Сетевое программирование

Лекция 4

Загрузка и генерация сетей. Генерация случайных графов

Усіпов Н.М., PhD · КазНУ имени аль-Фараби

Цель и план лекции

Цель

Освоение способов загрузки сетевых данных из файлов, изучение детерминированных и случайных генераторов графов, а также понимание свойств классических моделей случайных сетей.

1. Введение
2. Загрузка и сохранение сетей
3. Подготовка сетевых данных
4. Детерминированные генераторы
5. Модели случайных графов
6. Модели с учётом геометрии и заданной структуры
7. Воспроизводимость экспериментов
8. Сетевые наборы данных и контроль качества

1 Введение

- Исследователь и инженер работают с сетевыми данными двух типов. В первом случае топология реальной сети уже известна и хранится в файле, базе данных или системе управления.
- Случайные модели имеют также и самостоятельное научное значение: сравнение реальной сети со случайной моделью позволяет понять, какие её свойства являются закономерными, а какие могли возникнуть случайно.

2 Загрузка и сохранение сетей

NetworkX поддерживает основные форматы представления графов:

Список рёбер (edge list)

Текстовый файл, где каждая строка описывает одно ребро и, возможно, его атрибуты.

GraphML

Формат на основе XML с типизированными атрибутами, удобный для обмена данными с другими программами.

GML

Текстовый формат с иерархической структурой. Функции: `read_gml`, `write_gml`.

Таблицы pandas

Функция `from_pandas_edgelist` строит граф из таблицы, где столбцы содержат начало, конец и параметры канала.

При загрузке текстовых файлов следует помнить, что идентификаторы узлов по умолчанию считываются как строки.

2 Загрузка и сохранение сетей (продолжение)

Матрицы

Функции `from_numpy_array` и `to_numpy_array` выполняют преобразование из матрицы смежности и обратно.

При загрузке текстовых файлов следует помнить, что идентификаторы узлов по умолчанию считываются как строки.

3 Подготовка сетевых данных

- Реальные данные о сети часто содержат неточности: одно устройство может быть указано под разными именами, канал может быть описан дважды, встречаются отсутствующие значения.
- Если пропускная способность задана в разных единицах, например в Мбит/с и Гбит/с, все значения следует привести к одной единице.

3 Загрузка сети из таблицы и списка рёбер

Листинг 4.1.

```
import io
import networkx as nx
import pandas as pd

text = """R1,R2,10000,2
R2,R3,1000,5
R3,R1,10000,3
R3,R3,100,0"""
df = pd.read_csv(io.StringIO(text),
                 names=["src", "dst", "bw", "delay"])

G = nx.from_pandas_edgelist(df, "src", "dst", ["bw", "delay"])
print("Загружено рёбер:", G.number_of_edges())
print("Петли:", list(nx.selfloop_edges(G)))
G.remove_edges_from(list(nx.selfloop_edges(G)))
print("После очистки:", G.number_of_edges())

print("\n".join(nx.generate_edgelist(G, data=["bw"])))
```

3 Результат: Загрузка сети из таблицы и списка рёбер

```
Загружено рёбер: 4
Петли: [('R3', 'R3')]
После очистки: 3
R1 R2 10000
R1 R3 10000
R2 R3 1000
```

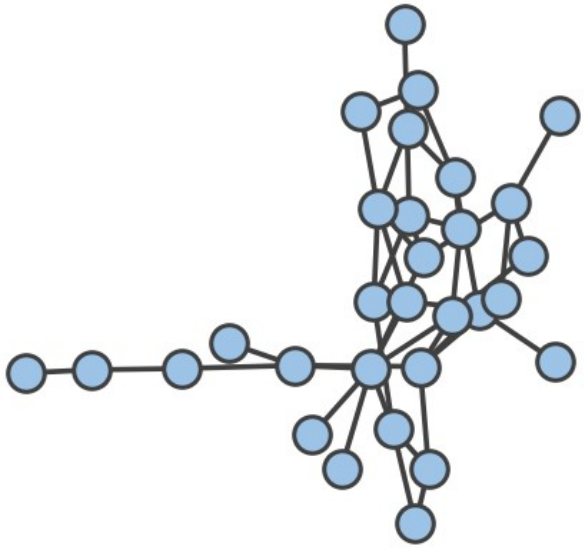
Как видно, запись о ребре R3–R3 была распознана как петля и удалена, поскольку не описывает реального канала между устройствами.

4 Детерминированные генераторы

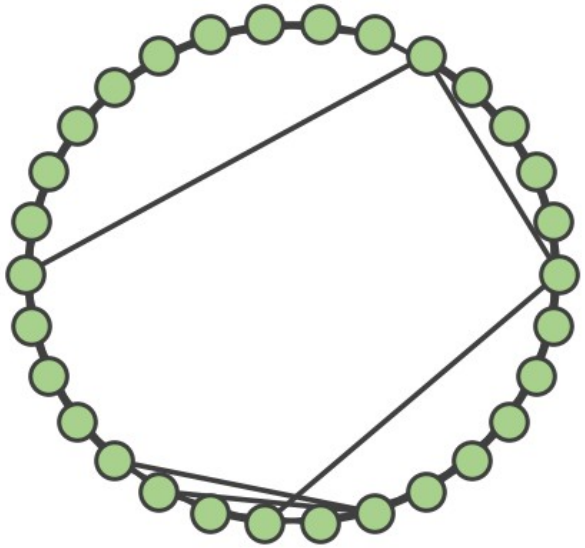
- Функции `complete_graph`, `path_graph`, `cycle_graph`, `star_graph`, `grid_2d_graph` и `balanced_tree` создают графы с заранее известной структурой.

5 Модели случайных графов

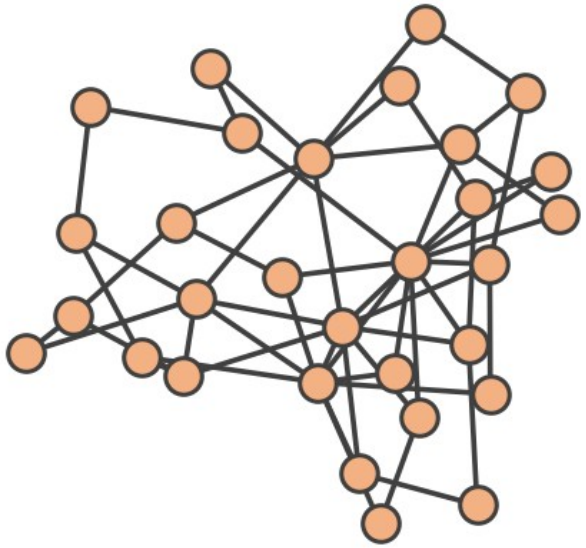
Эрдёш–Реньи



Уоттс–Строгац



Барабаши–Альберт



Примеры сетей, построенных по случайным моделям

5 Модели случайных графов (продолжение)

Модель Эрдёша–Реньи

В варианте G(n, p) каждая из n(n−1)/2 возможных пар вершин соединяется ребром независимо с вероятностью p.

Модель Уоттса–Строгаца

Построение начинается с кольцевой решётки, где каждый узел соединён с k ближайшими соседями.

Модель Барабаши–Альберт

Сеть растёт постепенно: каждый новый узел присоединяется к m существующим, причём вероятность выбрать узел пропорциональна его степени.

5 Модели случайных графов (продолжение)

- Выбор модели определяется изучаемым свойством. Следует также понимать ограничения: ни одна из этих моделей не учитывает географию, стоимость линий или техническую ёмкость маршрутизаторов, поэтому они не могут служить точным описанием реальной магистральной сети и используются прежде всего как базовые модели для сравнения.

6 Модели с учётом геометрии и заданной структуры

Для приближения модели к реальным сетям используются и другие генераторы.

Случайный геометрический граф

Узлы размещаются случайным образом в единичном квадрате, а ребро образуется, если расстояние между узлами не превышает заданного радиуса.

Модель Ваксмана

Вероятность появления ребра экспоненциально убывает с расстоянием между узлами, что отражает более высокую стоимость длинных линий.

Конфигурационная модель

Строит случайный граф с заданной последовательностью степеней. Она позволяет выяснить, какие свойства реальной сети объясняются только её распределением степеней.

7 Воспроизводимость экспериментов

- Все случайные генераторы NetworkX принимают параметр `seed`. Фиксация его значения обеспечивает получение одного и того же графа при повторном запуске, что необходимо при отладке и подготовке отчётов.

7 Генерация и сравнение случайных сетей

Листинг 4.2.

```
import networkx as nx

n = 1000
models = {
    "ER": nx.gnp_random_graph(n, 6 / (n - 1), seed=42),
    "WS": nx.watts_strogatz_graph(n, 6, 0.1, seed=42),
    "BA": nx.barabasi_albert_graph(n, 3, seed=42),
}

print("Модель  Рёбер  Ср.степ.  Макс.  Кластеризация")
for name, G in models.items():
    deg = [d for _, d in G.degree()]
    print(f"{name:6} {G.number_of_edges():6} "
          f"{sum(deg) / n:8.2f} {max(deg):6} "
          f"{nx.average_clustering(G):12.3f}")
```

7 Результат: Генерация и сравнение случайных сетей

Модель	Рёбер	Ср.степ.	Макс.	Кластеризация
ER	3019	6.04	15	0.009
WS	3000	6.00	9	0.452
BA	2991	5.98	93	0.033

Результаты подтверждают теоретические свойства: при близких средних степенях модель Уоттса–Строгаца даёт наибольшую кластеризацию, а модель Барабаши–Альберт формирует узлы с наибольшей степенью.

8 Сетевые наборы данных и контроль качества

- Для исследования применяются общедоступные коллекции сетевых данных. Internet Topology Zoo содержит карты реальных операторских сетей, а Stanford Large Network Dataset Collection (SNAP) – большие графы различного происхождения.
- Измерить топологию Интернета напрямую сложно. Инструменты наподобие traceroute позволяют увидеть только последовательность ответивших маршрутизаторов, а при выборке путей от небольшого числа источников возникает смещение: часть связей остаётся незамеченной, а степени узлов оцениваются неточно.
- В процессе подготовки к анализу необходимо определить следующее:

8 Сетевые наборы данных и контроль качества (продолжение)

Назначение и источник данных

Важно знать, кто и зачем их собрал и к какому моменту времени они относятся.

Уровень модели

Следует определить, идёт ли речь об устройствах, узлах связи, IP-подсетях или автономных системах.

Единицы атрибутов

Необходимо проверить и привести к единому виду единицы полосы и задержки.

Полнота

Следует выяснить, как обрабатываются отсутствующие значения.

Связность

Отдельные малые компоненты могут быть ошибками данных.

9 Степенные последовательности и нулевые модели

- Последовательность степеней не всегда может быть реализована простым графом.

$$\sum_{i \le k} d_i \le k(k-1) + \sum_{i>k} \min(d_i, k).$$

- В NetworkX проверка выполняется функцией `is_graphical`.
- Нулевая модель – это случайная сеть, сохраняющая определённые свойства исходной сети.

10 Фазовый переход в случайном графе

- Важнейшим результатом теории случайных графов является фазовый переход. При средней степени ⟨k⟩ < 1 все компоненты малы, их размер порядка ln n.

$$S = 1 - \exp(-\langle k \rangle S).$$

- При ⟨k⟩ = 2 решение равно приблизительно 0,80, при ⟨k⟩ = 4 — приблизительно 0,98.

11 Разбор примера

1. Рассмотрим модель $G(n, p)$ при $n = 1000$ и $p = 0{,}006$.
2. Математическое ожидание средней степени равно $(n-1)p = 5{,}994 \approx 6$.
3. Ожидаемое число рёбер составляет $n(n-1)/2 \cdot p = 499500 \cdot 0{,}006 = 2997$.
4. Порог появления гигантской компоненты $1/n = 0{,}001$ значительно превышен.
5. Порог связности $\ln(n)/n \approx 0{,}0069$ несколько больше заданного значения p, поэтому полная связность не гарантируется.
6. Ожидаемое число изолированных вершин приближённо равно $n \cdot \exp(-\langle k \rangle) = 1000 \cdot \exp(-6) \approx 2{,}5$.
7. Следовательно, в такой сети следует ожидать несколько изолированных узлов и одну очень большую компоненту.

12 Вопросы для обсуждения на занятии

Выбор модели

Какую модель целесообразно применить для сенсорной сети, а какую – для магистрали оператора?

Данные

Какие искажения могут возникнуть, если топология определяется только по результатам traceroute?

Нулевая модель

Какое свойство сети необходимо сохранить при проверке значимости её кластеризации?

Контрольные вопросы

1. Какие форматы хранения графов поддерживает NetworkX?
2. Какие проверки необходимо выполнить после загрузки сети из файла?
3. В чём заключается модель случайного графа Эрдёша–Реньи G(n, p)?
4. При каких условиях в случайном графе возникает гигантская компонента?
5. Что такое эффект «тесного мира» и как он воспроизводится в модели Уоттса–Строгаца?
6. Опишите механизм предпочтительного присоединения.
7. Для чего используется параметр seed? Почему нельзя ограничиваться одной реализацией графа?
8. Каковы ограничения случайных моделей при описании реальных телекоммуникационных сетей?

Список литературы

1. Erdős P., Rényi A. On random graphs I // Publicationes Mathematicae. – 1959. – Vol. 6. – P. 290–297.
2. Erdős P., Rényi A. On the evolution of random graphs // Publications of the Mathematical Institute of the Hungarian Academy of Sciences. – 1960. – Vol. 5. – P. 17–61.
3. Watts D.J., Strogatz S.H. Collective dynamics of ‘small-world’ networks // Nature. – 1998. – Vol. 393. – P. 440–442.
4. Barabási A.-L., Albert R. Emergence of scaling in random networks // Science. – 1999. – Vol. 286. – P. 509–512.
5. Waxman B.M. Routing of multipoint connections // IEEE Journal on Selected Areas in Communications. – 1988. – Vol. 6, No. 9. – P. 1617–1622.
6. Penrose M. Random Geometric Graphs. – Oxford: Oxford University Press, 2003.
7. Newman M. Networks. – 2nd ed. – Oxford: Oxford University Press, 2018. – 780 p.
8. NetworkX Documentation [Электронный ресурс]. – URL: https://networkx.org/documentation/stable/