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»

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Москва — 2019

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институте системного программирования им. В. П. Иванникова РАН.

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

Официальные оппоненты: **Райгородский Андрей Михайлович,**
доктор физ.-мат. наук, профессор,
главный научный сотрудник — заведующий лабораторией продвинутой комбинаторики и сетевых приложений Московского физико-технического института (МФТИ)

Добров Борис Викторович,
кандидат физико-математических наук,
заведующий лабораторией научно-исследовательского Вычислительного центра Московского государственного университета имени М. В. Ломоносова (НИВЦ МГУ)

Ведущая организация: Федеральный исследовательский центр "Информатика и управление" Российской академии наук (ФИЦ ИУ РАН)

Защита состоится «12» декабря 2019 г. в 15 часов на заседании диссертационного совета Д 002.087.01 при Федеральном государственном бюджетном учреждении науки Институте системного программирования им. В. П. Иванникова РАН по адресу: 109004, г. Москва,ул. А. Солженицына, дом 25.

С диссертацией можно ознакомиться в библиотеке и на сайте Федерального государственного бюджетного учреждения науки Институте системного программирования им. В. П. Иванникова РАН.

Автореферат разослан _____ 2019 года.

Ученый секретарь
диссертационного совета
Д 002.087.01,
кандидат физ.-мат. наук Зеленов С. В.

Общая характеристика работы

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*.

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. Позднее было обнаружено свойство безмасштабности во многих реальных сетях и предложены различные модели, его объясняющие, например, модель Барабаши-Альберт в 1999 году. 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. Известно, например, что простая перенумерация идентификаторов вершин социальной сети не предотвращает возможность выяснить существование связи между двумя пользователями.

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

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

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[2]:

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

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 в работе называется дисперсия их числовых характеристик в этом множестве.

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

1. Поиск закономерностей и извлечение статистических признаков из реальных данных.
2. Выбор признаков для моделирования графов, например, распределение степеней вершин, коэффициент кластеризации, диаметр графа.
3. Определение модели, задающей вероятностное 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,

[2]Определения указанных характеристик рассмотрены в подразделе 1.1.3.

например, *обучение* на заданном 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 организации узлов сети в более тесно связанные группы на основании общих 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: отклонения по каждой характеристике не выше соответствующих отклонений у других современных методов;
- вариабельность генерируемых графов: разброс значений числовых характеристик близок к соответствующему разбросу 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 удовлетворяющий 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 продемонстрирована 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 [1; 3] вместе с соавторами была поставлена задача и проводилась редакторская правка, остальная часть выполнена автором. В работе [2] 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 (вклад автора состоит в разработке концепции и ее реализации):

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

Объем и структура работы. Диссертация состоит из введения, трёх глав, заключения и двух приложений. Полный объем диссертации составляет 163 страницы с 42 рисунками и 8 таблицами. Список литературы содержит 183 наименований.

Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, формулируется цель и ставятся задачи работы, перечисляются основные положения, выносимые на защиту, излагается научная новизна, теоретическая и практическая значимость представляемой работы.

В **первой главе** дается обзор существующих подходов к генерации случайных графов. Вначале обсуждаются основные понятия и приводятся известные характеристики графов (раздел 1.1). Затем дается краткая сводка по существующим в литературе обзорам в области моделирования случайных графов и существующих классификаций подходов (раздел 1.2). В доступной литературе не было найдено удовлетворительного обзора существующих моделей случайных графов, а существующие попытки мало актуальны из-за отсутствия многих современных моделей, либо далеко не полны. Поэтому в следующем разделе (1.3) предлагается собственное видение области и представлена таксономия подходов к моделированию случайных графов с подробными примерами конкретных моделей и алгоритмов.

Предлагаемая иерархическая таксономия подходов состоит из трех классов верхнего уровня, основанных на различных мотивациях, лежащих в их основе (рисунок 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 Домен-специфичный класс 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. Далее в работе следует подробное описание и анализ их всех с иллюстрациями на конкретных моделях. Естественным образом некоторые модели встречаются в нескольких категориях, поскольку основаны на нескольких подходах. Хотя далеко не все релевантные модели упомянуты в каждой категории — целью является лишь проиллюстрировать подходы, — по возможности затронуто большинство известных моделей и генераторов графов.

Затем (раздел 1.4) следует обсуждение таксономии в контексте связи подходов между собой и использования их комбинаций в моделях случайных графов. После этого проведен анализ рассмотренных подходов в контексте шести основных приложений случайных графов: понимание сетей, анализ сетей, экстраполяция графов, тестирование (бенчмарки), нулевые модели и рандомизация графов.

В конце первой главы приводится краткий обзор генераторов графов, похожих на данный, и делается вывод о том, что существующие модели ориентированных случайных графов, похожих на данный, с автоматическим обучением не удовлетворяют остальным поставленным в работе требованиям, а именно: 1) произвольный размер генерируемых графов, 2) поддержка структуры сообществ и взвешенных ребер.

Во **второй главе** обосновывается и формулируется постановка задачи. Затем представлен подход, позволяющий автоматически обучаться на ориентированном графе из любого домена и генерировать похожие графы, масштабируя размер исходного графа на вещественный коэффициент. Предложенный подход основан на вложении исходного графа (***graph***

Рис. 1 — Таксономия подходов, встречающихся в моделях и генераторах случайных графов, с краткими описаниями каждой категории.

embedding) в низкоразмерное пространство, при котором признаки графа кодируются в наборе векторов его вершин, — ***Embedding based Random Graph Generation***, ERGG. Создание выборки графов одинакового размера позволяет тестировать статистическую значимость, а варьирование размера графа делает возможным оценку масштабируемости алгоритмов.

Как уже было отмечено, актуальной целью является возможность автоматически обучаться на заданном графе. В предложенном подходе ERGG сначала происходит обучение представления вершин (вложения) исходного графа векторами небольшой размерности. Затем из некоторого вероятностного распределения, которое аппроксимирует распределение выученных векторов вершин, сэмплируются новые вектора, соответствующие вершинам нового случайного графа. Наконец, полученные новые узлы связываются ребрами, завершая построение графа.

Формально, на вход ERGG подается граф $G = (N,E)$ и коэффициент масштабирования $x > 0$ ($x \in \mathbb{R}$). Будем обозначать число вершин в графе как $n = |N|$, число ребер $m = |E|$. 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.

Найдя 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 реализации такого случайного графа необходимо сэмплировать $\lfloor xn \rfloor$ векторов из $\mathcal{R}$ и, используя выученную функцию s_{ij}, соединить ребрами соответствующие пары новых узлов.

В рамках подхода ERGG предложен метод генерации случайных графов контролируемого размера, похожих на данный (раздел 2.4). Метод поддерживает направленные взвешенные графы со структурой сообществ — ERGG-dwc (от англ. ***directed***, ***weighted***, ***communities***). Метод ориентирован на обеспечение вариабельности (изменчивости) синтетических графов, сохраняя распределение степеней и распределение подграфов размера 3 близкими к таковым у исходного графа.

Формальное описание метода ERGG-dwc следующее. 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 параметром является величина шума ϵ со значением по умолчанию 0.2. Порядок выполнения алгоритма следующий:

1. **Вложение**. Обучить представление графа $G = (N,E)$ с помощью специального метода вложения COMBO 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 гауссовского шума $\vec{g} \sim \mathcal{N}(0, diag(\epsilon, ..., \epsilon))$ с небольшой амплитудой ϵ. 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 $G' = (N'',E')$.
4. **Атрибуты**.
    a) 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 $w_0 = \min_{(i,j)\in E} w_{ij}$.

Вместо решения задачи ERGG-dwc целиком, была по отдельности решена серия более простых подзадач. А именно: вложение + восстановление (подраздел 2.4.1), аппроксимация + сэмплирование (подраздел 2.4.2) и определение атрибутов (подраздел 2.4.3). Опишем более подробно все шаги алгоритма ERGG-dwc.

Вложение + восстановление. Первая подзадача — представить узлы данного графа векторами, сохраняя максимальное количество информации. Для этого был разработан метод вложения графа COMBO на основе существующих алгоритмов вложения BLM и LINE и ориентированный на максимизацию F_1-меры восстановленных ребер графа по отношению к исходным ребрам.

В методе COMBO каждой вершине графа соответствует вектор $\vec{r}_i$, состоящий из 3 компонент: $\vec{r}_i = \begin{bmatrix}\vec{u}_i & \vec{v}_i & Z_i\end{bmatrix}^T$. Вектора $\vec{u}_i$ и $\vec{v}_i$ ассоциированы с входным и выходным представлениями узла i, а Z_i — дополнительный коэффициент. В качестве функции оценки для пары вершин выступает билинейная модель из BLM: $s_{ij} = \vec{u}_i \cdot \vec{v}_j - Z_i$.

Смысл вычисления векторного представления графа (суть, обучения) состоит в нахождении таких параметров модели $\Theta = \{\vec{u}_i, \vec{v}_i, Z_i\}_{i=1}^n$, что функция $s_{ij} = s(\vec{r}_i, \vec{r}_j)$ оценивает ребра графа выше, чем пары узлов, не являющиеся ребрами. При этом размерность векторов представления d должна быть мала: $d \ll n$.

В качестве целевой функции максимизируется функция правдоподобия всего графа при условии модели, где s_{ij} выступает как вероятность ребра:

$$J_\Theta = \sum_{(i,j)\in E} \log p(i \to j) = \sum_{(i,j)\in E} \log s_{ij} \to \max_\Theta$$

Для решения этой задачи используется техника негативного сэмплирования (***Negative sampling***, при которой задача обучения плотности вероятности сводится к задаче отличения реального распределения (пары узлов-ребра) от шумогого (пары узлов-не-ребра). При этом целевая функция приобретает вид:

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

где шумовое распределение выбрано $p_n(j) \propto d_j^{3/4}$, шумовые пары узлов (в количестве ν для каждого ребра графа) фильтруются так, что только $(i,j') \notin E$ отбираются в качестве негативных примеров (шума). Вектора

вершин инициализируются из равномерного распределения как $\vec{u}_i, \vec{v}_i \sim \mathcal{U}[-\frac{1}{2\sqrt{d}}, \frac{1}{2\sqrt{d}}]$ и начальное значение $Z_i = \log n$.

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

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

$$= \sigma(-s_{ij}) \frac{\partial s_{ij}}{\partial \theta} + \sum_{j' \sim p_n(j')}^{\nu} \sigma(s_{ij'}) \frac{\partial s_{ij'}}{\partial \theta}, \quad (1)$$

где $\sigma(x) = \dfrac{1}{1+e^{-x}}$ — сигмоида, а производная от s_{ij}: $\dfrac{\partial s_{ij}}{\partial \theta} = \begin{bmatrix} -\vec{v}_j \\ -\vec{u}_i \\ 1 \end{bmatrix}$.

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 считается успешным, если его ребра можно восстановить с мерой $F_1 \geqslant 0.99$. 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.

Аппроксимация распределения + сэмплирование. После обучения представления исходного графа вектора его узлов содержат информацию о структуре графа. Основное предположение заключается в том, что ключевые статистические свойства графа закодированы в *распределении* векторов $\{\vec{r}_i\}_{i=1}^n$, а не в отдельных векторах. Далее, сэмплируя набор новых векторов узлов из $\mathcal{R}$ и строя новый граф согласно описанной выше процедуре, ожидается, что он будет иметь схожие характеристики.

Независимо от метода аппроксимации в рамках подхода ERGG для графов, генерируемых одной моделью, справедлив квадратичный закон роста числа ребер в зависимости от числа узлов: $m \propto n^2$, что доказывается следующей теоремой.

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 $m \propto n^2$.

В качестве модели распределения $\mathcal{R}$ в ERGG-dwc был выбран метод пересемплирования с добавлением шума. Он состоит в простом запоминании всего набора векторов $\{\vec{r}_i\}_{i=1}^n$ и добавлении гауссовского шума с малой амплитудой ϵ. Чтобы сэмплировать новый элемент из $\mathcal{R}$, случайным образом выбирается индекс $i \in \{1..n\}$ и возвращается вектор $\vec{r}_i + \vec{g}$, где $\vec{g} \sim \mathcal{N}(0, diag(\epsilon, ..., \epsilon))$.

Атрибуты: метки сообществ и веса ребер. Последняя подзадача заключается в корректной обработке и назначении меток сообществ и весов ребер в сгенерированном графе. При этом их согласованность с топологией графа должна быть сохранена: известно, например, что плотность ребер внутри сообществ выше, чем между сообществами.

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 предлагается наследовать эти метки в сгенерированном графе из исходного, используя метод пересэмплирования. Если вектор узла $k \in N'$ нового графа был сэмплирован из вектора узла $i \in N$ исходного, 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.

Аналогично, чтобы назначить веса ребер в сгенерированном графе, наследуются веса ребер исходного графа при соответствующих узлах. 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: $w'_{kl} = w_{ij}$. Если же исходный граф G(N,E) не имеет ребра (i,j), то выбирается минимальный вес (k,l): $w_0 = \min_{(i,j)\in E} w_{ij}$.

В конце главы оценивается вычислительная сложность алгоритма (подраздел 2.4.4), которая дается следующей теоремой.

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

Результаты, изложенные во второй главе, опубликованы в работе [2].

Третья глава посвящена программной реализации описанного алгоритма ERGG-dwc и его экспериментальным исследованиям.

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. Обеспечивает хранение реальных графов в определенном формате и доступ к ним.

2. Модуль обучения представления графа. Реализует обобщенный метод вложения COMBO с возможностью настройки параметров.
3. Работа с представлением графа. Организовано хранение векторных представлений для графа и различные методы аппроксимации распределения и сэмплирования.
4. Генерация графа. Модуль отвечает за этап соединения вершин графа ребрами по заданной схеме на основе его векторного представления. Кроме того, встроен запуск нескольких известных генераторов графов.
5. Подсчет характеристик. Доступен ряд известных графовых характеристик для анализа графов, реализовано построение графиков с результатами.
6. Фреймворк для тестирования. Предназначен для проведения анализа на всех этапах разработки и сравнительного тестирования моделей. С помощью фреймворка проведены все эксперименты и построены графики в настоящей работе.

Важной особенностью архитектуры является параллельная реализация алгоритма вложения COMBO, выполненная на языке `cython`, которая по производительности не уступает оригинальным реализациям BLM и LINE. Алгоритм построения ребер графа также поддерживает параллельное исполнение, поскольку этот этап является наиболее вычислительно затратным во всем алгоритме ERGG-dwc.

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

В процессе разработки алгоритма ERGG-dwc проводились экспериментальные исследования возможных решений каждой из рассмотренных подзадач (раздел 3.2). В результате были установлены оптимальные компоненты и параметры метода вложения COMBO (подраздел 3.2.1). В качестве аппроксимации распределения рассмотрены несколько вариантов и выбран метод пересэмплирования векторов с добавлением гауссовского шума и определена оптимальная амплитуда шума (подраздел 3.2.2). Проверена корректность способа присвоения атрибутов вершин на основе наследования меток вершин — путем сравнения модулярности [1] для исходных и сгенерированных сообществ (подраздел 3.2.3). Наконец, измерена производительность алгоритма ERGG-dwc (подраздел 3.2.4).

В разделе 3.3 проведено экспериментальное сравнение алгоритма ERGG-dwc с другими методами на графах из различных доменов с применением известных графовых характеристик.

Основной целью сравнения было проанализировать способность моделей случайных графов имитировать графы из различных графовых доменов. С одной стороны, для адекватного моделирования необходимо сохранение свойств исходного графа. При этом ясно, что изменение

нескольких ребер не обеспечит достаточного разнообразия в синтетических графах для проведения надежного тестирования значимости графовых алгоритмов. Поэтому хорошая модель графа должна удовлетворять двум требованиям: во-первых, сгенерированные графы должны быть *похожи* на исходный граф с точки зрения характеристик графа, а во-вторых, они должны имитировать *вариабельность* реальных сетей в одном домене.

Учитывая еще требование автоматического обучения, наиболее релевантными моделями являются Стохастические Кронекеровские графы (SKG)и алгоритм Gscaler, которые способны принимать на вход заданный граф и масштабировать его с заданным произвольным коэффициентом, не требуя дополнительных параметров.

Таблица 1 — Коллекция направленных взвешенных графов.

Описание графа	**домен**	**имя**	**n**	**m**
Белковые взаимодействия	биологич.	PPI	2 239	6 452
Сеть доверия Epinion	социальный	Epinion	49 288	487 183
Цитирования статей	информац.	CitHepTh	27 770	352 807
Смежность слов в текстах	информац.	Words	7 381	462 81
Мобильные звонки	социальный	WU	72 146	100 974
Перелеты	технол.	Flights	1 574	28 236
Зависимости софта JDK	технол.	JDK	6 434	53 892
E-мейлы	социальный	Enron	87 273	321 918

Для проведения тестирования были выбраны 8 направленных взвешенных графов размером от тысячи до сотни тысяч вершин из различных доменов. Описание коллекции данных представлено в таблице 1.

Поскольку не существует универсальной информативной метрики для оценки сходства графов, было проведено сравнение графов по набору известных характеристик.

Используемые характеристики разделены на три группы в соответствии с классификацией в подразделе 1.1.3.

- Степень вершины: распределение входящих и исходящих вершин, ассортативность степеней вершин.
- Подсчет подграфов: коэффициент кластеризации и его зависимость от степени вершины, кумулятивное распределение коэффициента кластеризации, распределение подграфов размера 3.
- Связность: достижимость вершин, радиус, диаметр и эффективный диаметр гигантской компоненты.
- Спектр: спектральный радиус.

Все указанные характеристики были вычислены для каждой модели (ERGG-dwc, SKG, GScaler) на каждом графе из коллекции (раздел 3.3.2). В результате экспериментов было установлено, что разработанный метод

ERGG-dwc не уступает современным методам в смысле близости сгенерированных графов к исходным по известным графовым характеристикам. При фиксированном пороге по отклонению в 10%, из 72 измерений меньшее отклонение по характеристике было в 37 измерениях для ERGG-dwc, в 49 для Gscaler и в 8 для SKG.

Таблица 2 — 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-

Для оценки вариабельности сгенерированных графов (подраздел 3.3.3) сравнивался разброс по нескольким числовым графовым характеристикам с соответствующими разбросами в реальных графах из одного домена.

В качестве коллекции данных была использована коллекция эгосетей twitter, из которой выбрано 15 графов, близких по числу узлов ($n \in [170; 180]$) и числу ребер ($m \in [2000; 3000]$). Набор сгенерированных данных был построен путем обучения модели на одном случайно выбранном графе из этой коллекции, и запуска генератора 15 раз. Результат тестирования представлен на рисунке 2 (красной стрелкой показаны значения характеристик для исходного графа) и таблице 3.

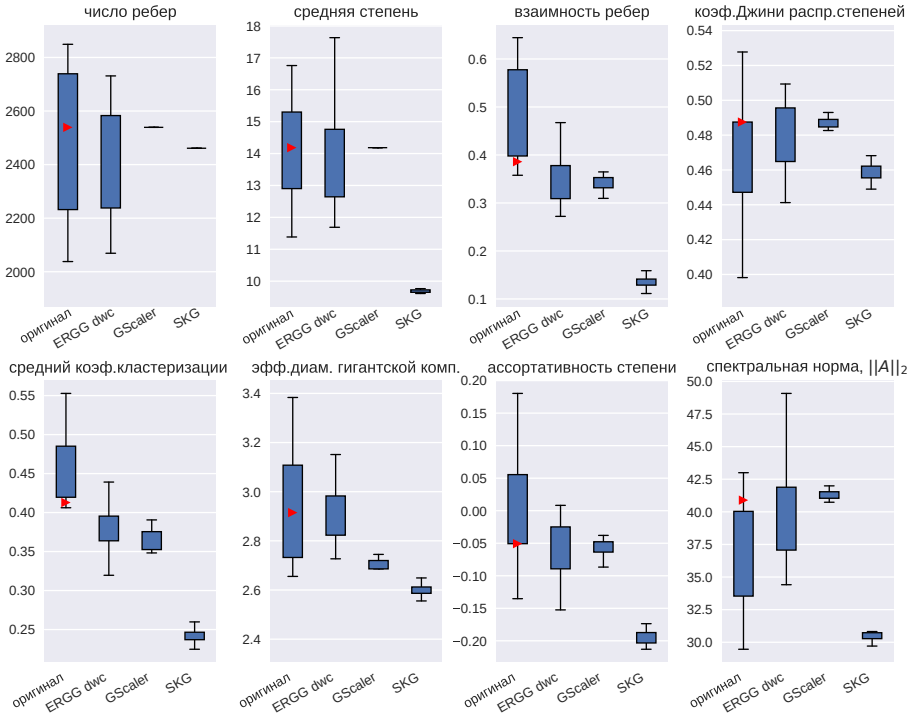

Рис. 2 — Сравнение вариабельности числовых характеристик при имитации домена. Оригинальная коллекция: 15 эго-сетей twitter с $n \in [170; 180]$ и $m \in [2000; 3000]$; ERGG-dwc; Gscaler; Gscaler+ (с искусственно заданным разбросом числа вершин и ребер); SKG. Красной стрелкой отмечено значение характеристик для графа, на котором происходило обучение.

На диаграммах размаха (рисунок 2) можно видеть, что разброс в характеристиках в сгенерированном наборе графов в случае ERGG-dwc гораздо ближе к оригинальному, чем аналогичный разброс, полученный с помощью двух других алгоритмов. Метод Gscaler генерирует графы, достаточно похожие на заданный в смысле графовых характеристик, однако вариабельность генерируемых графов недостаточна для имитации оригинальной вариабельности, наблюдаемой в исходной выборке графов из одного домена. В терминах дисперсии (таблица 3), 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%.

Таблица 3 — 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	**Исходная 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%)

Аналогичный результат получен и при сравнении вариабельности характеристик-распределений: распределение входящих и исходящих степеней вершин, распределение подграфов размера 3, кумулятивный коэффициент кластеризации, достижимость вершин (графики представлены в подразделе 3.3.3).

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 наборов данных для надежной проверки качества работы алгоритмов майнинга графов. Возможность генерирования графов контролируемого размера с сохранением исходных признаков позволяет также тестировать масштабируемость алгоритмов.

Доказаны теоремы о вычислительной сложности и масштабировании графов. Налагаемые ими ограничения на масштабируемость метода не являются существенными для его применимости в целевом классе задач.

Основные результаты третьей главы опубликованы в работе [3].

В **заключении** приведены основные результаты работы, которые 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.

Публикации автора по теме диссертации

1. *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.

2. *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.

3. *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.

4. *Filippov A., Drobyshevsky M., Korshunov A. et al.* Network analysis tool testing. — 23.08.2018. — Патент РФ 2018/151619. URL: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2018151619.

Дробышевский Михаил Дмитриевич

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

Автореф. дис. на соискание ученой степени канд. физ.-мат. наук

Подписано в печать ___.___._____. Заказ № _____
Формат 60×90/16. Усл. печ. л. 1. Тираж 100 экз.
Типография _____