大数据分析与应用

2020春

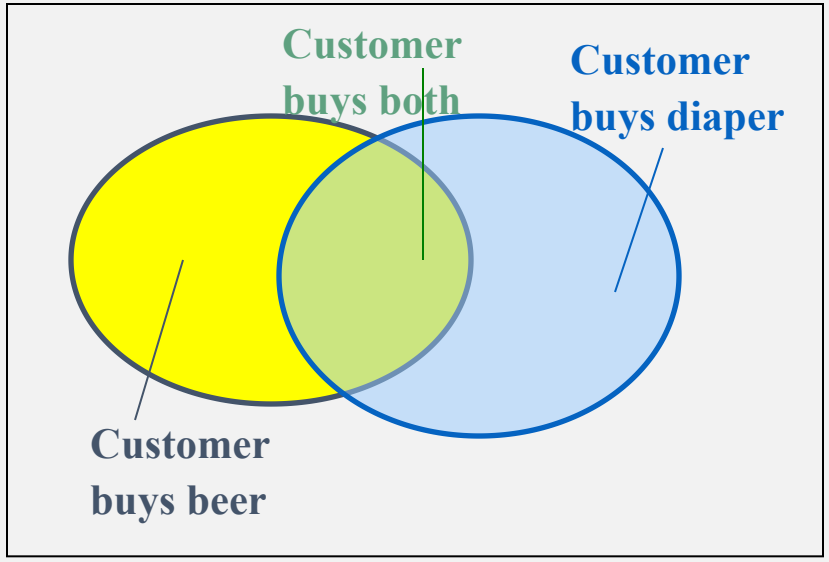
第3章 关联分析/频繁模式挖掘

■最原始的推荐系统

- 数据中频繁出现的模式：
  - 什么商品经常被一起购买?— Beer and diapers?!
  - 购买PC后消费者经常还会买什么?
  - 网站用户的行为模式分析——点击行为的模式
  - DNA片段的关联：通常同时出现的序列组合
- 应用
  - 篮子数据分析, 交叉市场营销, 分类设计, 销售分析, WEB记录流量分析, and DNA 序列分析.

3.1 Apriori 基本概念：频繁模式与支持度

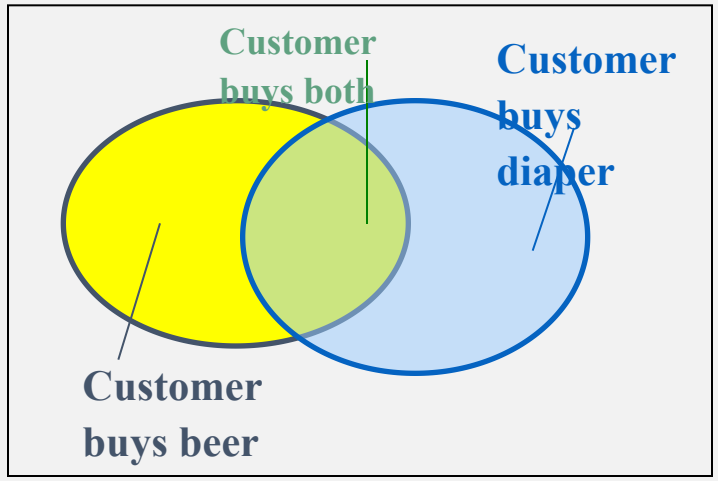
Tid	Items bought
1	Beer, Nuts, Diaper
2	Beer, Coffee, Diaper
3	Beer, Diaper, Eggs
4	Nuts, Eggs, Milk
5	Nuts, Coffee, Diaper, Eggs, Milk



- *项集*: 物品的集合
- *k-项集* $X = \{x_1, \ldots, x_k\}$
- *支持度 support count* of X: 项集X出现的次数
- *相对支持度*, 项集X 出现的频率
- 问题形式化：寻找支持度大于 *minsup* 最小阈值的项集

3.1 Apriori 基本概念：推荐准则

Tid	Items bought
1	Beer, Nuts, Diaper
2	Beer, Coffee, Diaper
3	Beer, Diaper, Eggs
4	Nuts, Eggs, Milk
5	Nuts, Coffee, Diaper, Eggs, Milk

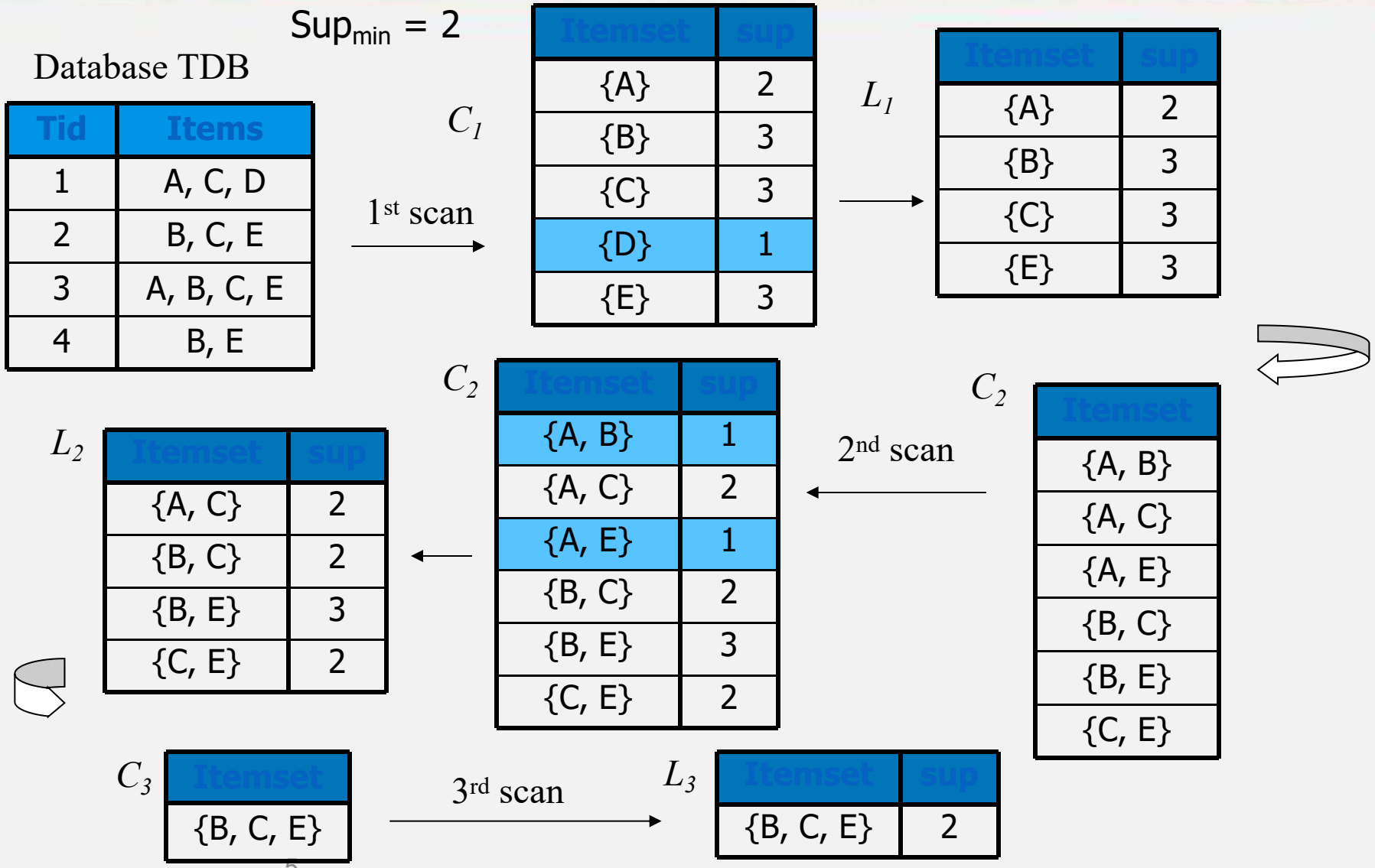


- 由项集得到规则（推荐准则）
  - rules $X \rightarrow Y$
  - 推荐准则的相对支持度，
  - 推荐准则的置信度，

举例：5条记录，商品出现次数： Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules关联规则
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

3.1 Apriori 一个例子



3.1 Apriori： k+1项集的构造和剪枝

- Assume the items in L_k are listed in an order (e.g., alphabetical)
- **Step 1:** ***self-joining*** L_k *(IN SQL)*

  insert into C_{k+1}

  select $p.item_1, p.item_2, \ldots, p.item_k, q.item_k$

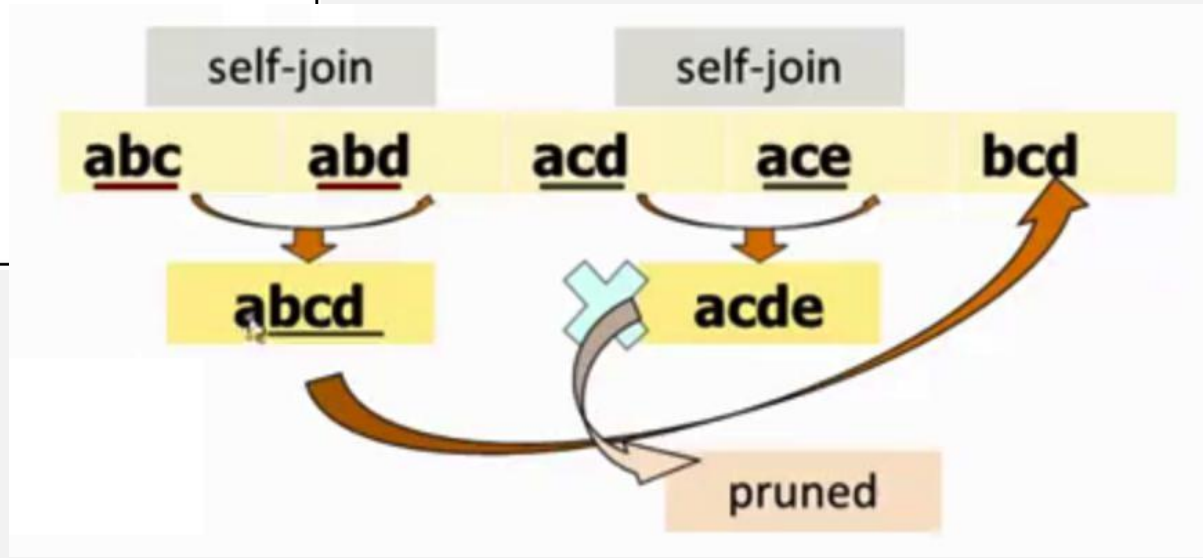
  from $L_k\ p, L_k\ q$

  where $p.item_1=q.item_1, \ldots, p.item_{k-1}=q.item_{k-1}, p.item_k < q.item_k$
- **Step 2:** ***pruning***

  forall ***itemsets c in*** C_{k+1} do

  forall ***k-subsets s of c*** do

  if *(s is* ***not*** *in* L_k*)* **then delete** *c* **from** C_{k+1}



3.1 Apriori ： 速度问题

□加速方法

- 事务压缩：不包含频繁k项集的事务，一定不包含频繁k+1项集；
- 分区：将数据集划分成小数据集（列），分别统计局部频繁项集，最终合并；
- 采样：牺牲一部分精度，行的方向让数据更小；

□缺点

- 仍然有大量候选项集
- 多次扫描数据库

3.2 FP-Grows：频繁模式树

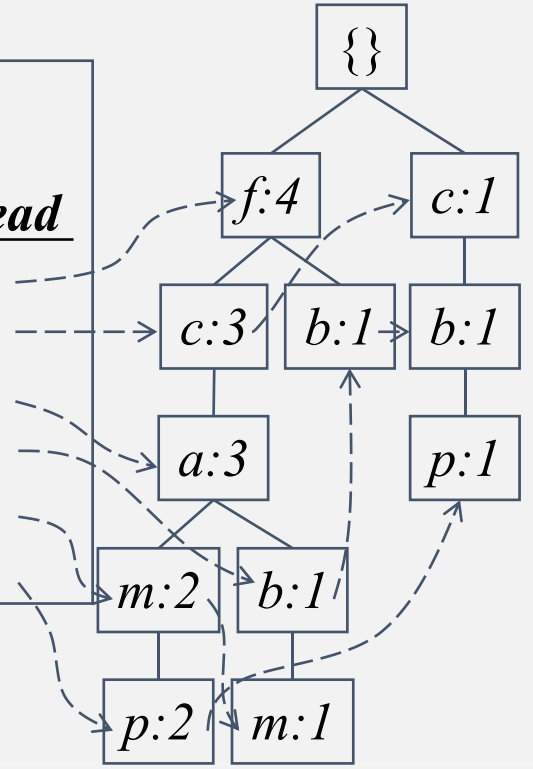
TID	Items bought	(ordered) frequent items
1	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
2	{a, b, c, f, l, m, o}	{f, c, a, b, m}
3	{b, f, h, j, o, w}	{f, b}
4	{b, c, k, s, p}	{c, b, p}
5	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

1. 扫描一次数据库找到1频繁项集
2. 按频繁程度排序
3. 再次扫描数据，构造频繁模式树

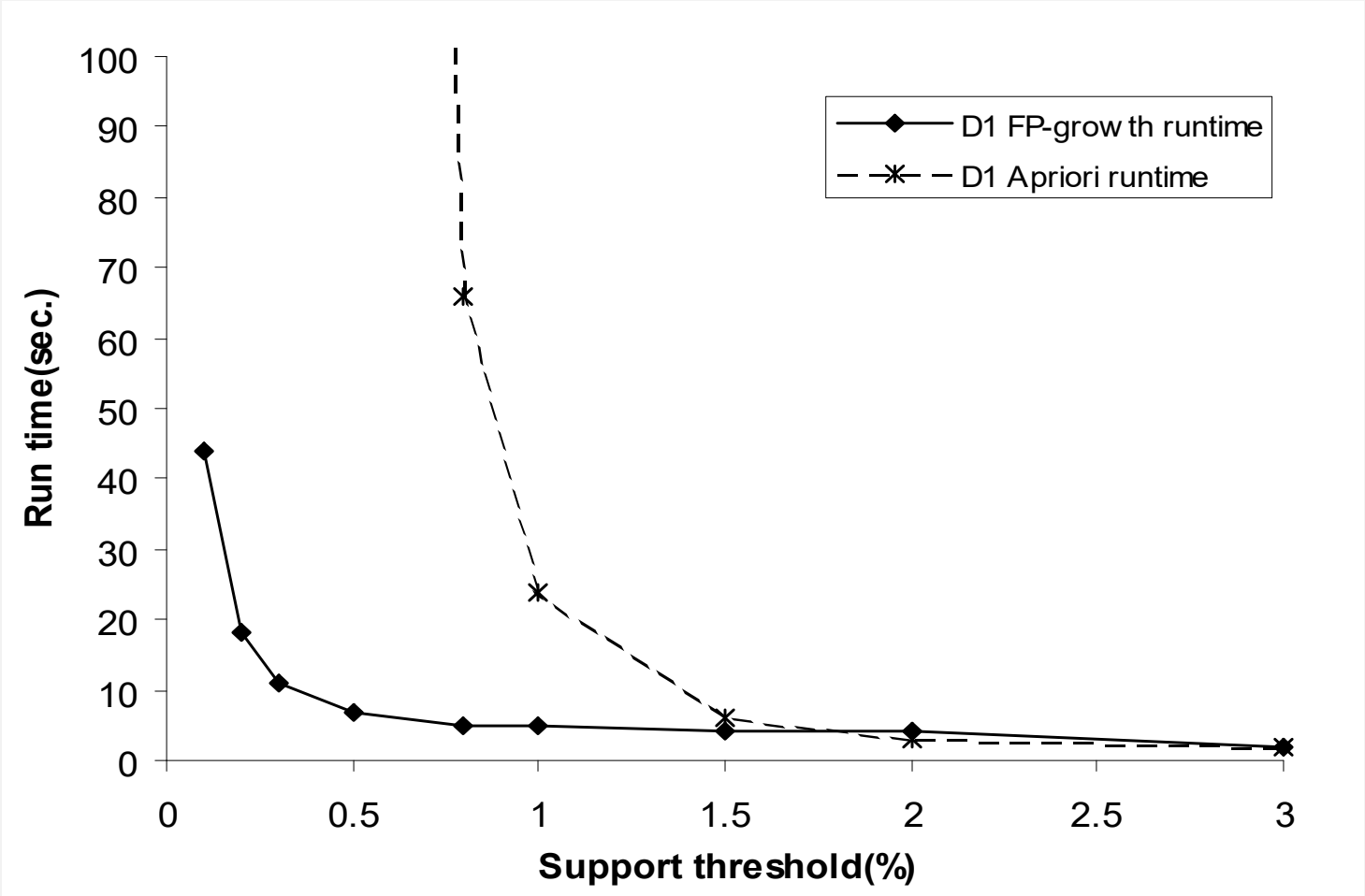
Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



F-list = f-c-a-b-m-p

3.2 FP-Grows ： 与Apriori方法比较



FP-Growth vs. Apriori

3.3 等价类变换方法

商品	交易号
A	{T1，T4，T5，T7，T8}
B	{T1，T2，T3，T4, T6, T8，T9}
C	{T3，T5，T6，T7, T8, T9}
D	{T2，T4}
E	{T1，T8}

{A,B} suport=3
{A,B,C} suport=1

3.4 结果的理解

- 高置信度高支持度一定是有意义的关联规则吗？
- 举例：10000次交易记录，6000次包含计算机设备，7500次包含游戏软件，4000次同时包含计算机设备与游戏软件，那么“购买计算机--->购买游戏软件”规则的支持度是40%，置信度是66%。

	购买计算机	没有购买计算机
购买游戏	4000	3500
没有购买游戏	2000	500

3.4 结果的理解：关联规则的评价准则

- Lift：提升

$$lift(A,B) = P(A \cup B)/(P(A)P(B))$$

3.4 结果的理解：与列相似性计算的关系

Tid	Items bought
1	Beer, Nuts, Diaper
2	Beer, Coffee, Diaper
3	Beer, Diaper, Eggs
4	Nuts, Eggs, Milk
5	Nuts, Coffee, Diaper, Eggs, Milk

- ✓ 列的相似性计算可以用来做关联规则分析
- ✓ 更多的时候：关联规则方法作为一种特征相似性挖掘方法存在

	Beer	Nuts	Diaper	Coffee	Eggs	Milk
1	1	1	1	0	0	0
2	1	0	1	1	0	0
3	1		1	0	1	0
4	0	1	0	0	1	1
5	0	1	1	1	1	1

关联规则挖掘（平时小）实验

- 1.运行代码
- 2.阅读代码与Apriori算法对应，将伪代码描述对应到代码，写清注释；
- 3.更换较大的数据集income.csv,

✓以最小支持度为0.1，最小置信度为0.5建立Apriori关联规则

✓以最小支持度为0.1，最小置信度为0.6建立Apriori关联规则

✓以最小支持度为0.2，最小置信度为0.5建立Apriori关联规则

- 比较三个关联规则的数目。