



Big Data Analytics

– Week 5: Frequent Itemset Mining –

Knowledge Discovery and Data Mining
Prof. Dr. Emmanuel Müller

What is Frequent Itemset Mining?

Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

- Given:

- A set of items $I = \{i_1, i_2, \dots, i_m\}$
- A database of transactions D , where a transaction $T \subseteq I$ is a set of items

- Task 1: find all subsets of items that occur together in many transactions.

e.g.: 85% of transactions contain the itemset {milk, bread, butter}

- Task 2: find all rules that correlate the presence of one set of items with that of another set of items in the transaction database.

e.g.: 98% of people buying tires and auto accessories also get automotive service done

- Applications:

Basket data analysis, cross-marketing, catalog design, clustering, classification, recommendation systems, etc.

Example: Basket Data Analysis

- Transaction database

$T = \{ \{ \text{butter, bread, milk, sugar} \};$
 $\{ \text{butter, flour, milk, sugar} \};$
 $\{ \text{butter, eggs, milk, salt} \};$
 $\{ \text{eggs} \};$
 $\{ \text{butter, flour, milk, salt, sugar} \} \}$

- Question of interest:

- Which items are bought together frequently?

- Applications

- Improved store layout; Cross marketing; Focused attached mailings / add-on sales

- Generalization of „association rules“ beyond marketing:

- $\text{buys}(x, \text{„diapers“}) \rightarrow \text{buys}(x, \text{„beers“})$
- $\text{major}(x, \text{„CS“}) \wedge \text{takes}(x, \text{„DB“}) \rightarrow \text{grade}(x, \text{„A“})$



Basic Notions I

- **Items** $I = \{i_1, i_2, \dots, i_m\}$: a set of literals (denoting items)
- **Itemset** X : Set of items $X \subseteq I$
- **Database** D :
 - Set of **transactions** T , each transaction is a set of items $T \subseteq I$
 - Transaction T **contains** an itemset X : $X \subseteq T$

The items in transactions and itemsets are sorted lexicographically:

- itemset $X = (x_1, x_2, \dots, x_k)$, where $x_1 \leq x_2 \leq \dots \leq x_k$

Length of an itemset: number of elements in the itemset

***k*-itemset**: itemset of length k

Basic Notions II

- **Items** $I = \{i_1, i_2, \dots, i_m\}$: a set of literals (denoting items)
- **Itemset** X : Set of items $X \subseteq I$

The **support** of an itemset X is defined as: $support(X) = |\{T \in D | X \subseteq T\}|$

Frequent itemset: an itemset X is called frequent for database D iff it is contained in more than $minSup$ many transactions: $support(X) \geq minSup$

Goal 1:

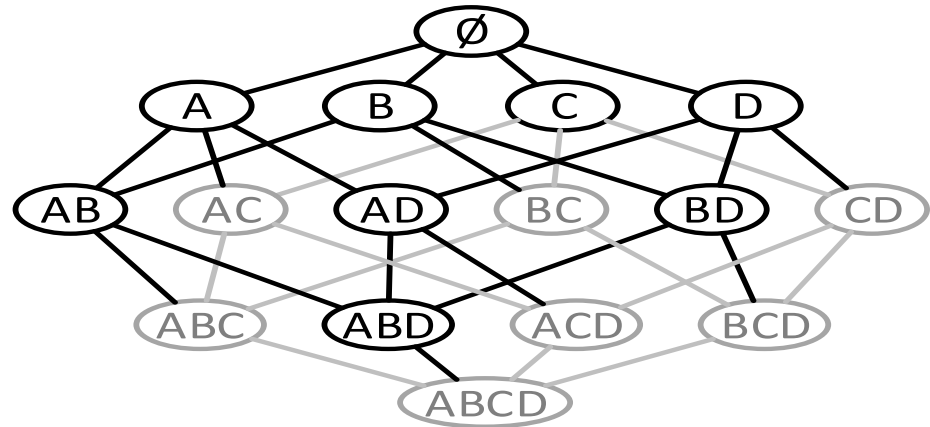
Given a database D and a threshold $minSup$,
find all frequent itemsets $X \in P(I)$.

Basic Idea

Naïve Algorithm

- count the frequency of all possible subsets of I in the database
- *too expensive* since there are 2^m such itemsets for $|I| = m$ items

↖ *cardinality of power set*



Basic Idea

The *Apriori* principle (anti-monotonicity):

Any non-empty subset of a frequent itemset is frequent, too!

$A \subseteq I$ with $\text{support}(A) \geq \text{minSup} \Rightarrow \forall A' \subset A \wedge A' \neq \emptyset: \text{support}(A') \geq \text{minSup}$

Any superset of a non-frequent itemset is non-frequent, too!

$A \subseteq I$ with $\text{support}(A) < \text{minSup} \Rightarrow \forall A' \supset A: \text{support}(A') < \text{minSup}$

Method based on the apriori principle

- First count the 1-itemsets, then the 2-itemsets, then the 3-itemsets, and so on
- When counting $(k+1)$ -itemsets, only consider those $(k+1)$ -itemsets where all subsets of length k have been determined as frequent in the previous step

