Christian-Albrechts-Universität zu Kiel Technische Fakultät

Inf-KDDM: Knowledge Discovery and Data Mining

Winter Term 2020/21

Lecture 3: Frequent Itemsets and Association Rule Mining

Lectures: Prof. Dr. Matthias Renz

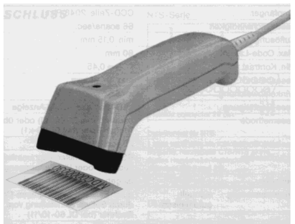
Exercises: Steffen Strohm

Outline

- Introduction
- Basic concepts
- Frequent Itemsets Mining (FIM) – Apriori
- Association Rules Mining

Introduction

- Frequent patterns are patterns that appear frequently in a dataset.
  - Patterns: items, substructures, subsequences …
- Typical example: Market basket analysis



transactions

items

Customer transactions

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

- We want to know: What products were often purchased together?
  - e.g.: beer and diapers?

The parable of the beer and diapers:
http://www.theregister.co.uk/2006/08/15/beer_diapers/

- Applications:
  - Improving store layout, Sales campaigns, Cross-marketing, Advertising

Applications beyond marked basket data

- Market basket analysis
  - Items are the products, transactions are the products bought by a customer during a supermarket visit
  - Example: *{"Diapers"} → {"Beer} (0.5%, 60%)*
- Similarly in an online shop, e.g. Amazon
  - Example: *{"Computer"} → {"MS office"} (50%, 80%)*
- University library
  - Items are the books, transactions are the books borrowed by a student during the semester
  - Example: *{"Kumar book"} →{"Weka book"} (60%, 70%)*
- University
  - Items are the courses, transactions are the courses that are chosen by a student
  - Example: *{"CS"} ∧ {"DB"} →{"Grade A"} (1%, 75%)*
- … and many other applications.
- Also, frequent patter mining is fundamental in other DM tasks.

Outline

- Introduction
- Basic concepts
- Frequent Itemsets Mining (FIM) – Apriori
- Association Rules Mining
- Homework
- Things you should know from this lecture

Basic concepts: Items, itemsets and transactions 1/2

- Items I: the set of items $I = \{i_1, ..., i_m\}$
  - e.g. products in a supermarket, books in a bookstore
- Itemset X: A set of items $X \subseteq I$
- Itemset size: the number of items in the itemset
- k-Itemset: an itemset of size k
  - e.g. {Butter, Bread, Milk, Sugar} is a 4-Itemset, {Butter, Bread} is a 2-Itemset
- Transaction T: $T = (tid, X_T)$
  - e.g. products bought during a customer visit to the supermarket
- *Database DB*: A set of transactions T
  - e.g. customers purchases in a supermarket during the last week
- Items in transactions or itemsets are lexicographically ordered
  - Itemset $X = (x_1, x_2, ..., x_k)$, such as $x_1 \leq x_2 \leq ... \leq x_k$

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

Basic concepts: Items, itemsets and transactions 2/2

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

Let X be an itemset.

- Itemset cover: the set of transactions containing *X*:

  $cover(X) = \{tid \mid (tid, X_T) \in DB, X \subseteq X_T\}$

- (absolute) Support/ support count of *X*: # transactions containing *X*

  $supportCount(X) = |cover(X)|$

- (relative) Support of *X*: fraction of transactions containing *X* (or the probability that a transaction contains *X*)

  $support(X) = P(X) = supportCount(X) / |DB|$

- Frequent itemset: An itemset *X* is frequent in DB if its support is no less than a *minSupport* threshold *s*:

  $support(X) \geq s$

- L_k: the set of frequent *k*-itemsets
  - *L* comes from "Large" ("large itemsets"), another term for "frequent itemsets"

Example: Itemsets

- I = {Butter, Bread, Eggs, Flour, Milk, Salt, Sugar}

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

- support(Butter) = 4/5=80%
  - cover(Butter) = {1,2,3,4}
- support(Butter, Bread) = 1/5=20%
  - cover(Butter, Bread) = ….
- support(Butter, Flour) = 2/5=40%
  - cover(Butter, Flour) = ….
- support(Butter, Milk, Sugar) = 3/5=60%
  - Cover(Butter, Milk, Sugar)= ….

The Frequent Itemsets Mining (FIM) problem

Problem 1: Frequent Itemsets Mining (FIM)

- Given:
  - A set of items I
  - A transactions database DB over I
  - A *minSupport* threshold s
- Goal: Find all frequent itemsets in DB, i.e.:

$$\{X \subseteq I \mid support(X) \geq s\}$$

transactionID	items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Support of 1-Itemsets:

(A): 75%, (B), (C): 50%, (D), (E), (F): 25%,

Support of 2-Itemsets:

(A, C): 50%,

(A, B), (A, D), (B, C), (B, E), (B, F), (E, F): 25%

Support of 3-Itemsets:

(A, B, C), (B, E, F): 25%

Support of 4-Itemsets: -

Support of 5-Itemsets: -

Support of 6-Itemsets: -

Basic concepts: association rules, support, confidence

Let X, Y be two itemsets: $X, Y \subseteq I$ *and* $X \cap Y = \emptyset$.

- Association rules: rules of the form

$$\boxed{X} \quad \rightarrow \quad \boxed{Y}$$

head or LHS (left-hand side) or antecedent of the rule — body or RHS (right-hand side) or consequent of the rule

- Support *s* of a rule**:** the percentage of transactions containing $X \cup Y$ in the DB

$$\text{support}(X \rightarrow Y) = \text{support}(X \cup Y)$$

- Confidence *c* of a rule**:** the percentage of transactions containing $X \cup Y$ in the set of transactions containing *X*. Or, in other words the conditional probability that a transaction containing *X* also contains *Y*

$$confidence(X \rightarrow Y) = P(E_Y | E_X) = \frac{P(E_X \cap E_Y)}{P(E_X)} = \frac{support(X \cup Y)}{support\ (X)}$$

E_X := Event that itemset X appears in a transaction

Explain the rules:

- {Diapers} → {Beer} (0.5%, 60%)
- {Toast bread} → {Toast cheese} (50%, 90%)

- Support and confidence are measures of rules' interestingness.
- Rules are usually written as follows: X→Y (support, confidence)

Example: association rules

- I = {Butter, Bread, Eggs, Flour, Milk, Salt, Sugar}

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

Sample rules:

- {Butter}→{Bread} (20%, 25%)
  - support(Butter ∪Bread)=1/5=20%
  - support(Butter)=4/5=80%
  - Confidence = 20%/80%=1/4=25%
- {Butter, Milk} → Sugar (60%, 75%)
  - support(Butter, Milk ∪ Sugar) = 3/5=60%
  - Support(Butter,Milk) = 4/5=80%
  - Confidence = 60%/80%=3/4=75%

The Association Rules Mining (ARM) problem

Problem 2: Association Rules Mining (ARM)

- Given:
  - A set of items *I*
  - A transactions database *DB* over *I*
  - A *minSupport* threshold *s* and a *minConfidence* threshold *c*
- Goal: Find all association rules $X \rightarrow Y$ in *DB* w.r.t. minimum support *s* and minimum confidence *c*, i.e.:

$$\{X \rightarrow Y \mid support(X \cup Y) \geq s,\ confidence(X \rightarrow Y) \geq c\}$$

  - These rules are called *strong*.

transactionID	items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Association rules:

$A \Rightarrow C$ (Support = 50%, Confidence= 66.6%)

$C \Rightarrow A$ (Support = 50%, Confidence= 100%)

…

Solving the problems

- Problem 1 (FIM): Find all frequent itemsets in *DB*, i.e.: $\{X \subseteq I \mid support(X) \geq s\}$

- Problem 2 (ARM): Find all association rules $X \rightarrow Y$ in *DB*, w.r.t. min support *s* and min confidence *c*, i.e.,: $\{X \rightarrow Y \mid support(X \cup Y) \geq s, confidence(X \rightarrow Y) \geq c, X,Y \subseteq I \text{ and } X \cap Y = \emptyset\}$

- Problem 1 is part of Problem 2:
  - Once we have support($X \cup Y$) and support(X), we can check if $X \rightarrow Y$ is strong.

- 2-step method to extract the association rules:
  - Step 1: Determine the frequent itemsets w.r.t. min support *s*:
    - "Naïve" algorithm: count the frequencies for all *k*-itemsets
      - Inefficient!!! There are $O(\binom{|I|}{k})$ such subsets
      - Total cost: $O(2^{|I|})$

      => Apriori-algorithm and variants

    FIM problem
  - Step 2: Generate the association rules w.r.t. min confidence *c*:

    from frequent itemsets *X*, generate $Y \rightarrow (X - Y)$, $Y \subset X$, $Y \neq \emptyset$, $Y \neq X$

Step 1(FIM) is the most costly, so the overall performance of an association rules mining algorithm is determined by this step.

Itemset lattice complexity

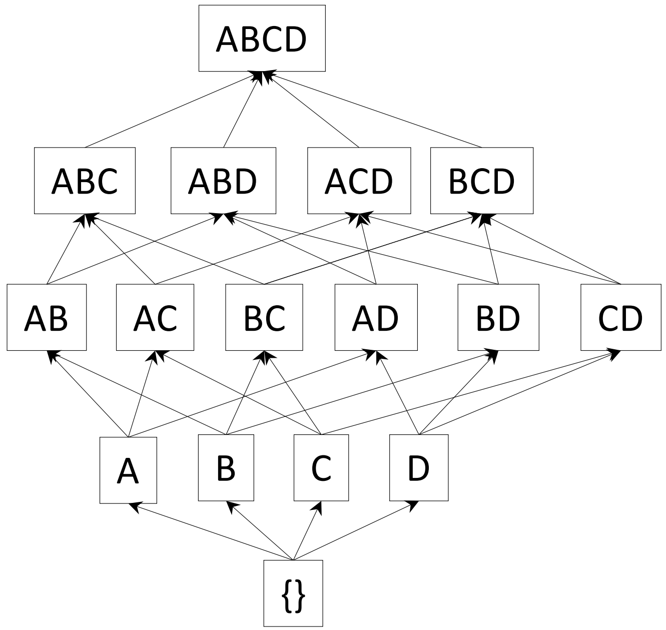
- The number of itemsets can be really huge. Let us consider a small set of items: $I = \{A,B,C,D\}$

- \# 1-itemsets: $\binom{4}{1} = \frac{4!}{(4-1)!*1!} = \frac{4!}{3!} = 4$

- \# 2-itemsets: $\binom{4}{2} = \frac{4!}{(4-2)!*2!} = \frac{4!}{2!*2!} = 6$

- \# 3-itemsets: $\binom{4}{3} = \frac{4!}{(4-3)!*3!} = \frac{4!}{3!} = 4$

- \# 4-itemsets: $\binom{4}{4} = \frac{4!}{(4-4)!*4!} = 1$



- In the general case, for $|I|$ items, there exist:

$$\binom{|I|}{1} + \binom{|I|}{2} + \ldots + \binom{|I|}{k} = 2^{|I|} - 1$$

- So, generating all possible combinations and computing their support is inefficient!