2015. — Pp. 1067–1077.

163. *Shavitt Yuval, Tankel Tomer.* Big-bang simulation for embedding network distances in euclidean space // *IEEE/ACM Transactions on Networking (TON).* — 2004. — Vol. 12, no. 6. — Pp. 993–1006.

164. *Kobourov Stephen G.* Spring embedders and force directed graph drawing algorithms // *arXiv preprint arXiv:1201.3011.* — 2012.

165. *Luo Bin, Wilson Richard C, Hancock Edwin R.* Spectral embedding of graphs // *Pattern recognition.* — 2003. — Vol. 36, no. 10. — Pp. 2213–2230.

166. *Morin Frederic, Bengio Yoshua.* Hierarchical Probabilistic Neural Network Language Model. // Aistats / Citeseer. — Vol. 5. — 2005. — Pp. 246–252.

167. *Ivanov Oleg U, Bartunov Sergey O.* Learning representations in directed networks // International Conference on Analysis of Images, Social Networks and Texts / Springer. — 2015. — Pp. 196–207.

168. *Gutmann Michael, Hyvärinen Aapo.* Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. // AISTATS. — Vol. 1. — 2010. — P. 6.

169. *Mnih Andriy, Teh Yee Whye.* A fast and simple algorithm for training neural probabilistic language models // *arXiv preprint arXiv:1206.6426.* — 2012.

170. *Grover Aditya, Leskovec Jure.* node2vec: Scalable Feature Learning for Networks.

171. Generative adversarial nets / Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza et al. // Advances in neural information processing systems. — 2014. — Pp. 2672–2680.

172. *Kunegis Jérôme.* KONECT – The Koblenz Network Collection // Proc. Int. Conf. on World Wide Web Companion. — 2013. — Pp. 1343–1350. — URL: http://userpages.uni-koblenz.de/~kunegis/paper/kunegis-koblenz-network-collection.pdf.

173. *Leskovec Jure, Krevl Andrej.* SNAP Datasets: Stanford Large Network Dataset Collection. — http://snap.stanford.edu/data. — 2014. — jun.

174. Finding statistically significant communities in networks / Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, Santo Fortunato // *PloS one.* — 2011. — Vol. 6, no. 4. — P. e18961.

175. *Leskovec Jure, Sosič Rok.* SNAP: A General-Purpose Network Analysis and Graph-Mining Library // *ACM Transactions on Intelligent Systems and Technology (TIST).* — 2016. — Vol. 8, no. 1. — P. 1.

176. *Leskovec Jure, Kleinberg Jon, Faloutsos Christos.* Graphs over time: densification laws, shrinking diameters and possible explanations // Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining / ACM. — 2005. — Pp. 177–187.

177. *Zager Laura A, Verghese George C.* Graph similarity scoring and matching // *Applied mathematics letters.* — 2008. — Vol. 21, no. 1. — Pp. 86–94.

178. UpSizeR: Synthetically scaling an empirical relational database / YC Tay, Bing Tian Dai, Daniel T Wang et al. // *Information Systems*. — 2013. — Vol. 38, no. 8. — Pp. 1168–1183.

179. *Seshadhri C, Pinar Ali, Kolda Tamara G.* An in-depth analysis of stochastic Kronecker graphs // *Journal of the ACM (JACM)*. — 2013. — Vol. 60, no. 2. — P. 13.

180. Tied Kronecker product graph models to capture variance in network populations / Sebastian Moreno, Sergey Kirshner, Jennifer Neville, SVN Vishwanathan // Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on / IEEE. — 2010. — Pp. 1137–1144.

Приложение А

Эксперименты в процессе разработки ERGG-dwc

А.1 Метод аппроксимации распределения

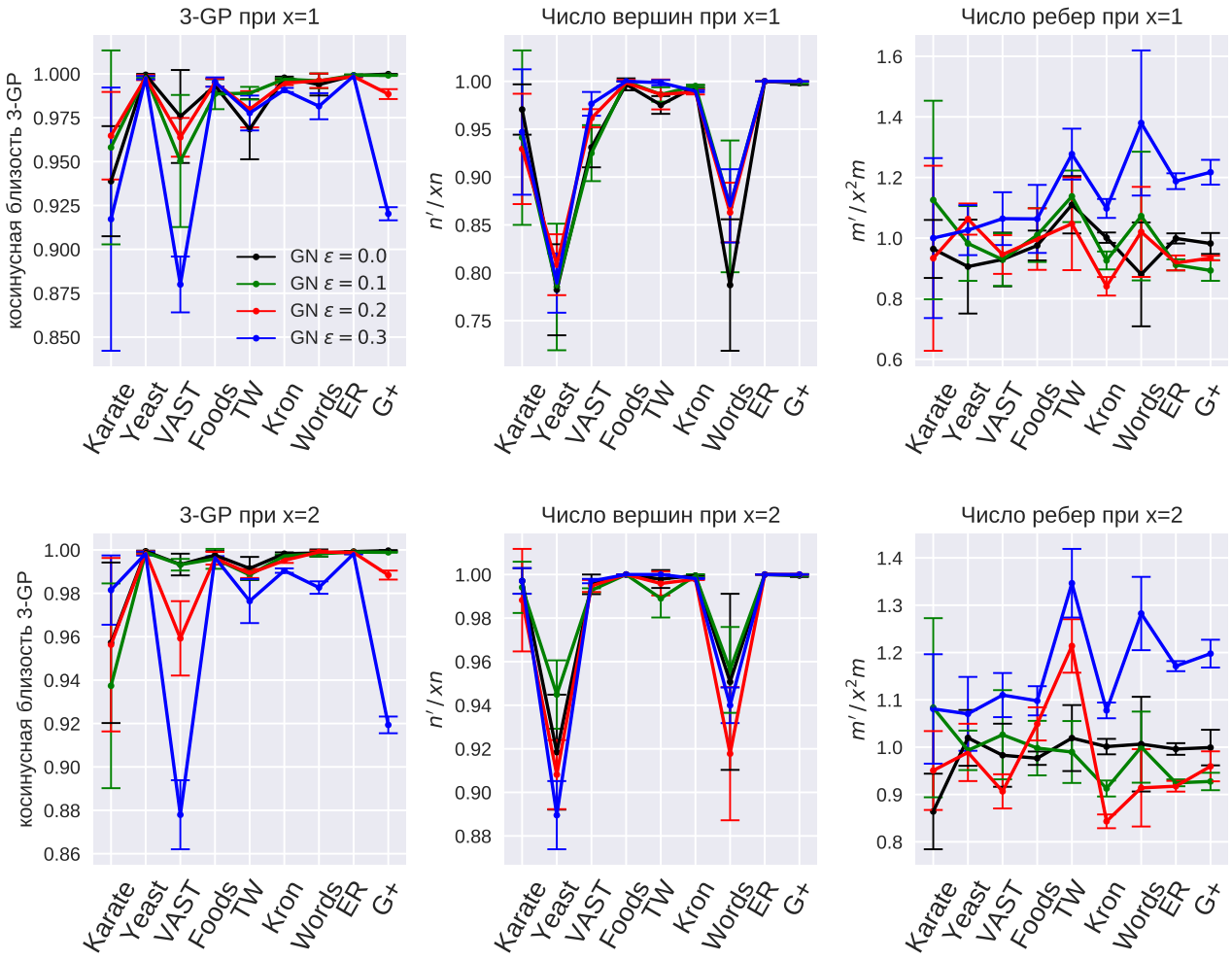

Рисунок А.1 — Варьирование величины шума $\varepsilon \in \{0.0; 0.1; 0.2; 0.3\}$ в методе GN при коэффициенте масштабирования $x = 1; 2$. Оценка косинусной близости 3-GP, отношений количества вершин и ребер к их ожидаемым количествам. Разброс значений оценен по 5 запускам.

А.2 Атрибуты

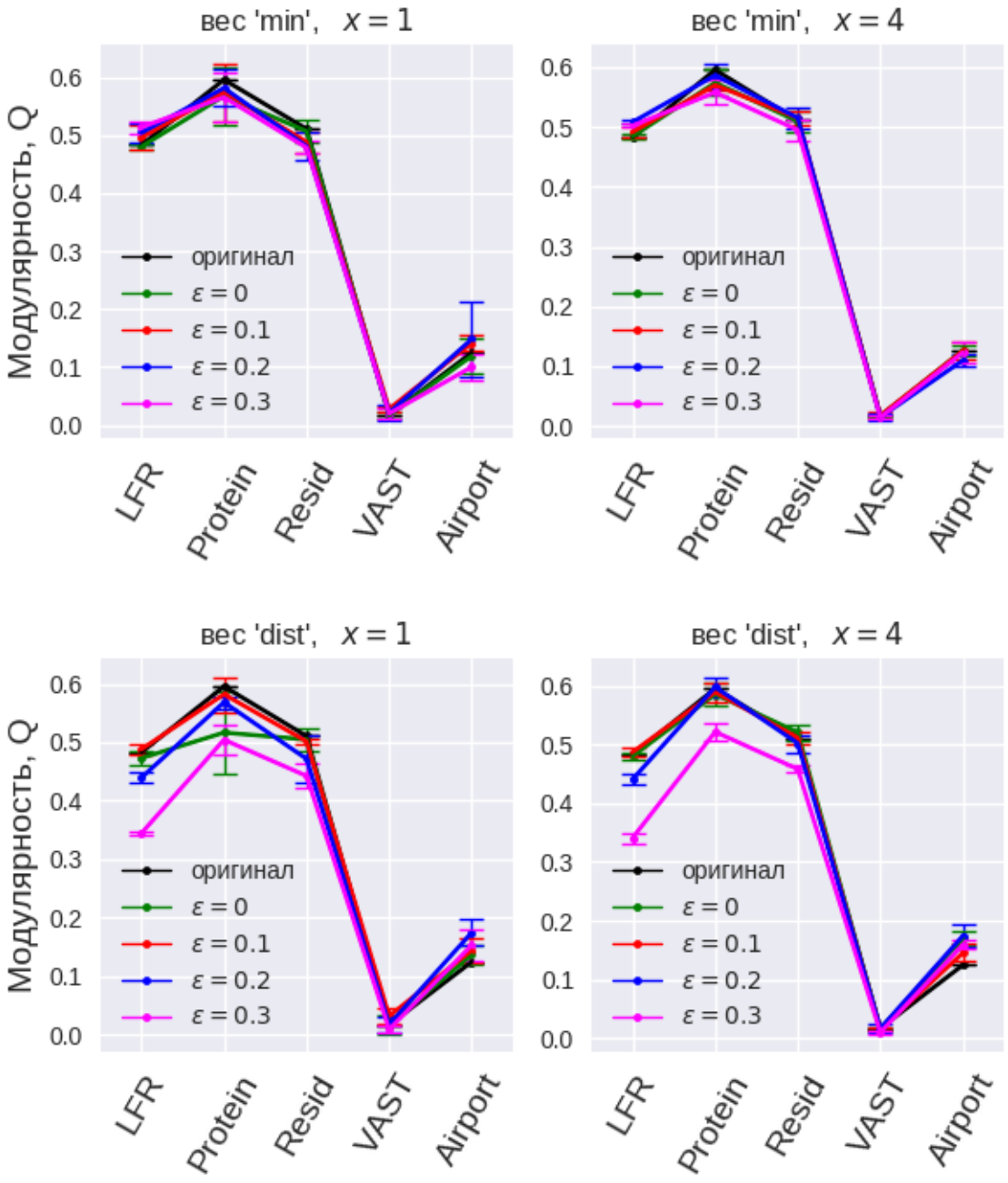

Рисунок А.2 — Зависимость модулярности в генерируемых графах от амплитуды шума ε в GN. Вес по умолчанию ’min’ (сверху) и ’dist’ (снизу). Коэффициент масштабирования $x = 1$ (слева) и $x = 4$ (справа). Разброс значений оценен по 5 запускам.

Приложение Б

Экспериментальное сравнение ERGG-dwc с другими методами

Б.1 Измерение похожести

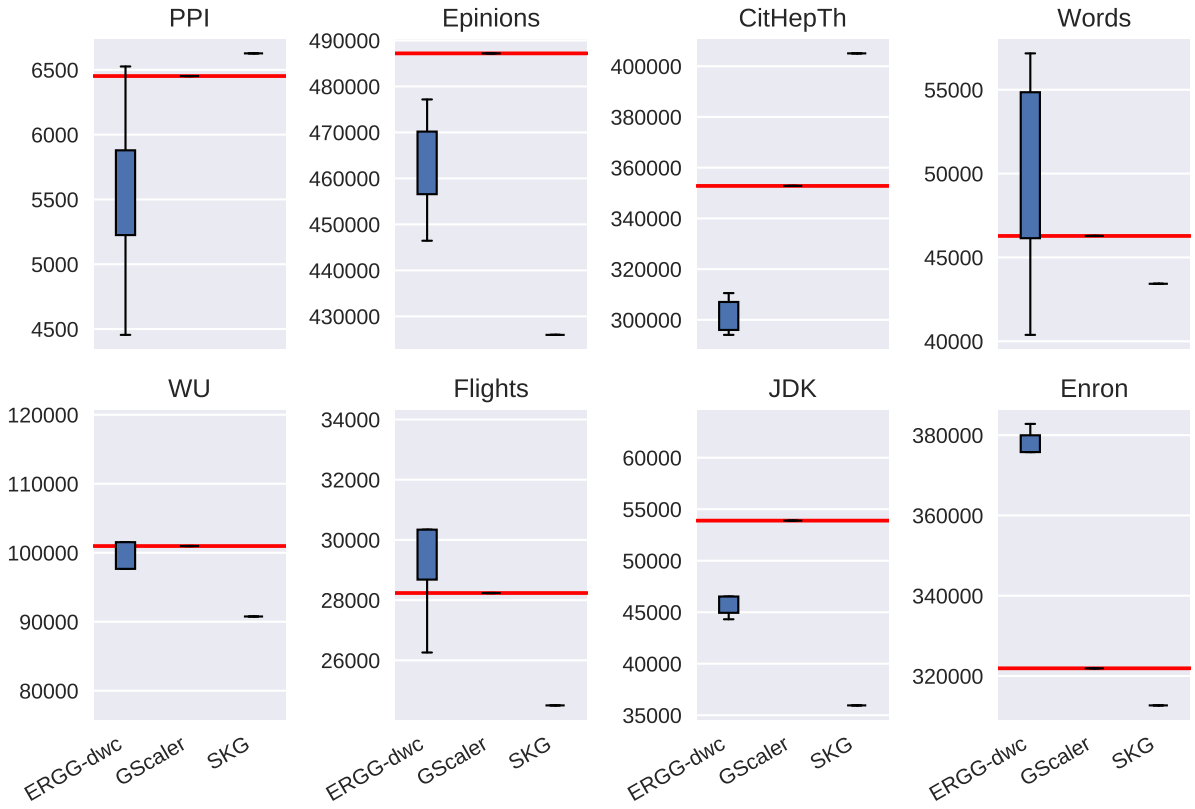

Рисунок Б.1 — Количество ребер m.

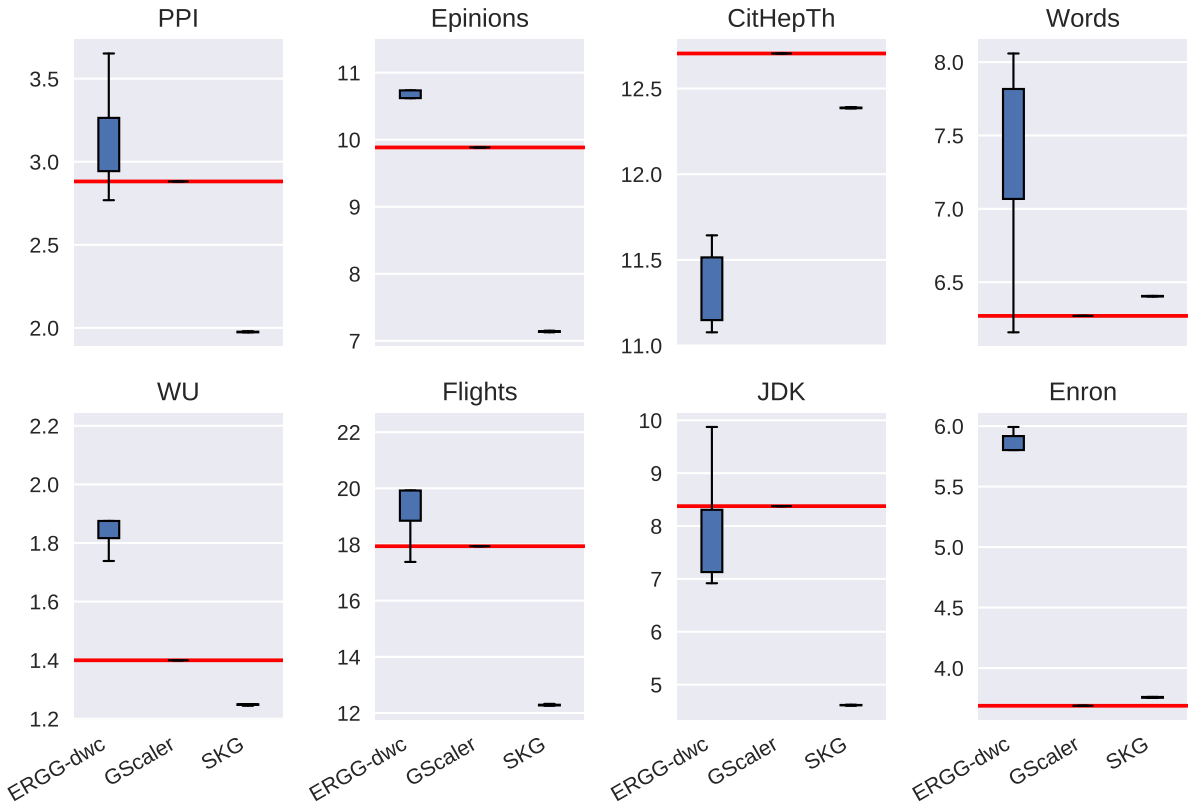

Рисунок Б.2 — Средняя степень.

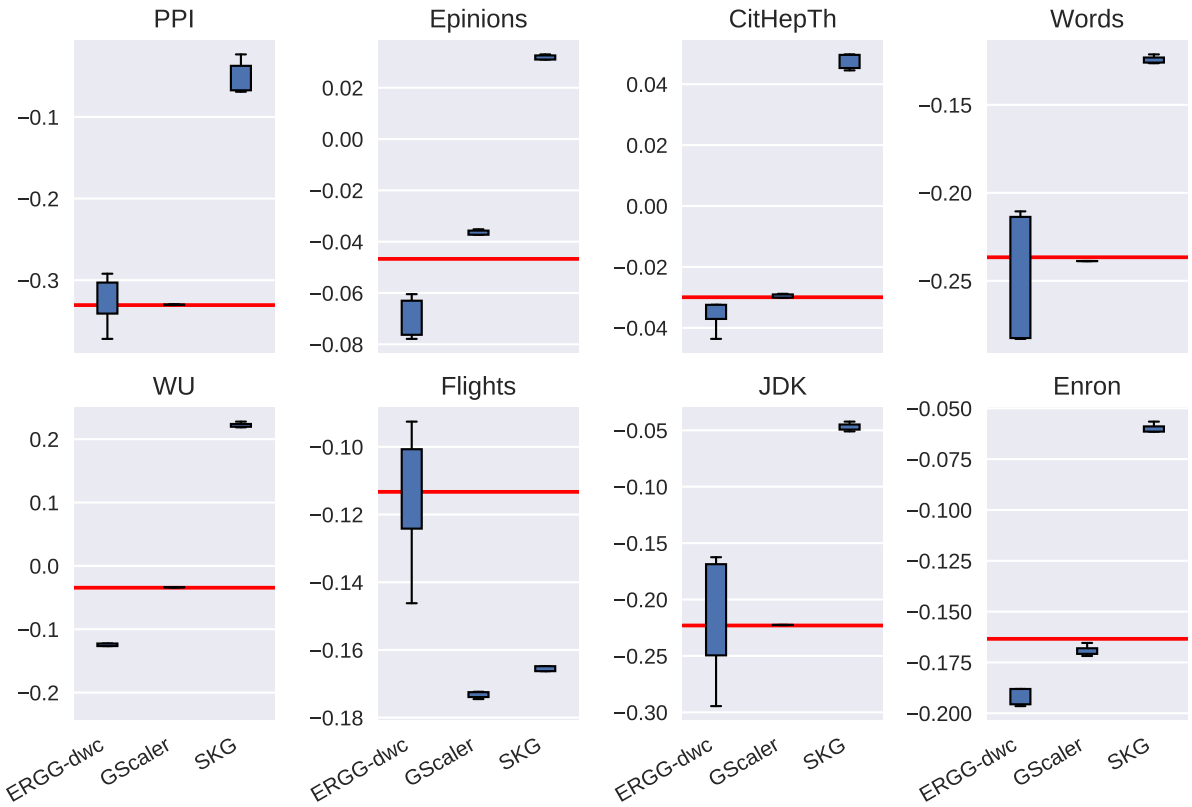

Рисунок Б.3 — Ассортативность степеней.

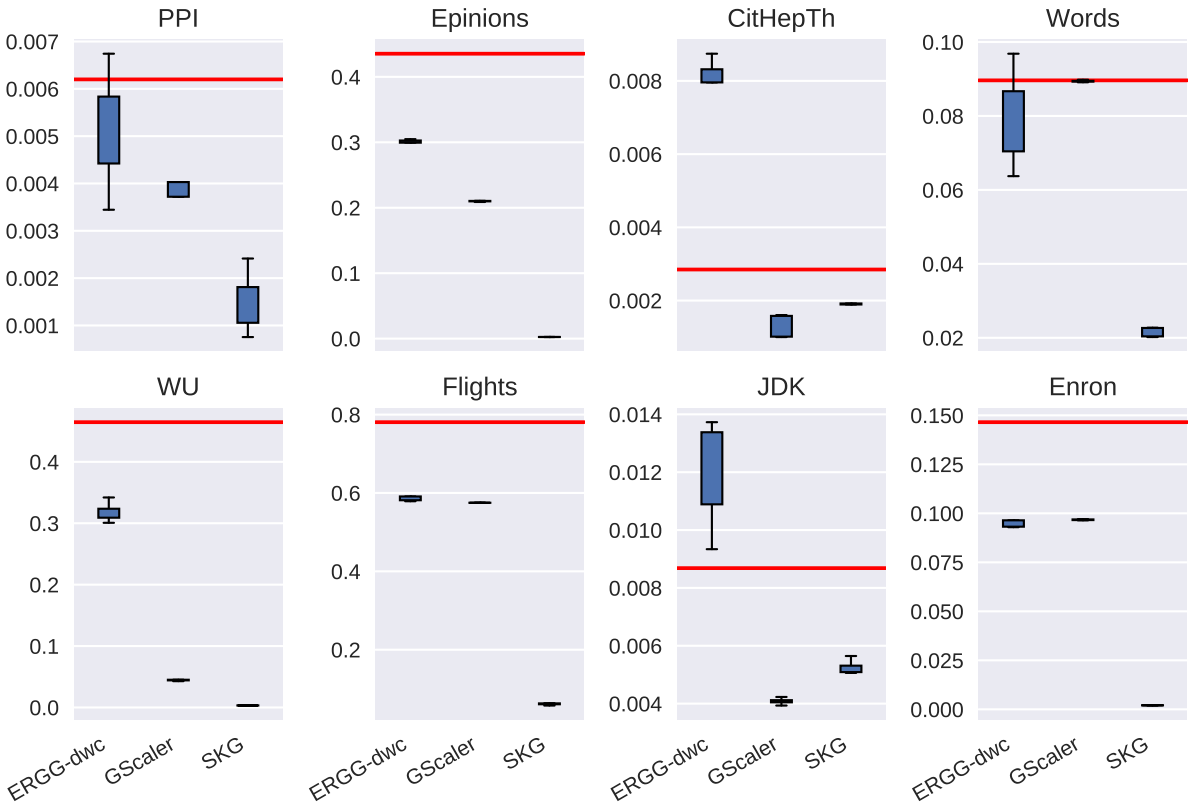

Рисунок Б.4 — Взаимность ребер.

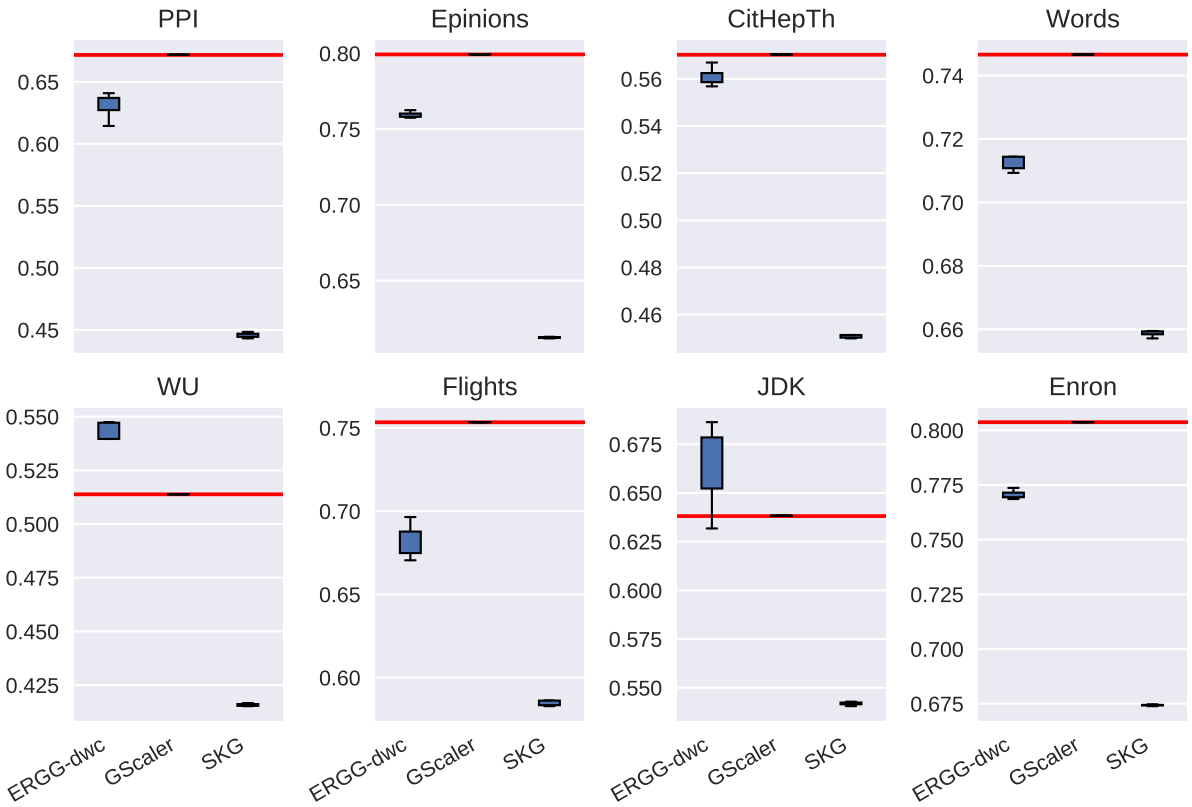

Рисунок Б.5 — Коэффициент Джини распределения степеней.

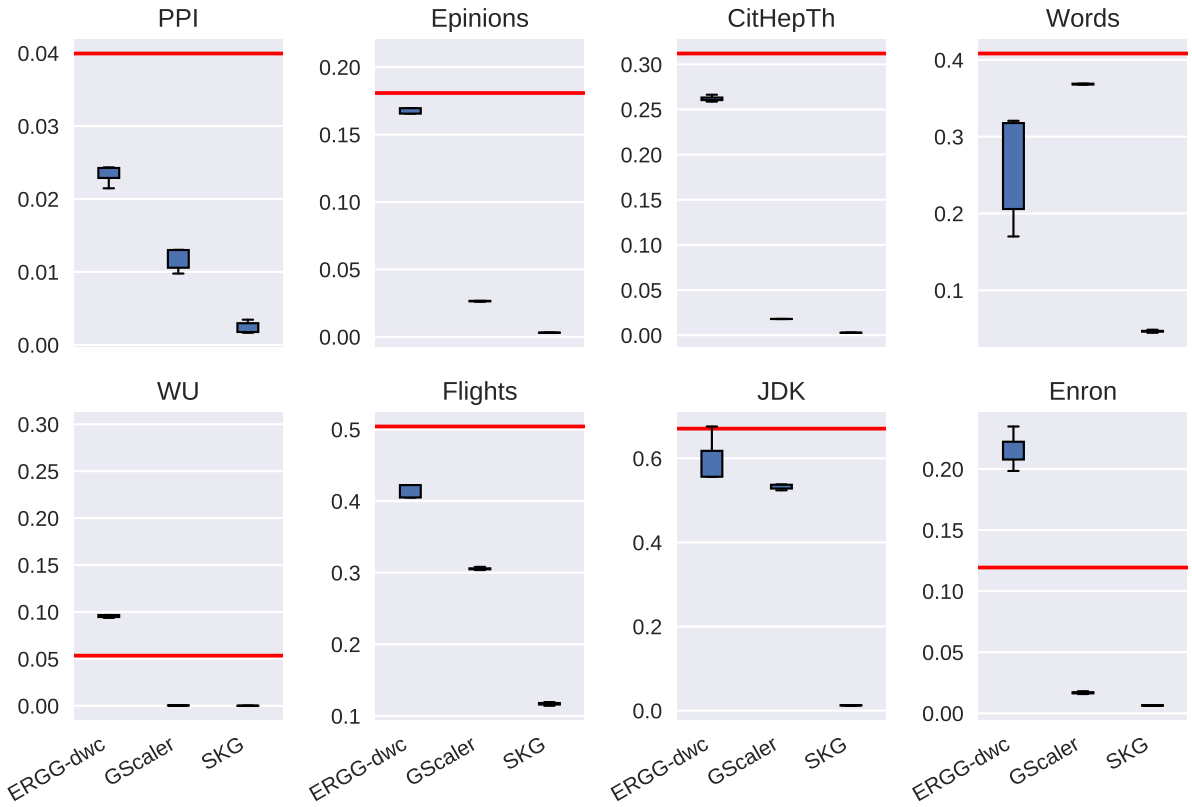

Рисунок Б.6 — Средний коэффициент кластеризации.

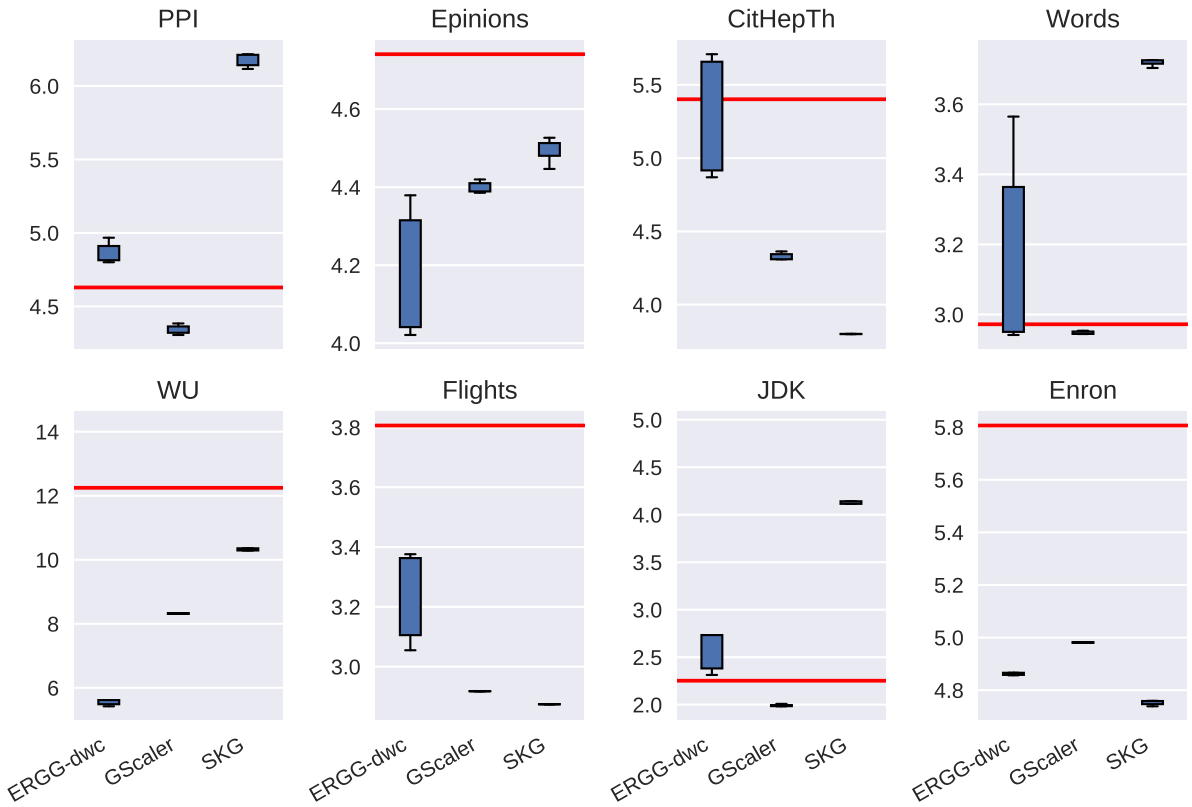

Рисунок Б.7 — 90% эффективный диаметр.

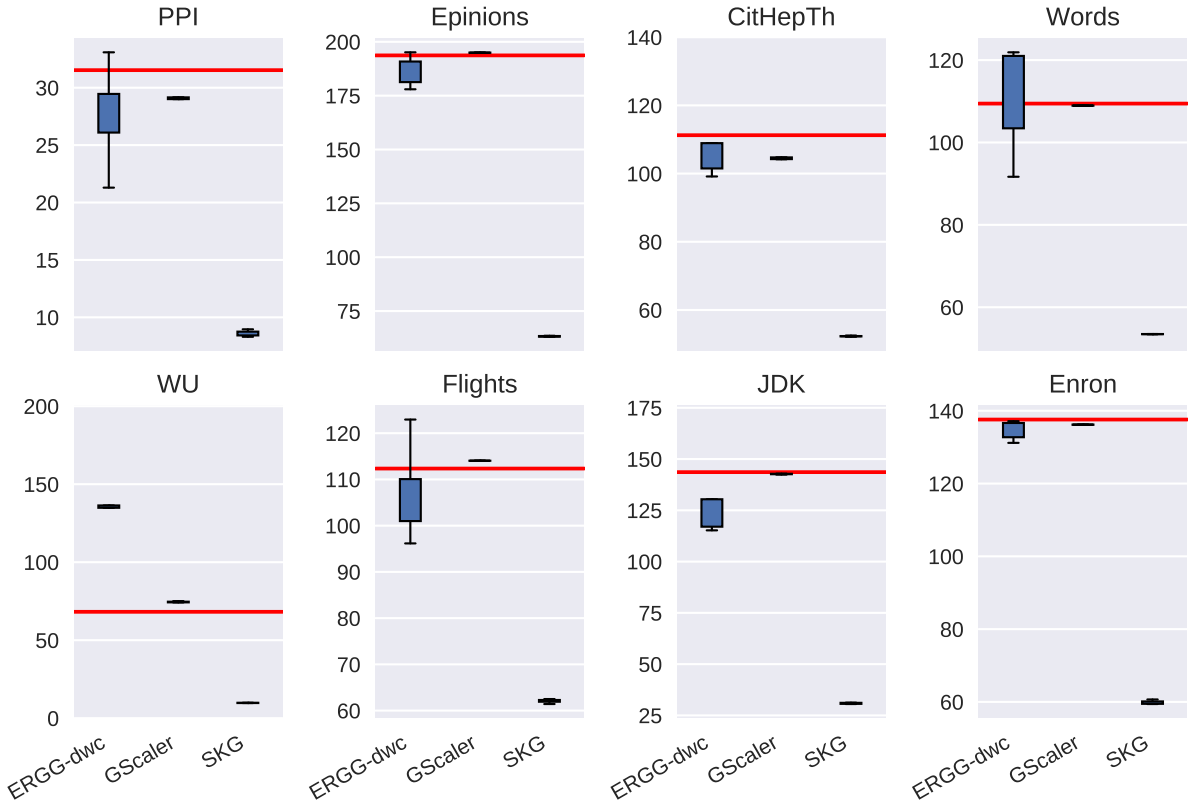

Рисунок Б.8 — Спектральный радиус.

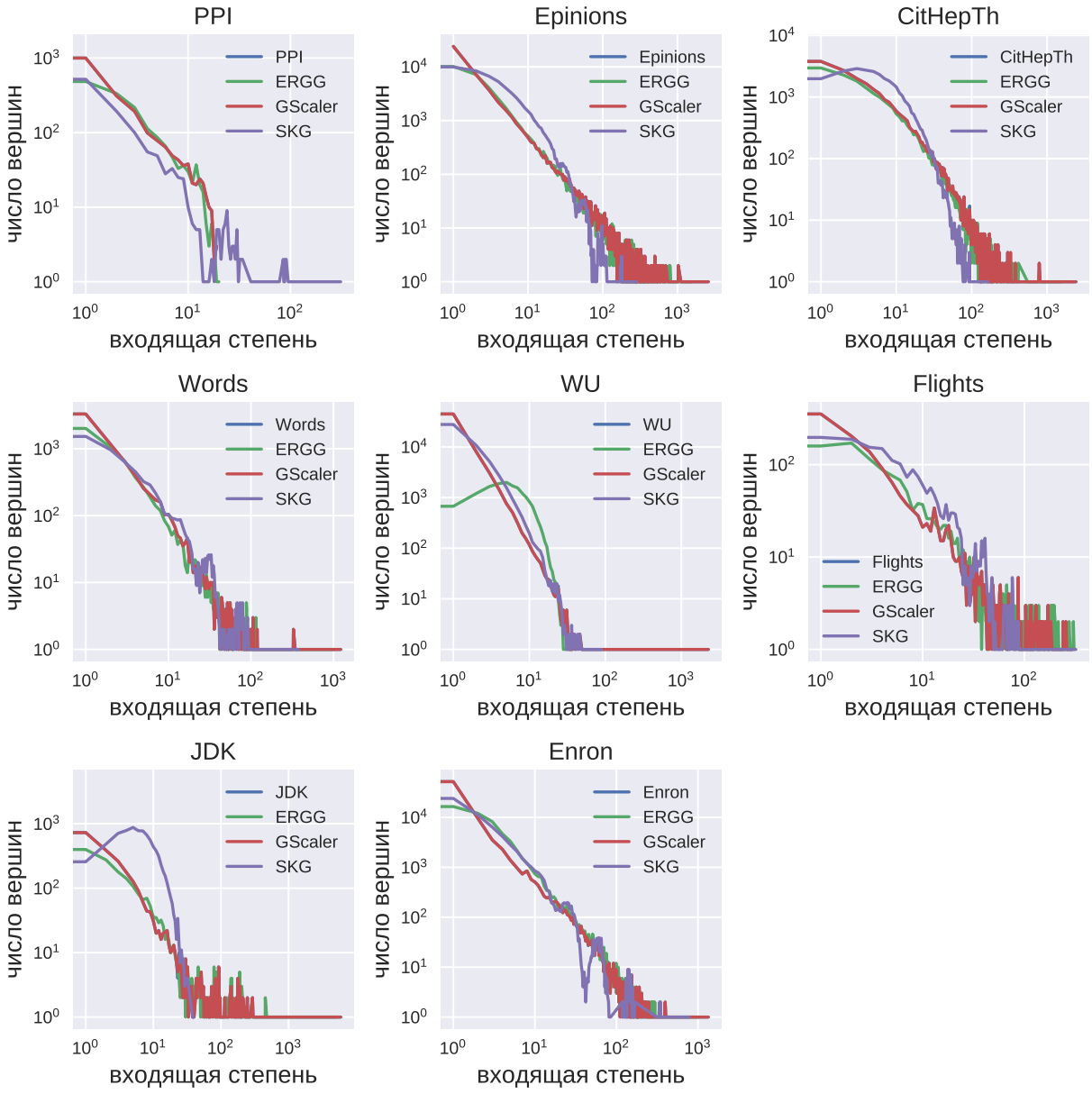

Рисунок Б.9 — Распределение входящих степеней. График оригинального графа перекрывается результатом Gscaler.

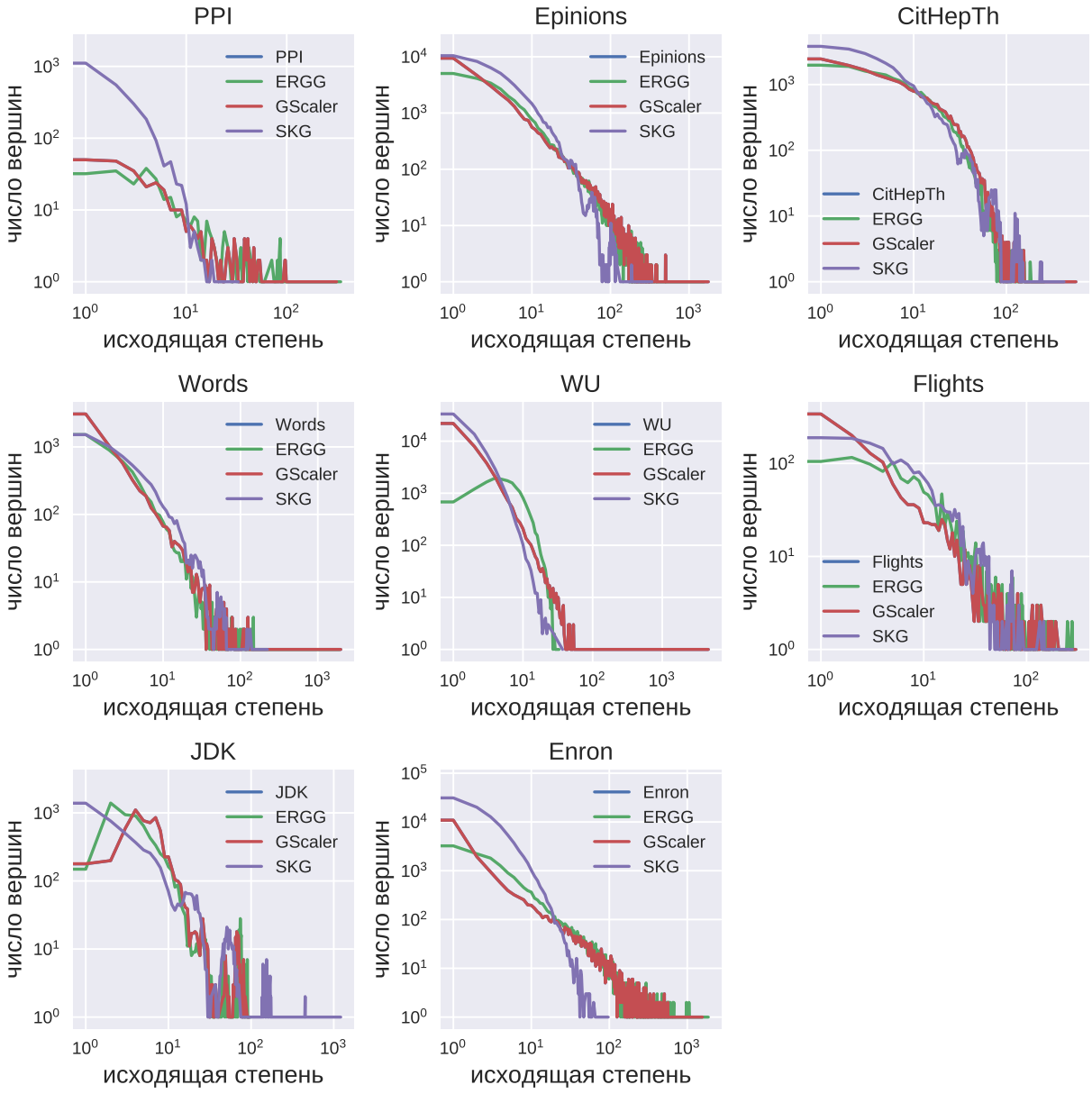

Рисунок Б.10 — Распределение исходящих степеней. График оригинального графа перекрывается результатом Gscaler.

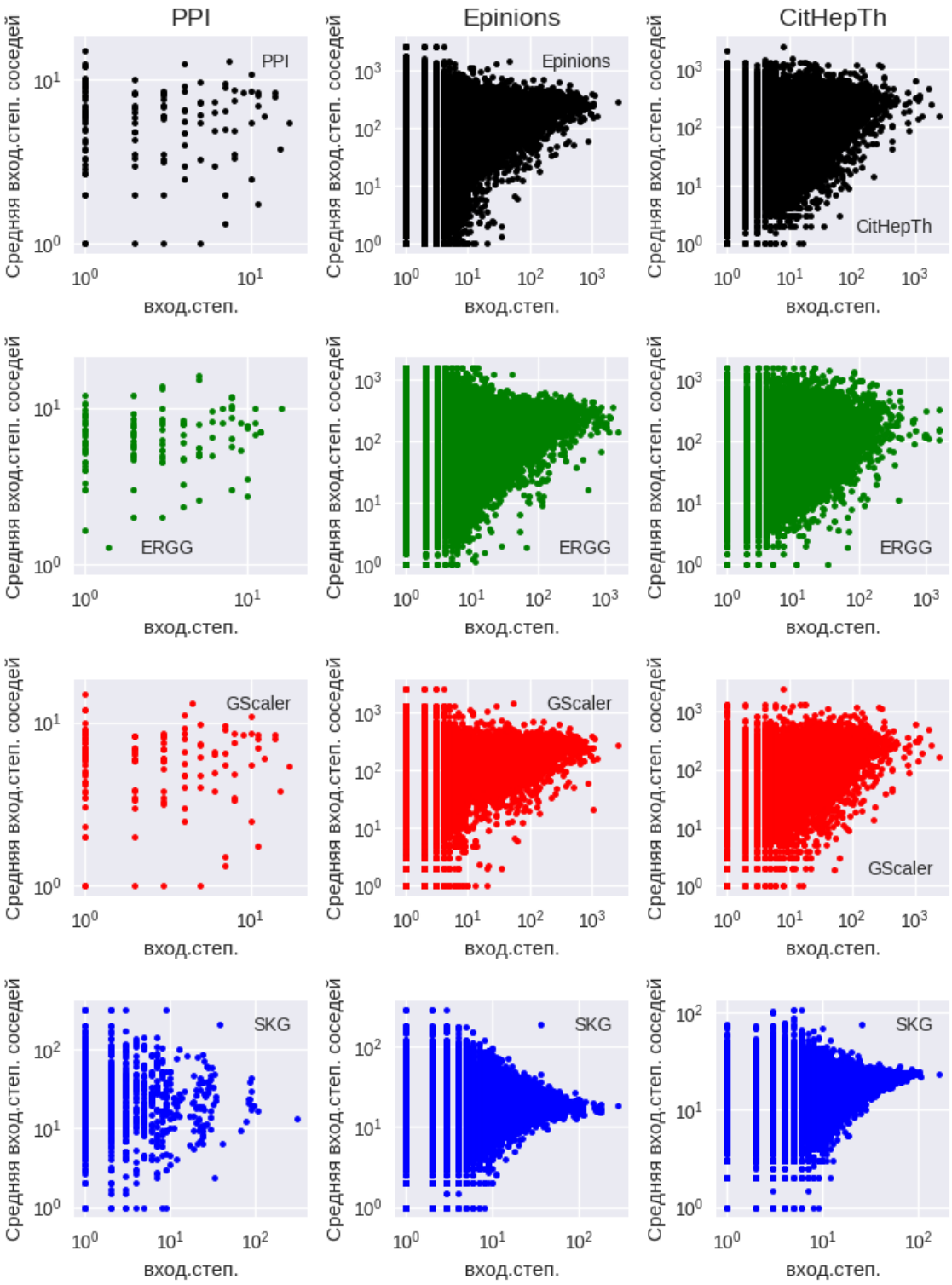

Рисунок Б.11 — Ассортативность входящих степеней (1 из 3).

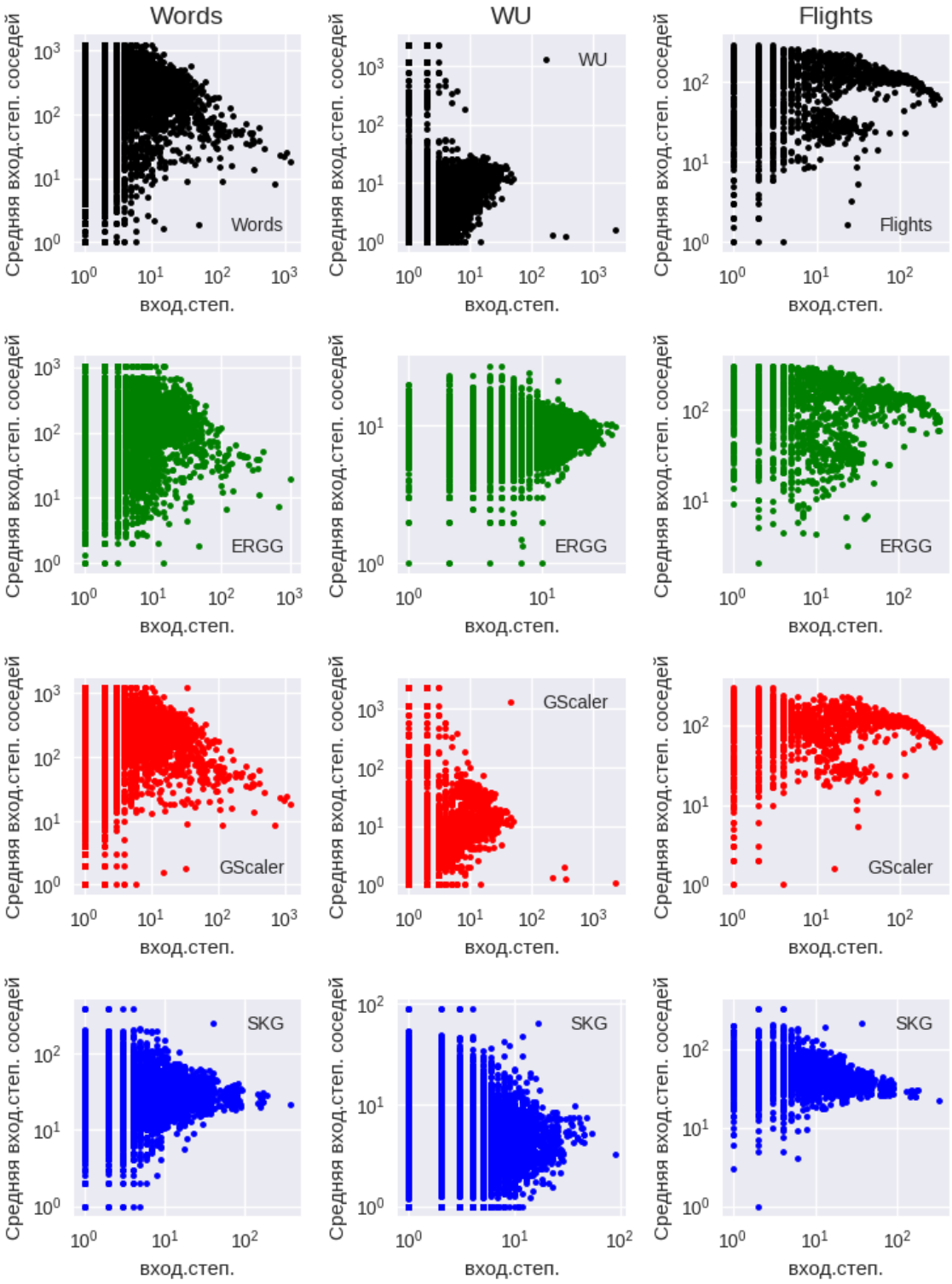

Рисунок Б.12 — Ассортативность входящих степеней (2 из 3).

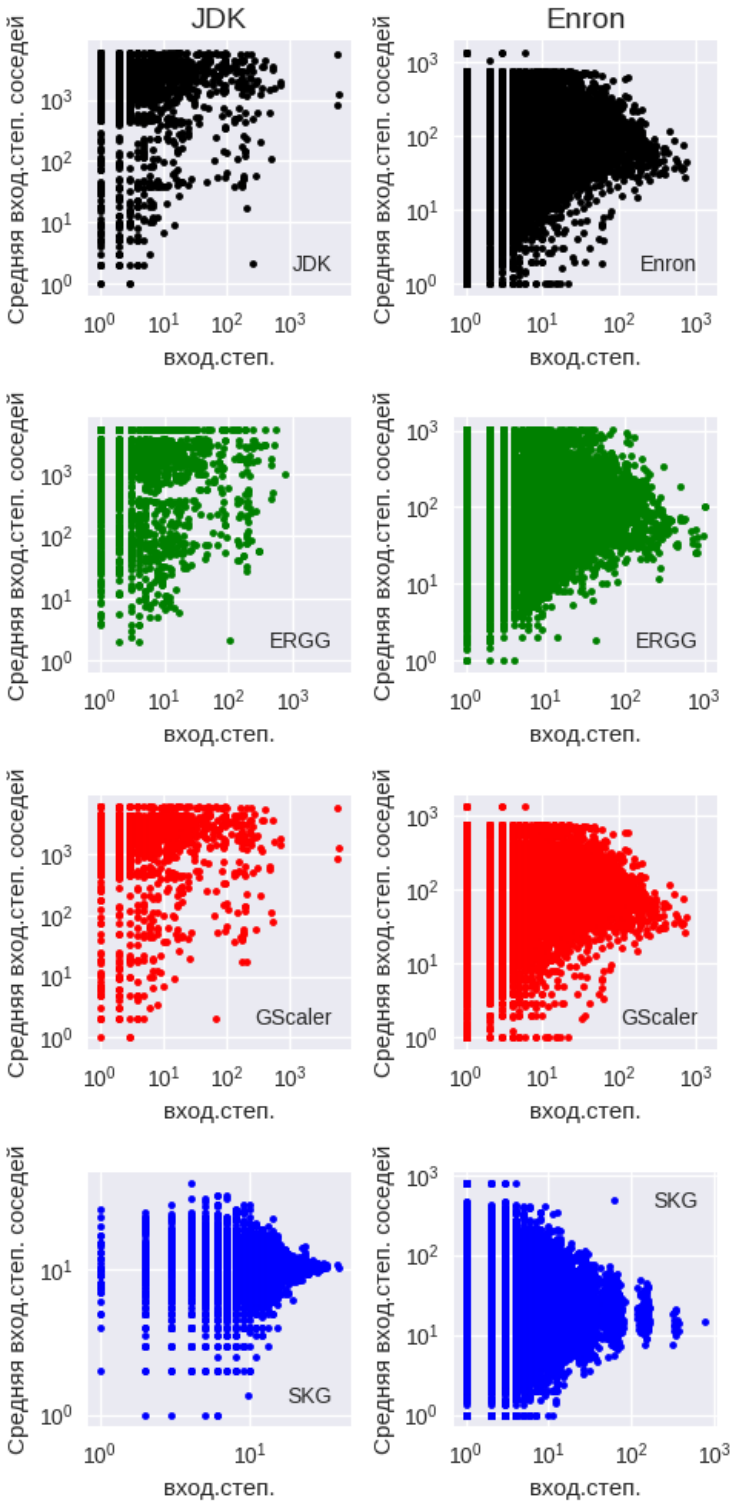

Рисунок Б.13 — Ассортативность входящих степеней (3 из 3).

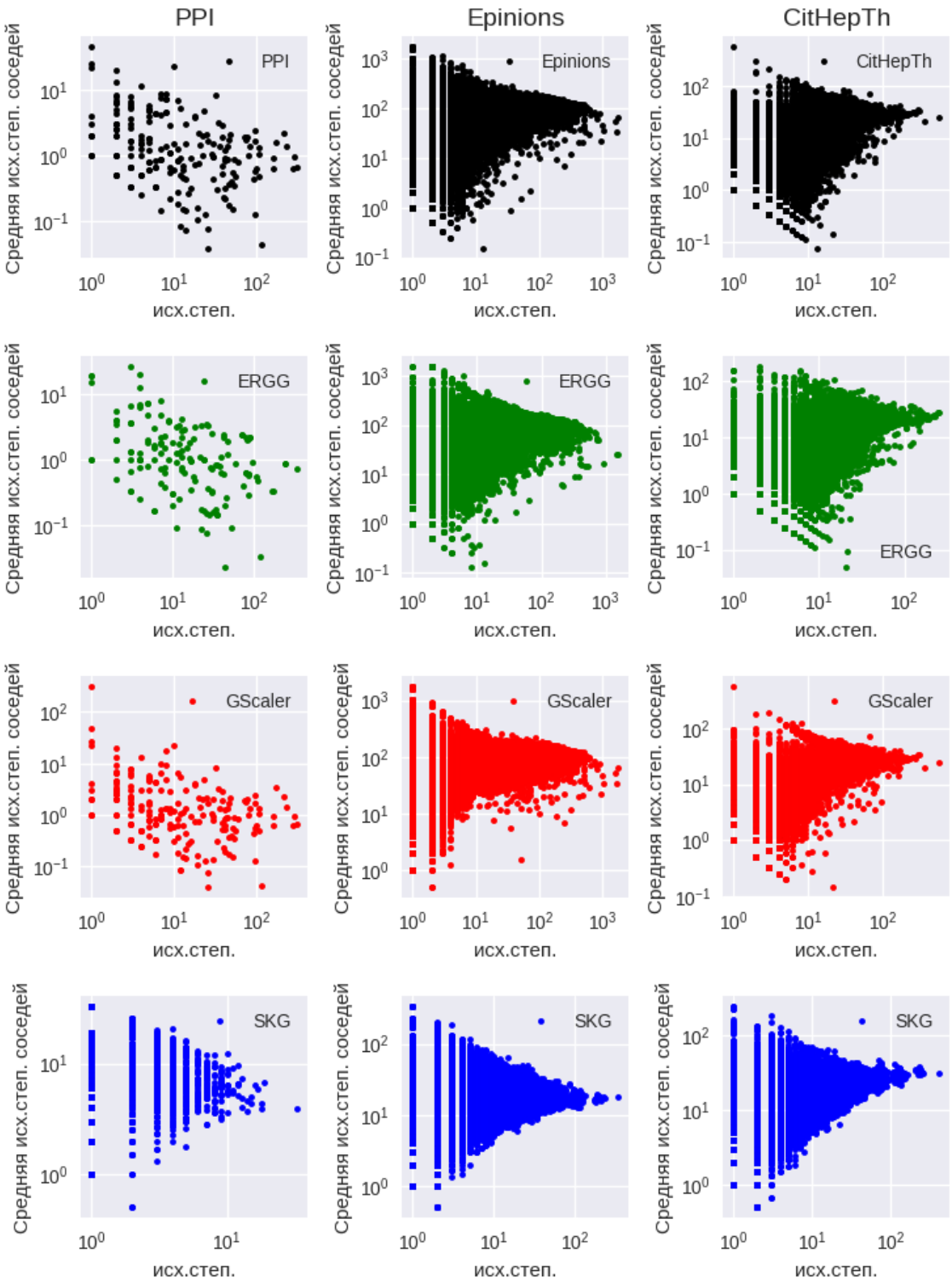

Рисунок Б.14 — Ассортативность исходящих степеней (1 из 3).

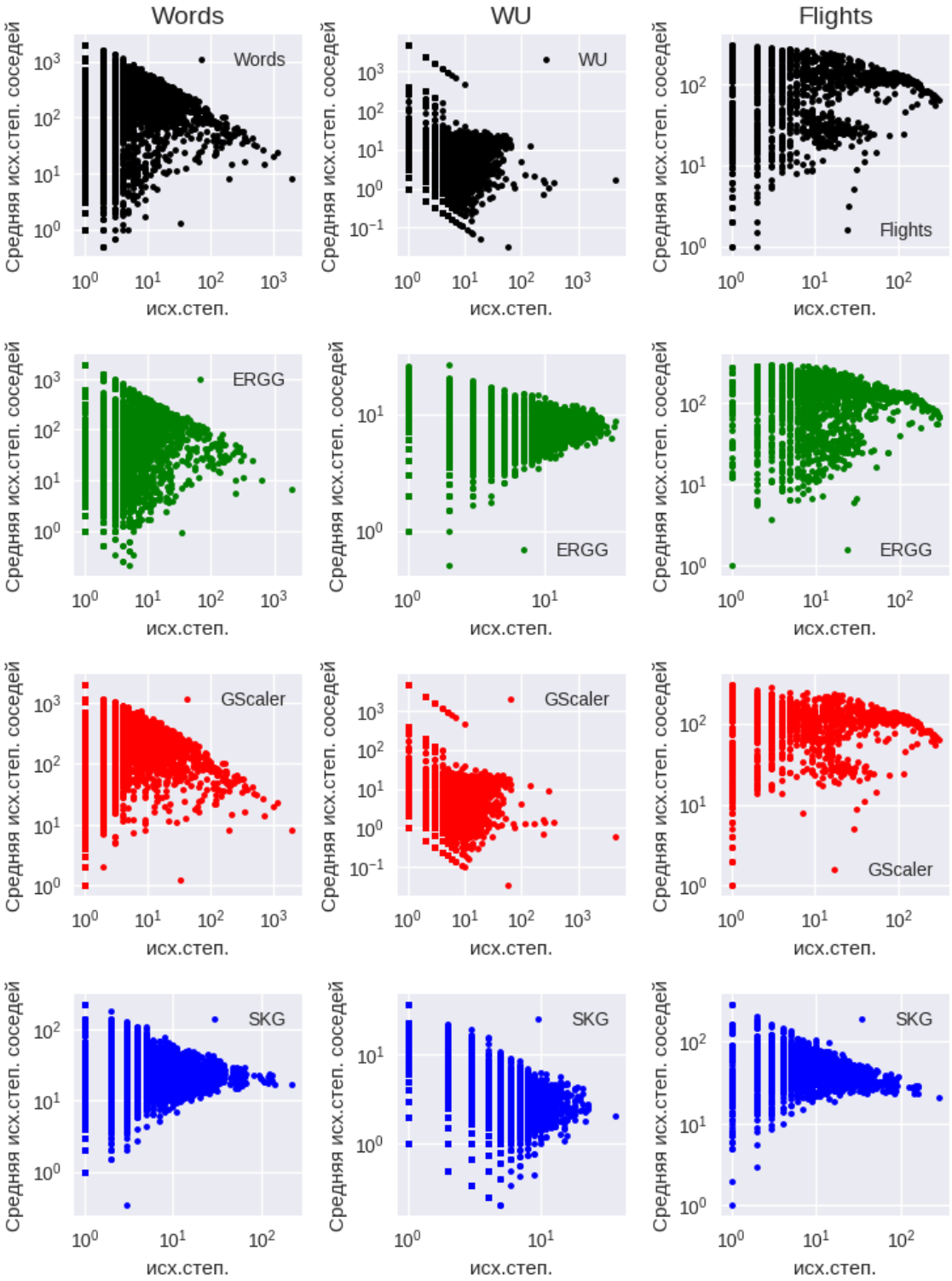

Рисунок Б.15 — Ассортативность исходящих степеней (2 из 3).

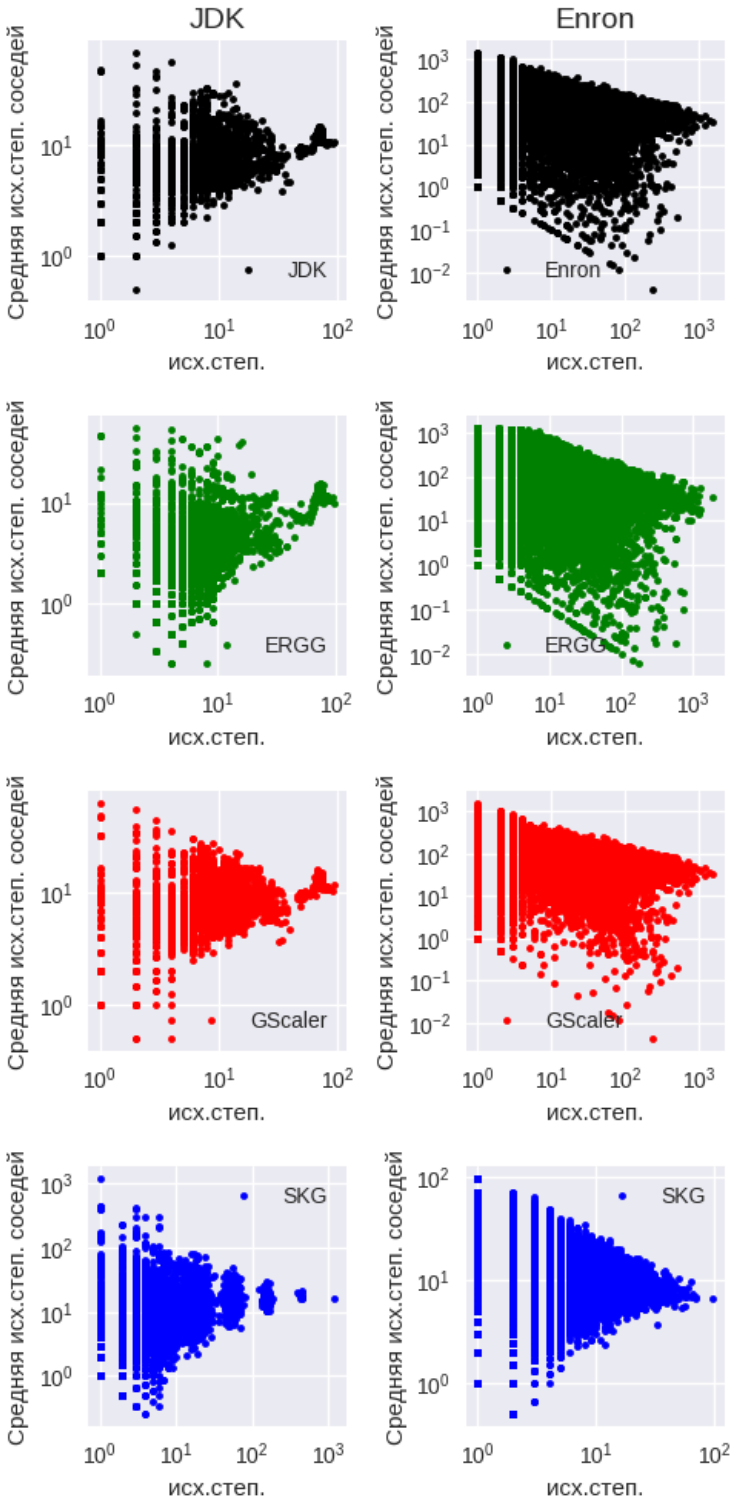

Рисунок Б.16 — Ассортативность исходящих степеней (3 из 3).

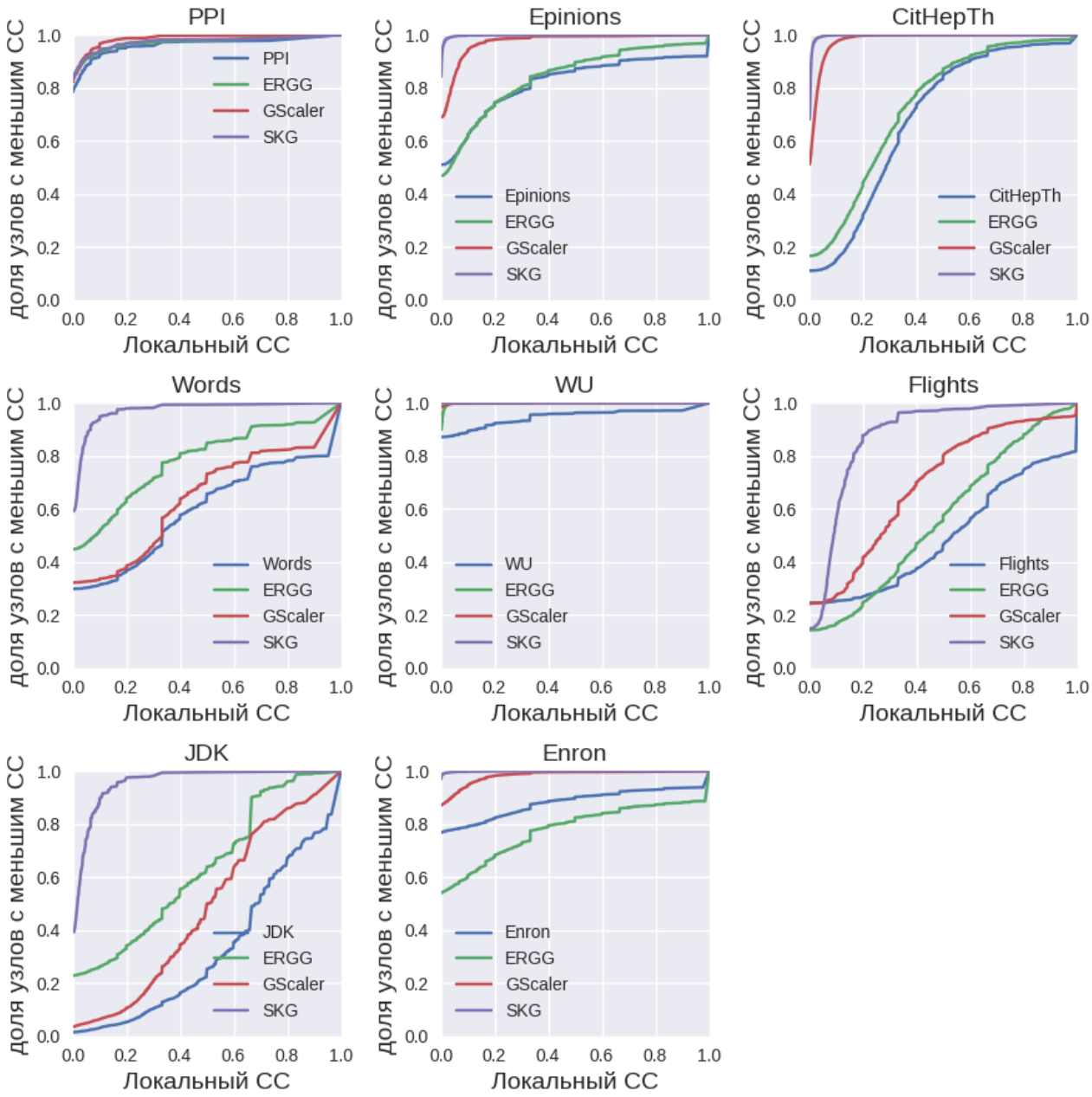

Рисунок Б.17 — Кумулятивное распределение коэффициента кластеризации (CC).

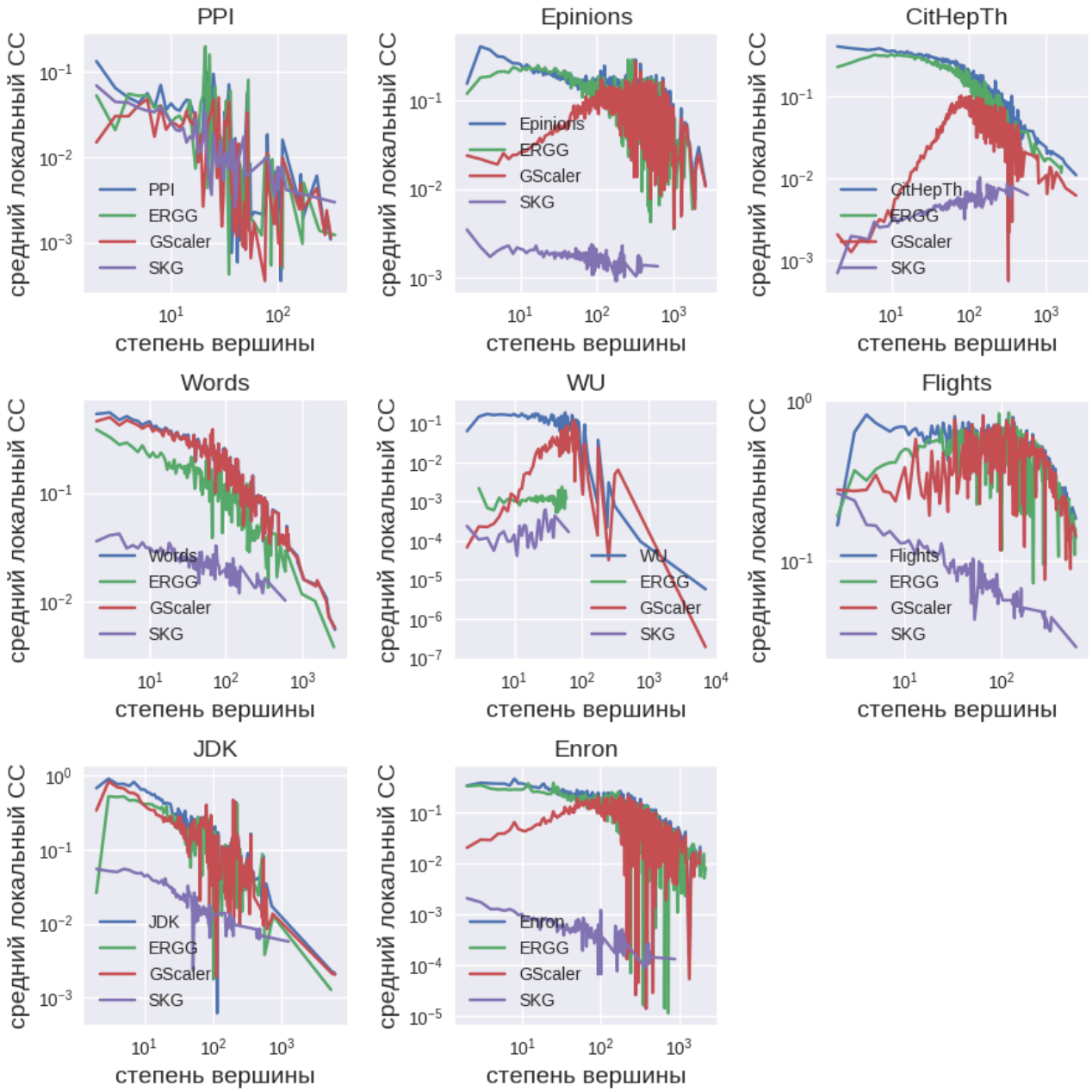

Рисунок Б.18 — Зависимость коэффициента кластеризации от степени узла.

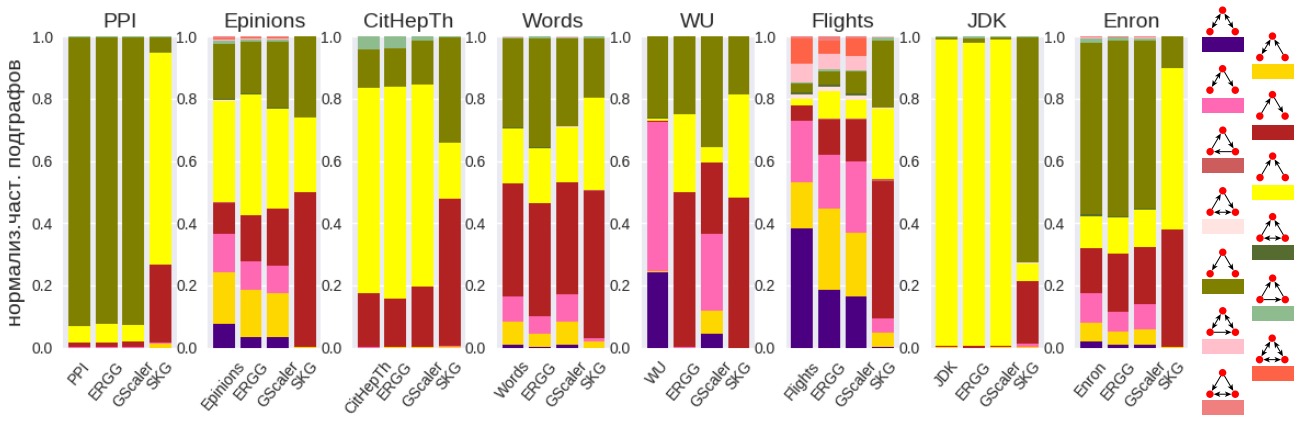

Рисунок Б.19 — Распределение подграфов размера 3 (3-GP).

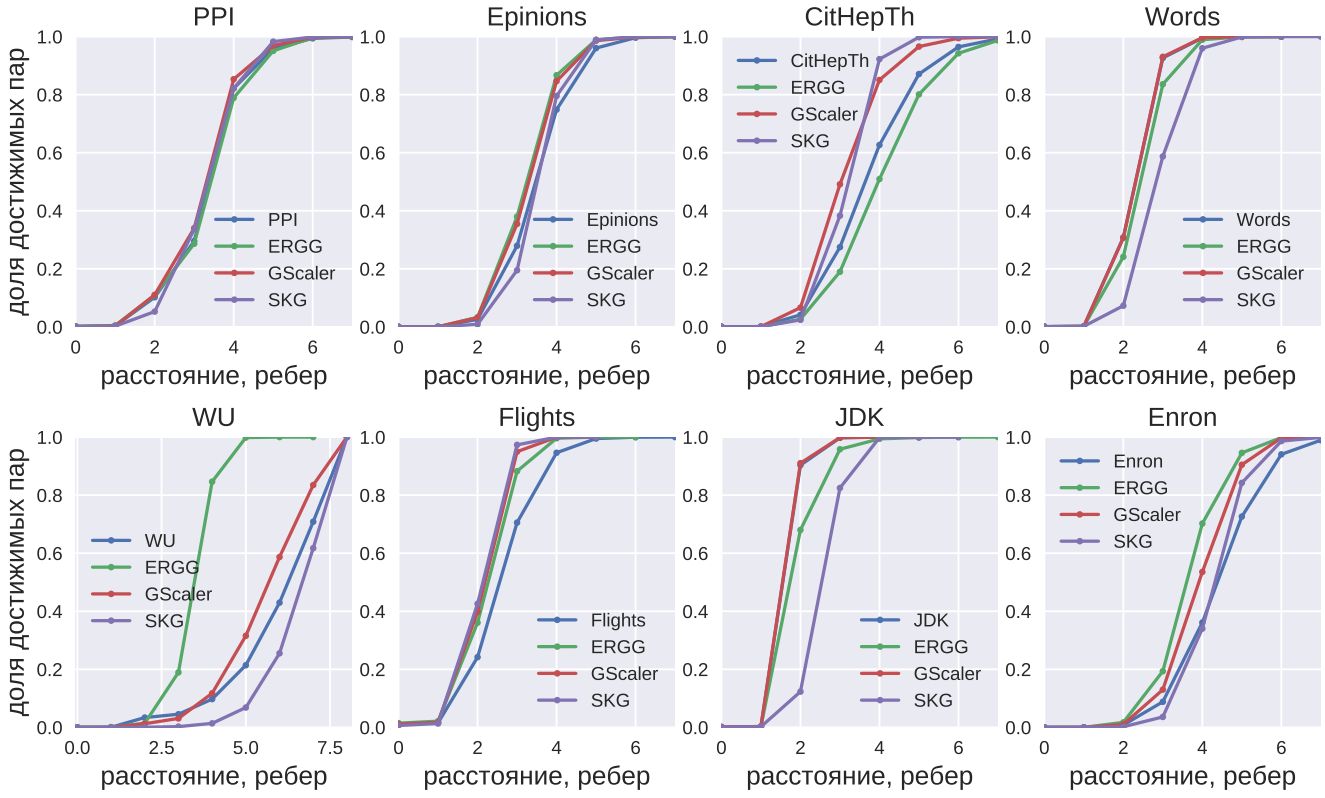

Рисунок Б.20 — График достижимости пар вершин (hop-plot).

Б.2 Измерение вариабельности

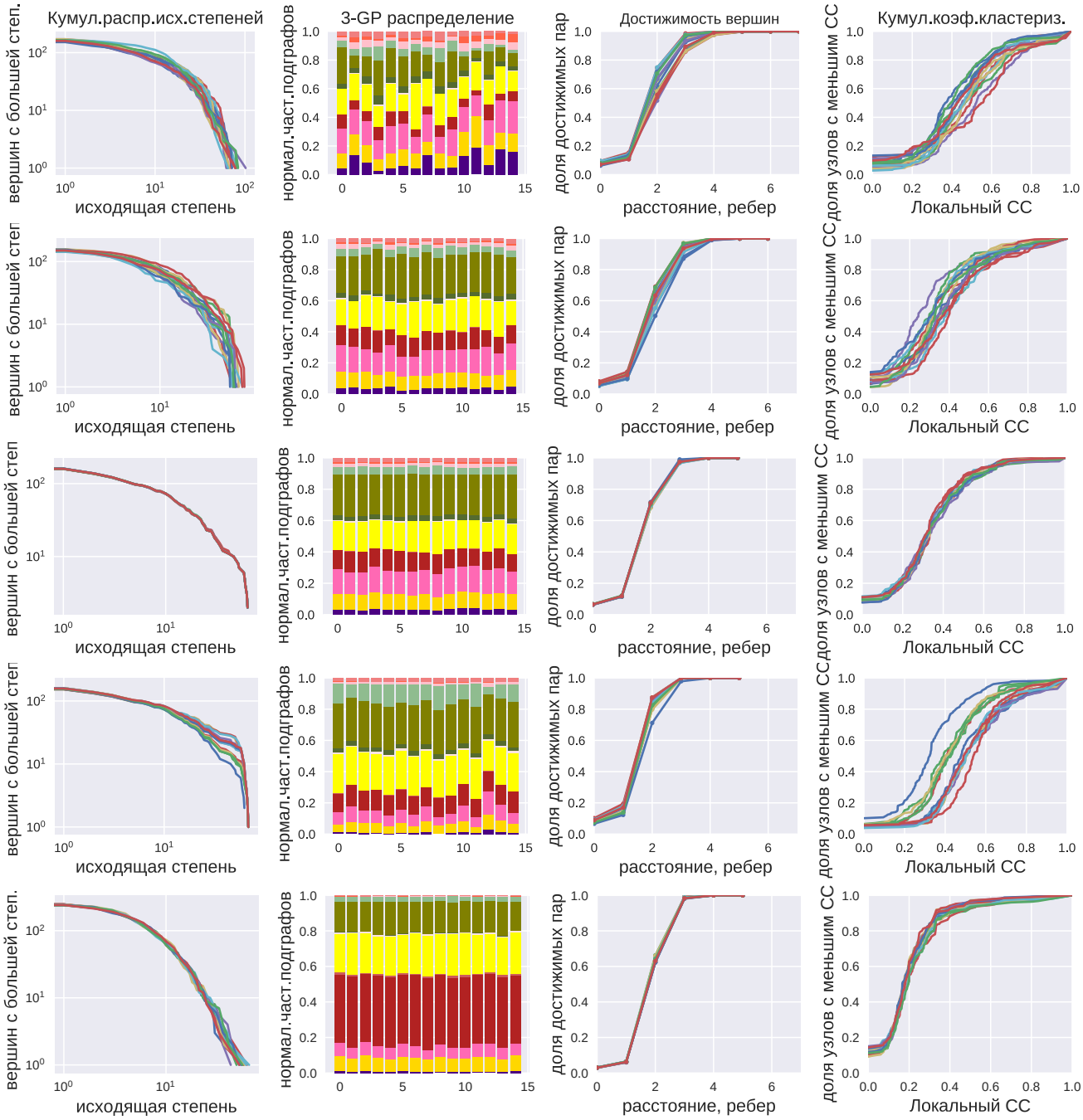

Рисунок Б.21 — Сравнение вариабельности признаков-распределений при имитации домена. Сверху вниз: оригинальный датасет (15 эго-сетей twitter с $n \in [170; 180]$ и $m \in [2000; 3000]$); ERGG-dwc; Gscaler; Gscaler+ (с искусственно заданным разбросом числа вершин и ребер); SKG.