МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

МОДЕЛЬ ТРОЙСТВЕННОГО ЗАМЫКАНИЯ ГЕНЕРАЦИИ СЛУЧАЙНОГО ГРАФА

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Летеева Артура Шавкетовича

Научный руководитель
к.ф.-м.н., доцент ____________ С. В. Миронов

Заведующий кафедрой
к.ф.-м.н., доцент ____________ А. С. Иванов

Саратов 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Кроме очевидной задачи моделирования динамики социальных сетей, модели случайных графов [1] применимы к таким популярным сейчас практическим задачам, как поиск сообществ, максимизация распространения влияния, оценка безопасности и выносливости сложных сетей и другие. Они позволяют сравнивать качество предлагаемых алгоритмов на целых семействах случайных графов и сравнивать получаемые результаты с предсказанными теоретически.

Данное направление исследований переживает период интенсивного развития и привлекает внимание математиков, инженеров, специалистов по компьютерным наукам, социологов и других специалистов.

Данная бакалаврская работа представляет из себя исследование модели построения случайного графа, основа которого состоит из двух принципов: тройственного замыкания и предпочтительного соединения. Важность исследования достаточно высока, поскольку множество моделей построенных по такой же концепции, очень точно описывают поведение и характеристики актуальных веб-графов.

Целью работы является эмпирическое изучение модели построения случайного графа [2]. Для достижения этой цели ставятся следующие задачи:

— написать программу для генерации построения случайного графа с помощью предложенной модели в которой: на вход приходит некоторое количество графов T и количество вершин в каждом графе M, на выход — рисунок сети состоящей из заданного количества графов и вершин;
— сравнить реализацию, использующую для представления графов матрицы смежности с реализацией, использующей библиотеку для построения графов и сетей;
— провести анализ результатов использования предложенной модели.

1 Постановка задачи

Большое количество систем во многих отраслях науки можно смоделировать с помощью механизма тройственное замыкание (triple closure), его используют в сложных сетях таких как социальные сети [3], компьютерные и мобильные сети, финансовые сети [4], а также метаболические сети некоторых организмов которые имеют логарифмически растущую среднюю геодезическую длину (кратчайший путь) [5]. Кроме того, социальные сети обычно демонстрируют высокую кластеризацию, или локальную транзитивность: если человек А знает В и С, то В и С, скорее всего, знают друг друга.

Стандартная модель тройственного замыкания (triple closure), предложенная Питтером Холмом и Бом Джун Кимом [2], представляет собой модификацию модели предпочтительного соединения (preferential attachment), которую принято называть «модель Барабаши—Альберт». Эта модель описывает ряд важных эмпирических закономерностей в поведении веб-графов, в последствии она получила большую популярность в этой среде.

Как и модель Барабаши—Альберт [6], модель тройственного замыкания генерирует безмасштабные сети [7]. Однако в отличие от модели Барабаши—Альберт, модель предложенная Питтером Холмом и Бом Джун Кимом создает сети с гораздо более высоким коэффициентом кластеризации по сравнению с реальными социальными сетями. [8] В данной модели вершина соединяется с уже существующей вероятностью, пропорциональной степени этой вершины (такая же как и в модели Барабаши—Альберт). Но для каждой из оставшихся ребер с вероятностью p выбирается сосед вершины, с которой произошло соединение на прошлом этапе, и с вероятностью $1 - p$ этап пропускается, т.е выбирается вершина с помощью предпочтительного соединения (preferential attachment).

Модель генерирует графы с разными коэффициентами кластеризации в зависимости от p. Степенное распределение же подчиняется степенному закону с коэффициентами -3 для любого p. В данной работе рассматривается модель построения сети, которая основана на двух принципах тройственного замыкания (triple closure) и предпочтительного соединения (preferential attachment), исследования поведения модели является основной целью работы.

Выполняется следующий список выполнения задач:

— написать программу для генерации построения случайного графа с помощью предложенной модели в которой: на вход приходит некоторое количество графов T и количество вершин в каждом графе M, на выход — рисунок сети состоящей из заданного количества графов и вершин;
— сравнить реализацию, использующую для представления графов матрицы смежности с реализацией, использующей библиотеку для построения графов и сетей;
— провести анализ результатов использования предложенной модели.

1.1 Детали реализации

Реализация модели построения случайного графа будет проводиться на языке программирования Python [9, 10].

Реализация модели осуществляется в двух вариантах:

— с использованием матрицы смежности, с представлением матрицы в виде двумерного массива;
— с использованием специальной библиотеки для представления графов и сетей.

1.1.1 `matplotlib.pyplot`

Библиотека `matplotlib.pyplot` состоит из множества модулей, они наполнены разными классами и функциями, которые иерархически связаны между собой. В программе же используется интерфейс `pyplot` т.к, в нем уже присутствуют готовые настройки рисунков и нам просто необходимо выбрать такой, какой нам в конкретный момент нужен.

1.1.2 `random`

Модуль `random` предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.

Непосредственно в самом коде приложений будем активно использовать функцию `choice`. Данная функция может принимать один аргумент — список элементов: в этом случае функция выдаст случайный элемент списка с равновероятным выбором.

Когда функция `choice` принимает два аргумента, второй аргумент представляет список чисел, в котором столько же элементов, сколько в первом аргументе. В этом случае она выдает элемент первого списка с вероятностью пропорциональной соответствующему элементу второго списка.

1.1.3 `numpy`

Библиотека `numpy` добавляет в Python поддержку больших многомерных массивов и матриц, вместе с большой коллекцией высокоуровневых (и очень быстрых) математических функций для операций с этими массивами. В реализации использовались следующие функции библиотеки.

`full(shape =, fill_value =, dtype =)` — данная функция создает массив, заполненный тем же значением. Параметр `shape` определяет форму входного массива. формой массива является количество строк и столбцов. Параметр `fill_value` — значение которое будет использоваться в качестве отдельных элементов массива. Параметр *dtype* — позволяет указать тип данный элементов выходного массива.

`reshape` — функция меняет размерность массива.

1.1.4 `networkx`

Библиотека `networkx` предназначена для работы с графами и другими сетевыми структурами. Основными возможностями библиотеки являются:

— классы для работы с простыми, ориентированными и взвешенными графами;
— встроенные процедуры для создания графов базовых типов;
— визуализировать сети в виде 2D и 3D графиков;
— получение таких характеристик графа как степени вершин, высота графа, диаметр, радиус, длины путей, центр.

Библиотека `networkx` является очень производительной и может оперировать весьма большими сетевыми структурами, уровня графа с 10 миллионами узлов и 100 миллионами дуг между ними. В виду того, что он базируется на низкоуровневой структуре данных языка Python под названием «словарь-словарей», память расходуется эффективно, графы хорошо масштабируются, мало зависят от особенностей операционной системы в которой выполняется скрипт и отлично подходят для популярного на данный момент направления по анализу данных из социальных сетей и графов. Также в библиотеке реализовано большое количество типичных для работы над графами алгоритмов. Реализованы такие алгоритмы как нахождение кратчайшего пути, поиска в высоту и ширину, кластеризация, нахождение изоморфизма графов и многое другое.

В работе были использованы следующие команды библиотеки.

`Graph()` — создание простого неориентированного графа. Дополнительные вершины между двумя узлами игнорируются, возможны узлы соединённые с самим собой.

`G.add_node(i)` — процедура для добавления в граф `G` вершины `i`.

`G.add_edge(i, j)` — процедура для добавления в граф `G` ребра между вершинами `i` и `j`.

`G.degree` — функция выдает вектор степеней вершин графа `G`.

`draw_circular(G, **options)` — процедура чертит граф `G` с круговой разметкой. Параметр `**options` предоставляет опции для отрисовки графа, с помощью которых можно изменить цвет вершин и ребер, размер, фон и многое другое.

2 Простая модель тройственного замыкания

Предметом исследования настоящей работы является простая модель тройственного замыкания.

Модель тройственного замыкания (triple closure) была предложена Питтером Холмом и Бом Джун Кимом и описывает модель в которой рост сети происходит итеративно согласно следующим правилам:

1. (*Рост*) На каждой итерации t добавляется новая вершина;
2. На каждой итерации в сеть добавляется m ребер, и каждая новая вершина t соединяется с m вершинами следующим образом:
   а) (*Предпочтительное соединение*) новый узел соединяется с одной из существующих вершин i с вероятностью, пропорциональной ее степени $\deg(i)$;
   б) оставшиеся $m - 1$ вершины соединяются с новой следующим образом:
   — (*Формирование триады*) с вероятностью p новая вершины t соединяется со случайно выбранном соседом вершины i, которая была выбрана в шаге предпочтительного соединения.
   — с вероятностью $1 - p$ вершина t соединяется с любой вершиной сети s с вероятностью пропорциональной степени вершины $\deg(s)$

Данное описание модели показывает, что рост степеней узлов сети подчиняется степенному закону:

$$\overline{k}_i(t) = m\left(\frac{t}{i}\right)^{\frac{1}{2}}, \tag{1}$$

где $\overline{k}_i(t)$ обозначает ожидаемую степень вершины i на итерации t. Питтер Холм и Бом Джун Ким показали, что распределение степеней также подчиняется степенному закону с показателем $\gamma = 3$, т.е $p_k \sim k^{-3}$, где p_k обозначает вероятность того, что случайно выбранная вершина имеет степень k.

Благодаря этапу формирования триады (Triad formation), коэффициент кластеризации модели П. Холма и Б. Д. Кима выше, чем в модели Барабаши—Альберт, а геодезическая длина в обеих моделях растет как логарифм от N, где N — размер сети.

2.1 Описание предложенной модели

Модель в данный работе, будет опираться на принципы тройственного замыкания и предпочтительного соединения [11], т.е в какой-то степени это частный случай модели, предложенной Питтером Холмом и Бом Джун Кимом. Правила по которым будет строиться безмасштабная сеть:

— на каждой итерации t добавляется новая вершина,ей приписывается метка t;
— вершина t соединяется ребром с вершиной u с вероятностью, пропорциональной её степени $\deg(u)$;
— среди вершин, смежных с вершиной u проводится равновероятный выбор одной из них — v, с которой вершина t соединяется вторым ребром.

Интересом анализа новой модели будет выполняться в следующих моментах:

— степенное распределение будет также подчиняться степенному закону с показателем 3;
— динамика роста степени вершины, которая будет выполнять асимптотику выше освещенных моделей, и будет изменяться как квадратный корень от времени.

2.2 Построение графов

Было проведено несколько серий экспериментов. В первой серии экспериментов в реализации программы с использованием библиотеки `networkx` выводились построенные графы для малых значений `M`.

- Первый шаг однозначный. После него имеем полный граф. От второго шага зависит, какие из двух вершин, имеющихся на текущий момент увеличат свою степень. Такими вершинами оказались вершины с номерами 0 и 1. С большей вероятностью последующие шаги будут выбирать эти вершины, для проведения очередного ребра;
- Следующий шаг добавление 4-й вершины это подтверждает. Выбрана вершина 1;
- При добавлении 5-й вершины, выбрана вершина 0;
- Дальнейшие шаги приводят к росту степени вершины 1 и смежных с ней вершин. Так можно заметить, что несмежная с ней вершина 5 после окончания построения графа так и осталась вершиной со степенью 2.

2.3 Сравнение работы двух реализаций

Во второй серии экспериментов программы запускались для больших значений параметра `M`. Сравнивалось время работы алгоритмов. Результаты запусков приведены в таблице.

Таблица 1 – Время работы реализаций
с разными параметрами `M` и `T`

`T`	`M`	Время построения набора графов, сек	
		`numpy`	`networkx`
100	100	0.499	0.364
50	200	0.868	0.639
20	1000	7.779	6.054
20	10000	620.195	589.299
10	50000	-	8136.447
		14322.712	13346.453

В ходе выполнения экспериментов было выявлено, что выполнение программы с использованием матрицы смежности работа не только выполнялась дольше, чем с использованием библиотеки `networkx`, но еще и оказалась более требовательным к памяти при больших значениях параметров. Практически все запуски приложений проводились на компьютере с процессором i5 4.8 Ghz и 4-мя ядрами с оперативной памятью 8-ми гигабайт. Но исполнение на этом компьютере приложения, использующего матрицу смежности, при значении параметра `M` равном 50000, оказалось невозможным. Этим обусловлен прочерк в первой строке таблицы, соответствующей этому параметру. Исполнение программы с использованием библиотеки `networkx` оказалось успешным.

Вторая строка результатов при `M` равном 50000 получена на компьютере с процессором i5 2.5 Ghz и 2-мя ядрами.

2.4 Результат работы

В ходе выполнения программ собиралась и сохранялась в файл информация о среднем значении количества в построенных графах вершин с определенной степенью. В конце работы алгоритмов эта информация представляется в виде графиков.

Несмотря на то, что графы генерировались случайным образом, графики распределения степеней оказались практически идентичными. Часть графиков от 0 до 10^2 почти вырождена в отрезок прямой.

Визуализированные результаты вычислений, показывают, что коэффициент угла наклона прямой на логарифмически шкалированном графике распределения степеней равен -3. Это подтверждает теоретические результаты представленные в [2], которые говорят о том, что сети, генерируемые данной моделью являются безмасштабными со степенным законом распределения степеней, экспонента которого имеет значение -3.

ЗАКЛЮЧЕНИЕ

В данной работе была изучена модель построения генерации случайного графа, основанном на принципах предпочтительного соединения и тройственного замыкания, являющаяся частным случаем модели, предложенной Питтером Холмом и Бом Джун Кимом.

Данная модель вызывает интерес по той причине, что она может быть усложнена, что дает простор для дальнейших исследований. Можно предложить модель, в которой на каждой итерации добавляется не одна вершина, а другое константное случайно выбранное число, причем выбор распределения для этого числа может послужить для появления новой модели. Так же можно поступить по отношению к добавляемому числу ребер.

В ходе работы была проведена реализация на языке Python модели построения случайного графа с использованием матрицы смежности и с использованием библиотеки `networkx` для работы с графовыми структурами. Был сделан вывод, что использование матрицы смежности является более требовательным к памяти компьютера, на котором исполняется программа и приводит к увеличению времени генерации графа.

Полученные эмпирические результаты подтверждают теоретические выводы, касающиеся модели построения графа, полученные в [2].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Колчин, В. Ф.* Случайные графы / В. Ф. Колчин. — Спб: Физматлит, 2004.

2 *Holme, P.* Growing scale-free networks with tunable clustering / P. Holme, B. J. Kim // *Phys. Rev. E*. — Jan 2002. — Vol. 65. — P. 026107. https://link.aps.org/doi/10.1103/PhysRevE.65.026107.

3 *Bianconi, G.* Triadic closure as a basic generating mechanism of communities in complex networks / G. Bianconi, R. Darst, J. Iacovacci, S. Fortunato. — Vol. 65.: Phys. Rev. E., 2014.

4 The local closure coefficient: A new perspective on network clustering. [Электронный ресурс]. — URL: https://doi.org/10.1145/3289600.3290991. (Дата обращения 13.05.2020). Загл. с экр. Яз. англ.

5 *Райгородский, А. М.* Экстремальные задачи теории графов и анализ данных / А. М. Райгородский. — Мск: Регулярная и хаотичная динамика, 2009.

6 *Barabashi, L. A.* Emergence of scaling in random networks / L. A. Barabashi, R. Albert. — Cambridge: Science, 1999.

7 *Barabashi, L. A.* Scale-free characteristics of random networks: the topology of the world-wide web / L. A. Barabashi, R. Albert, H. Jeong. — Cambridge: Physica, 2000.

8 *Маргулис, Г. А.* Вероятностные характеристики графов с большой связностью / Г. А. Маргулис. — Мск: Проблемы передачи информации, 1974.

9 *Саммерфилд, М.* Программирование на Python 3. Подробное руководство / М. Саммерфилд. — Спб: Символ-Плюс, 2009.

10 Learnpython [Электронный ресурс]. — URL: https://www.learnpython.org/en/Modules_and_Packages (Дата обращения 22.05.2020). Загл. с экр. Яз. англ.

11 Researchgate [Электронный ресурс]. — URL: https://www.researchgate.net/publication/11497687_Growing_Scale-Free_Networks_with_Tunable_Clustering (Дата обращения 20.05.2020). Загл. с экр. Яз. англ.