Сетевое программирование

Лекция 11

Динамические и временные сети. Моделирование изменения структуры сети во времени. Добавление и удаление рёбер. Анализ эволюции сети

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

Цель и план лекции

Цель

Изучение способов представления сетей, изменяющихся во времени, понятия временного пути и получение навыков анализа эволюции структуры сети средствами Python и NetworkX.

1. Введение
2. Способы представления временных сетей
3. Моделирование изменения структуры
4. Временные пути
5. Анализ эволюции сети
6. Модели роста и эволюции сетей
7. Анализ изменений структуры

1 Введение

- До сих пор сеть рассматривалась как неизменный объект. Реальные сети связи постоянно изменяются: устанавливается новое оборудование, вводятся и выводятся из эксплуатации каналы, абоненты подключаются и отключаются, а мобильные устройства перемещаются между зонами обслуживания.
- Если не учитывать время, можно прийти к ошибочным выводам. Например, два канала могут существовать в разные моменты, и тогда статический граф покажет наличие пути, которого в действительности не было.
- Принято различать два вида процессов. Динамика самой сети (dynamics of network) – это изменение узлов и рёбер.

2 Способы представления временных сетей

Последовательность снимков

Для моментов t_1, t_2, ..., t_T хранится отдельный статический граф. Этот способ прост и позволяет применять все обычные алгоритмы, однако точность зависит от выбранного шага, а объём памяти увеличивается пропорционально числу снимков.

Последовательность контактов

Сеть описывается набором записей (u, v, t), означающих, что в момент t существовало мгновенное взаимодействие.

Интервальный граф

Для каждого ребра указываются интервалы существования (u, v, $t_{нач}$, $t_{кон}$). Этот способ удобен для каналов, которые работают продолжительное время.

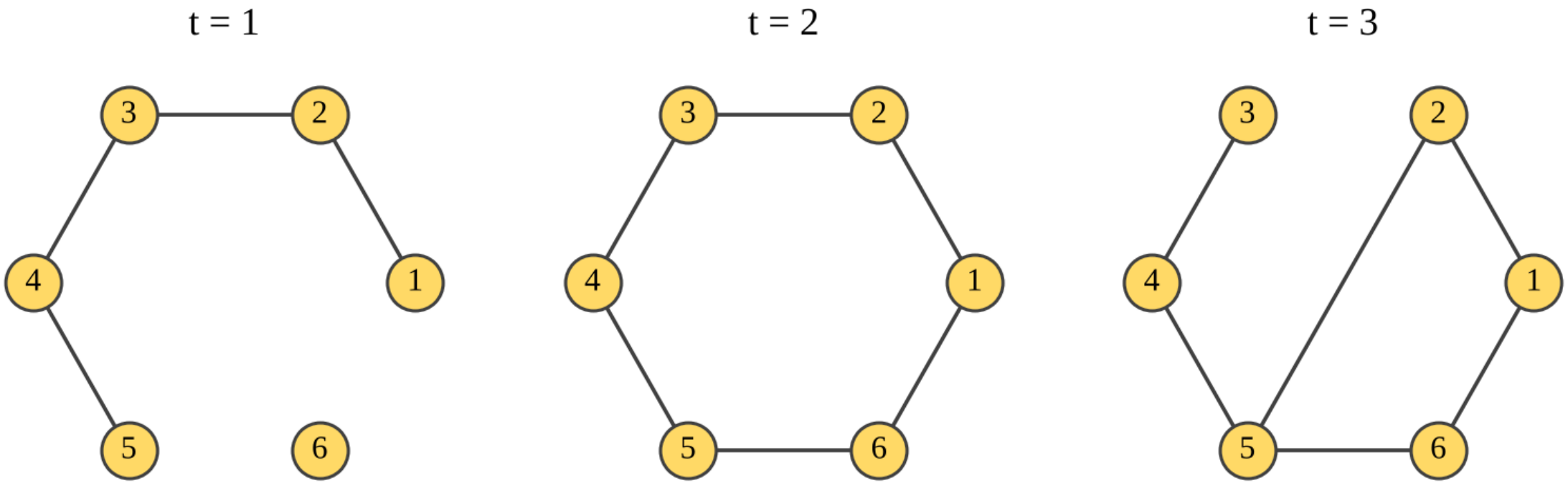
Агрегированный граф

Все ребра за период объединяются в один статический граф, а вес отражает частоту или длительность связи.

2 Способы представления временных сетей (продолжение)

- Специального класса для временных сетей в NetworkX нет. Обычно применяют список снимков либо хранят время в атрибутах рёбер и формируют нужные подграфы.
- В момент t = 2 сеть образует кольцо, а к моменту t = 3 канал 2–3 исчезает и появляется новый канал 2–5.

2 Способы представления временных сетей (продолжение)



Снимки сети в последовательные моменты времени

3 Моделирование изменения структуры

Изменение сети моделируется с помощью последовательного применения событий.

Добавление ребра

Ввод в эксплуатацию канала или установление радиосвязи (`add_edge`).

Удаление ребра

Отказ канала, плановые работы, выход абонента из зоны действия (`remove_edge`).

Добавление узла

Установка нового оборудования (`add_node`).

Удаление узла

Выход устройства из строя или вывод его из эксплуатации (`remove_node`).

Изменение атрибута

Изменение загрузки, задержки или доступной полосы канала без изменения топологии.

Интенсивность событий может определяться по статистике, законом распределения времени между отказами или расписанием работ.

3 Построение снимков сети по интервалам

Листинг 11.1.

```
import networkx as nx

links = [(1, 2, 1, 3), (2, 3, 1, 2), (3, 4, 1, 3), (4, 5, 1, 3),
         (5, 6, 2, 3), (6, 1, 2, 3), (2, 5, 3, 3)]

def snapshot(t):
    G = nx.Graph()
    G.add_nodes_from(range(1, 7))
    G.add_edges_from((u, v) for u, v, a, b in links if a <= t <= b)
    return G

snaps = {t: snapshot(t) for t in (1, 2, 3)}
for t, G in snaps.items():
    print(f"t={t}: рёбер {G.number_of_edges()}, "
          f"компонент {nx.number_connected_components(G)}")

def jaccard(A, B):
    a = {frozenset(e) for e in A.edges}
    b = {frozenset(e) for e in B.edges}
    return len(a & b) / len(a | b)

print("Сходство 1-2:", round(jaccard(snaps[1], snaps[2]), 2))
print("Сходство 2-3:", round(jaccard(snaps[2], snaps[3]), 2))
```

3 Результат: Построение снимков сети по интервалам

```
t=1: рёбер 4, компонент 2
t=2: рёбер 6, компонент 1
t=3: рёбер 6, компонент 1
Сходство 1-2: 0.67
Сходство 2-3: 0.71
```

4 Временные пути

Временной путь – это последовательность рёбер, в которой время использования каждого следующего ребра не меньше, чем время использования предыдущего.

Отсутствие симметрии

Если из A можно передать данные в B, это не означает, что возможна передача из B в A.

Отсутствие транзитивности

Наличие временных путей из A в B и из B в C не гарантирует наличия пути из A в C, поскольку второй из них мог завершиться раньше начала первого.

Различные критерии оптимальности

Самым коротким может быть путь с наименьшим числом переходов, с наиболее ранним прибытием или с наименьшей длительностью.

В сетях, устойчивых к задержкам (DTN), устройства сохраняют данные и передают их дальше при следующем контакте.

4 Самое раннее время прибытия

Листинг 11.2.

```
def earliest_arrival(contacts, source):
    arrive = {source: 0}
    for t, u, v in sorted(contacts):
        for a, b in ((u, v), (v, u)):
            if arrive.get(a, float("inf")) <= t < arrive.get(b, float("inf")):
                arrive[b] = t
    return arrive

contacts = [(1, "A", "B"), (2, "C", "D"), (3, "B", "C"), (4, "D", "E")]
print("Из A:", earliest_arrival(contacts, "A"))
print("Из E:", earliest_arrival(contacts, "E"))
```

```
Из A: {'A': 0, 'B': 1, 'C': 3}
Из E: {'E': 0, 'D': 4}
```

5 Анализ эволюции сети

- Для оценки изменений строят временные ряды показателей: числа узлов и рёбер, средней степени, диаметра, размера наибольшей компоненты.

- Практическое значение анализа эволюции связано с проблемой сходимости протоколов маршрутизации.

6 Модели роста и эволюции сетей

- Статистический анализ временных данных показывает, что реальные сети подчиняются определённым закономерностям изменения.
- Для моделирования роста используются следующие механизмы:

6 Модели роста и эволюции сетей (продолжение)

Случайное присоединение

Новый узел присоединяется к случайно выбранным существующим узлам.

Предпочтительное присоединение

Вероятность присоединения пропорциональна степени узла, что воспроизводит формирование концентраторов.

Предпочтение с учётом расстояния

Вероятность уменьшается с расстоянием, так как прокладка длинных линий обходится дорого.

Копирование связей

Новый узел копирует часть соединений случайно выбранного узла, формируя сходные группы.

Модели мобильности определяют изменение топологии беспроводных сетей. В простой модели random waypoint устройство выбирает случайную точку назначения, движется к ней с некоторой скоростью и затем делает паузу.

6 Модели роста и эволюции сетей (продолжение)

Активность узлов

В модели activity-driven узлы случайным образом становятся активными и создают кратковременные связи в соответствии с индивидуальной активностью.

Модели мобильности определяют изменение топологии беспроводных сетей. В простой модели random waypoint устройство выбирает случайную точку назначения, движется к ней с некоторой скоростью и затем делает паузу.

7 Анализ изменений структуры

Для временных сетей меры обобщаются с учётом хронологии.

Временная достижимость

Показывает, какая доля пар узлов соединена временным путём.

Временная близость и посредничество

Учитывают время прибытия по временным путям.

Время между контактами

Промежутки между контактами в реальных системах часто имеют распределение с тяжёлыми хвостами, из-за чего распространение информации может быть замедлено по сравнению с пуассоновским процессом.

Обнаружение изменений

Поиск моментов, когда сеть резко изменяет свои свойства. Для этого применяют статистические критерии к временным рядам метрик.

7 Анализ изменений структуры (продолжение)

Устойчивость сообществ

Анализ того, как сообщества рождаются, делятся, сливаются и исчезают.

7 Анализ изменений структуры (продолжение)

- Для корректного анализа важно проводить сравнение с рандомизированными данными.
- Учитываются и особенности работы сети. Планируемые работы можно формализовать в виде расписания недоступности каналов, а затем проверить, что сеть остаётся связной во всех интервалах времени.

8 Разбор примера

1. Вычислим сходство снимков, приведённых на рис. 11.1.
2. Снимок 1 содержит рёбра {1–2, 2–3, 3–4, 4–5}, снимок 2 содержит {1–2, 2–3, 3–4, 4–5, 5–6, 6–1}.
3. Пересечение содержит 4 ребра, объединение – 6, коэффициент Жаккара равен 4/6 ≈ 0,67.
4. Снимок 3 содержит рёбра {1–2, 3–4, 4–5, 5–6, 6–1, 2–5}.
5. Со снимком 2 он имеет 5 общих рёбер, а объединение содержит 7 рёбер, поэтому коэффициент равен 5/7 ≈ 0,71.
6. Рассмотрим контакты (t = 1, A–B), (t = 2, C–D), (t = 3, B–C), (t = 4, D–E).
7. Пакет из A попадает в B при t = 1, затем в C при t = 3.

8 Разбор примера (продолжение)

8 Контакт C–D произошёл при t = 2, раньше прибытия пакета, поэтому узлы D и E из A недостижимы, хотя статический граф связен.

9 Вопросы для обсуждения на занятии

Шаг снимков

Какой шаг вы выбрали бы для анализа сети сотовых абонентов и магистрали оператора?

Предсказуемость

В чём различие между маршрутизацией в сетях со случайной и заранее известной топологией?

Применение

Какие изменения топологии могут быть признаком неисправности?

Контрольные вопросы

1. Чем динамические сети отличаются от статических? Приведите примеры.
2. В чём состоит различие между динамикой сети и динамикой на сети?
3. Назовите основные способы представления временных сетей и их недостатки.
4. Как выбор шага дискретизации влияет на результат анализа?
5. Какие события используются при моделировании изменения топологии?
6. Дайте определение временного пути. Почему он не обладает свойством транзитивности?
7. Как определяется коэффициент Жаккара для двух снимков сети?
8. Как изменения топологии связаны со сходимостью протоколов маршрутизации?

Список литературы

1. Holme P., Saramäki J. Temporal networks // Physics Reports. – 2012. – Vol. 519, No. 3. – P. 97–125.
2. Kempe D., Kleinberg J., Kumar A. Connectivity and inference problems for temporal networks // Journal of Computer and System Sciences. – 2002. – Vol. 64, No. 4. – P. 820–842.
3. Fall K. A delay-tolerant network architecture for challenged internets // Proc. ACM SIGCOMM 2003. – Karlsruhe, 2003. – P. 27–34.
4. Newman M. Networks. – 2nd ed. – Oxford: Oxford University Press, 2018. – 780 p.
5. NetworkX Documentation [Электронный ресурс]. – URL: https://networkx.org/documentation/stable/