ИНТЕЛЛЕКТУАЛЬНОЕ МОДЕЛИРОВАНИЕ

Образовательная программа «6B06103 – Компьютерная инженерия»

Лекция 14

Поиск ассоциативных правил. Сравнительный анализ алгоритмов

Неделя 14 · Tan, Steinbach, Kumar: 6.1–6.7

Цель и план лекции

Цель: понять методы поиска ассоциативных правил, оценивать их полезность и обоснованно выбирать алгоритм

1. Транзакции, поддержка, достоверность
2. Принцип и алгоритм Apriori, ручной расчёт
3. Интересность правил: лифт
4. FP-Growth и Eclat
5. Сравнение алгоритмов, mlxtend

Анализ покупательской корзины

«Покупатели подгузников часто покупают и пиво»

- Торговля: перекрёстные продажи, выкладка, наборы, рекомендации
- Также: биоинформатика, диагностика, веб-логи, науки о Земле
- Проблемы: поиск дорог вычислительно; часть правил случайна или неинтересна

База транзакций (таблица 6.1)

№	Хлеб	Молоко	Подгузники	Пиво	Яйца	Кола
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

- Бинарный вид: товар — асимметричный бинарный признак
- **k-набор** — набор из k товаров; $\sigma(X)$ — число транзакций с X

Правило, поддержка, достоверность

$$s(X \to Y) = \frac{\sigma(X \cup Y)}{N}$$

$$c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Пример 6.1: {Молоко, Подгузники} → {Пиво}

0,4

поддержка $= \frac{2}{5}$

0,67

достоверность $= \frac{2}{3}$

Правило — совместная встречаемость, а не причинно-следственная связь

Постановка задачи

Найти все правила с поддержкой ≥ minsup и достоверностью ≥ minconf

Всего правил: $R = 3^d - 2^{d+1} + 1$ → для 6 товаров уже 602

Этап 1

Поиск частых наборов (самый дорогой)

Этап 2

Генерация сильных правил

Принцип Apriori

Если набор редкий, то все его надмножества тоже редкие

- Эквивалентно: подмножества частого набора частые
- {a, b} редкий → не проверяем {a, b, c}, {a, b, d}, …
- Основа — **антимонотонность** поддержки
- Решётка из $2^k - 1$ наборов резко сокращается

Алгоритм Apriori (Agrawal, Srikant, 1994)

1. Проход по данным: поддержка товаров → F_1
2. Кандидаты C_k из F_{k-1}; отсечь имеющих редкое подмножество
3. Проход по данным: подсчёт поддержки C_k
4. Частые кандидаты → F_k
5. Стоп, когда F_k пусто

По уровням, «породить и проверить»: проходов = размер наибольшего частого набора + 1

Пример: minsup = 60 % (счётчик ≥ 3)

Шаг 1: 1-наборы

Набор	σ	Частый
{Пиво}	3	Да
{Хлеб}	4	Да
{Кола}	2	Нет
{Подгузники}	4	Да
{Молоко}	4	Да
{Яйца}	1	Нет

Шаг 2: 6 пар вместо 15

Набор	σ	Частый
{Пиво, Хлеб}	2	Нет
{Пиво, Подгузники}	3	Да
{Пиво, Молоко}	2	Нет
{Хлеб, Подгузники}	3	Да
{Хлеб, Молоко}	3	Да
{Подгузники, Молоко}	3	Да

Шаг 3 и итог

- Единственный кандидат: {Хлеб, Подгузники, Молоко}
- Встречается в транзакциях 4 и 5: $\sigma = 2$, поддержка 40 % — **редкий**
- На рис. 6.5 учебника указан счётчик 3, но по таблице он равен 2
- Частые наборы: 4 товара и 4 пары

41

кандидат при полном переборе

13

кандидатов у Apriori — на 68 % меньше

Генерация правил (minconf = 70 %)

Правило	Поддержка	Достоверн.	Лифт
{Пиво} → {Подгузники}	0,6	1,00	1,25
{Подгузники} → {Пиво}	0,6	0,75	1,25
{Хлеб} ↔ {Подгузники}	0,6	0,75	0,94
{Хлеб} ↔ {Молоко}	0,6	0,75	0,94
{Подгузники} ↔ {Молоко}	0,6	0,75	0,94

- Из k-набора — $2^k - 2$ правил; поддержка уже известна
- Перенос элементов из условия в следствие не повышает достоверность
- ↔ — оба направления дают одинаковые значения

Ловушка достоверности: чай и кофе

	Пьют кофе	Не пьют	Всего
Пьют чай	150	50	200
Не пьют чай	650	150	800
Всего	800	200	1000

- {Чай} → {Кофе}: поддержка 15 %, достоверность 75 % — выглядит сильным
- Но кофе пьют 80 % всех, а среди любителей чая — только 75 %

Достоверность игнорирует частоту следствия

Лифт

$$\text{Lift} = \frac{c(X \to Y)}{s(Y)} = \frac{s(X \cup Y)}{s(X) \cdot s(Y)}$$

= 1

независимы

> 1

положительная связь

< 1

отрицательная связь

- Чай → кофе: $\frac{0{,}75}{0{,}8} = 0{,}9375$; пиво → подгузники: 1,25
- Другие меры: leverage, conviction; осторожно — парадокс Симпсона

Компактное представление и недостатки Apriori

Максимальные наборы

Нет частых надмножеств; восстанавливают наборы, но не поддержку

Замкнутые наборы

Нет надмножеств с той же поддержкой; восстанавливают всё

- Apriori: много проходов по данным, медленен на плотных данных
- Альтернативы различаются обходом решётки (в ширину / в глубину) и представлением данных (горизонтальное / вертикальное)

FP-Growth (Han и соавт., 2000)

1

Проход 1: поддержка товаров, упорядочить частые по убыванию

2

Проход 2: построить FP-дерево; общие начала — общий путь

3

Извлечь наборы «разделяй и властвуй» по условным деревьям

Два прохода, без кандидатов. Дерево мало, если у транзакций много общего

Eclat (Zaki, 2000)

Вертикальное представление (TID-списки) и обход в глубину

```
TID(Пиво)       = {2, 3, 4}
TID(Подгузники) = {2, 3, 4, 5}
Пересечение     = {2, 3, 4} → σ = 3
```

- Поддержка без новых проходов по данным
- Чем больше набор, тем короче списки
- Недостаток: TID-списки должны поместиться в память

Сравнительный анализ алгоритмов

Критерий	Apriori	FP-Growth	Eclat
Данные	Горизонтальные	FP-дерево	TID-списки
Поиск	В ширину	В глубину	В глубину
Кандидаты	Да	Нет	Да, без проходов
Проходов	k + 1	2	1
Память	Мало, но много кандидатов	Дерево в памяти	Списки в памяти
Лучше для	Разреженных, высокий minsup	Плотных, низкий minsup	Средних объёмов

Все три находят одно и то же — различаются скорость и память

Ассоциативные правила в mlxtend

```
te = TransactionEncoder()
basket = pd.DataFrame(te.fit_transform(transactions),
                      columns=te.columns_)
frequent = apriori(basket, min_support=0.6,
                   use_colnames=True)  # 4 товара, 4 пары
rules = association_rules(frequent, metric='confidence',
                          min_threshold=0.7)
```

При minsup = 40 % (fpgrowth): 17 наборов; {Молоко, Подгузники} → {Пиво}: 0,4 / 0,667 / лифт 1,11

Скорость: 10 000 транзакций, 100 товаров

Алгоритм	minsup = 5 %	minsup = 2 %
Частых наборов	727	3172
Apriori, с	≈ 1,4	≈ 9
FP-Growth, с	≈ 1,1	≈ 3

При низком пороге наборов в 4 раза больше, и FP-Growth примерно втрое быстрее

Выводы

- Сила правила — поддержка и достоверность, полезность — лифт
- Два этапа: частые наборы, затем правила
- Принцип Apriori отсекает большую часть поиска
- FP-Growth и Eclat — быстрее на плотных данных
- Выбор алгоритма — по размеру, плотности и minsup

Далее: итоговый контроль (Endterm)

Контрольные вопросы

1. Набор, σ, правило, поддержка, достоверность; почему правило не означает причинности?
2. Зачем два этапа? Число возможных правил для 4 товаров.
3. Принцип Apriori и антимонотонность поддержки.
4. Apriori для {A, B, C}, {A, B}, {A, C}, {B, C}, {A, B, C, D} при $\sigma \geq 3$.
5. Пример «чай — кофе»; формула лифта и его значения.
6. Сравните Apriori, FP-Growth, Eclat; программа на mlxtend с правилами lift > 1.

Литература

1. Tan P.-N., Steinbach M., Kumar V. Introduction to Data Mining. — Pearson, 2014. — Chapter 6: 6.1–6.7.
2. Agrawal R., Srikant R. Fast Algorithms for Mining Association Rules. — VLDB, 1994.
3. Han J., Pei J., Yin Y. Mining Frequent Patterns without Candidate Generation. — SIGMOD, 2000.
4. Zaki M. J. Scalable Algorithms for Association Mining // IEEE TKDE. — 2000.
5. Документация mlxtend: Frequent itemsets and association rules.