

1 Lecture 2: Pattern Discovery Basic Concepts

1.1 Frequent Itemsets (Patterns)

X = itemset

- **(absolute) support (count) of X :** Frequency or the number of occurrences of an itemset X
- **(relative) support, s :** The fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is **frequent** if the support of X is no less than a *minsup* threshold (denoted as σ): $\text{sup}(X) \geq \sigma$.

1.2 Association Rules

Association rules: $X \rightarrow Y(s, c)$:

- **Support, s :** The probability that a transaction contains $X \cup Y$
- **Confidence, c :** The conditional probability that a transaction containing X also contains Y :

$$c = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}$$

1.3 Expressing Patterns in Compressed Form

Solution 1: Closed patterns: A pattern (itemset) X is closed if X is frequent, and there exists no super-pattern $Y \supset X$, with the same support as X .

Closed pattern is a lossless compression of frequent patterns.

Solution 2: Max-patterns: A pattern X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \supset X$.

Max-pattern is a lossy compression!

1.4 Recommended readings

- R. Agrawal, T. Imielinski, and A. Swami, «Mining association rules between sets of items in large databases», in Proc. of SIGMOD'93
- R. J. Bayardo, «Efficiently mining long patterns from databases», in Proc. of SIGMOD'98
- N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, «Discovering frequent closed itemsets for association rules», in Proc. of ICDT'99
- J. Han, H. Cheng, D. Xin, and X. Yan, «Frequent Pattern Mining: Current Status and Future Directions», Data Mining and Knowledge Discovery, 15(1): 55-86, 2007

2 Lecture 3. Efficient Pattern Mining Methods

2.1 The Downward Closure Property of Frequent Patterns

The downward closure (also called «Apriori») property of frequent patterns: **Any subset of a frequent itemset must be frequent.** Apriori pruning principle: **If there is any itemset which is infrequent, its superset should not even be generated!**

Scalable mining Methods: Three major approaches

- Level-wise, join-based approach: Apriori (2.2)
- Vertical data format approach: Eclat (2.4)
- Frequent pattern projection and growth: FPgrowth (2.5)

2.2 The Apriori Algorithm

2.2.1 Algorithm pseudocode

C_k : Candidate itemset of size k

F_k : Frequent itemset of size k

TDB = transactional database

Algorithm 1 The Apriori Algorithm

```
 $k := 1$   
 $F_k :=$  frequent items # frequent 1-itemset  
while  $F_k \neq \emptyset$  do  
     $C_{k+1} :=$  candidates generated from  $F_k$  # candidate generation  
    Derives  $F_{k+1}$  by counting candidates in  $C_{k+1}$  with respect to TDB at  
    minsup  
     $k := k + 1$   
end while  
return  $\cup_k F_k$  # return  $F_k$  generated at each level
```

2.2.2 How to generate candidates?

- Step1: self-joining F_k
- Step2: pruning

Algorithm 2 Step1: self-joining F_k

```
insert into  $C_k$   
select  $p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$   
from  $F_{k-1}$  as  $p, F_{k-1}$  as  $q$   
where  $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$ 
```

Algorithm 3 Step2: pruning

```
for all itemsets  $c$  in  $C_k$  do
  for all  $(k-1)$  subsets  $s$  of  $c$  do
    if  $s$  is not in  $F_{k-1}$  then
      delete  $c$  from  $C_k$ 
    end if
  end for
end for
```

2.3 Extensions or Improvements of Apriori

- Reduce passes of transaction database scans
 - Partitioning
 - Dynamic itemset counting
- Shrink the number of candidates
 - Hashing
 - Pruning by support lower bounding
 - Sampling
- Exploring special data structures
 - Tree projection
 - H-miner
 - Hypercube decomposition

2.3.1 Partitioning

Theorem: *Any itemset that is potentially frequent in TDB must be frequent in at least one of the partitions of TDB*

Method: Scan Database Only Twice:

- Scan 1: Partition database (how?) and find local frequent patterns
- Scan 2: Consolidate global frequent patterns (how to?)

2.3.2 Direct Hashing and Pruning (DHP)

Observation: *A k -itemset whose corresponding hashing bucket count is below the threshold cannot be frequent*

2.4 Vertical Data Format

ECLAT - Equivalence Class Transformation

Frequent patterns are derived based on vertical intersections. To accelerate data mining you can use **diffset**: only keep track of differences of tids.

2.5 A Pattern Growth Approach

FP-tree - frequent pattern tree

TID	Items in the Transaction	Ordered, frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

Figure 1: Transational DB

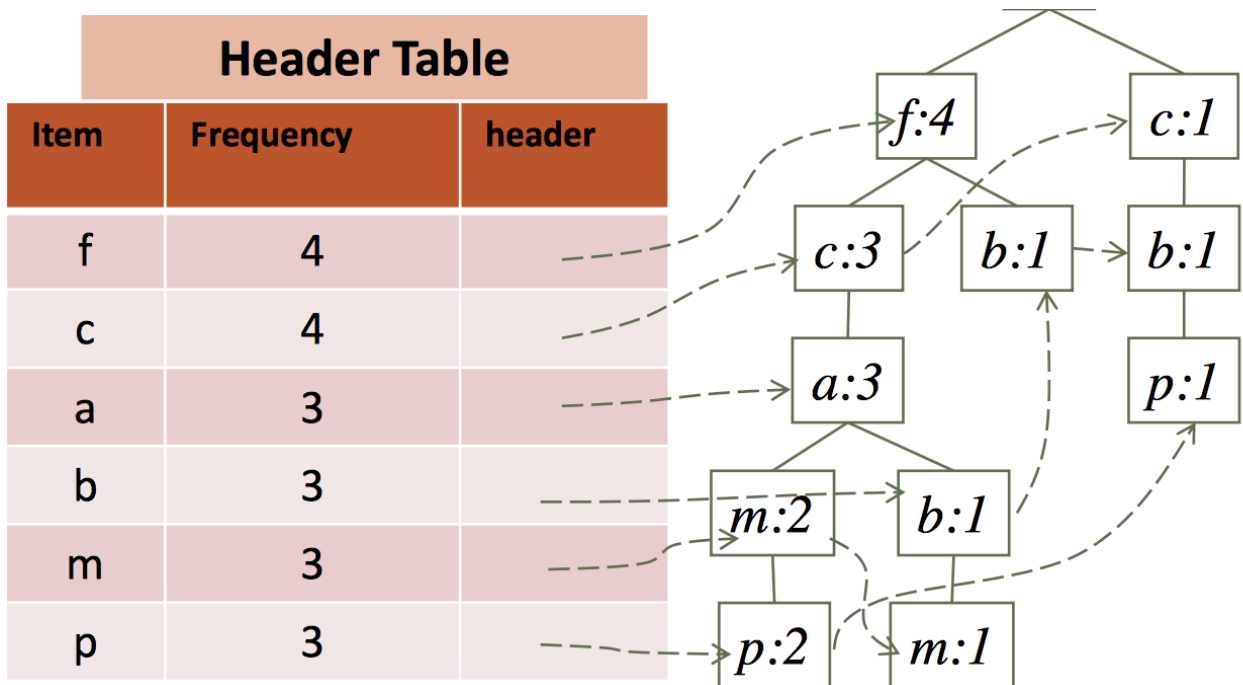


Figure 2: FP-tree

2.6 CLOSET+: Mining Closed Itemsets by Pattern-Growth

Itemset merging: *If Y appears in every occurrence of X, then Y is merged with X*

2.7 Recommended readings

- R. Agrawal and R. Srikant, «Fast algorithms for mining association rules», VLDB'94
- A. Savasere, E. Omiecinski, and S. Navathe, «An efficient algorithm for mining association rules in large databases», VLDB'95
- J. S. Park, M. S. Chen, and P. S. Yu, «An effective hash-based algorithm for mining association rules», SIGMOD'95

- S. Sarawagi, S. Thomas, and R. Agrawal, «Integrating association rule mining with relational database systems: Alternatives and implications», SIGMOD'98
- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, «Parallel algorithm for discovery of association rules», Data Mining and Knowledge Discovery, 1997
- J. Han, J. Pei, and Y. Yin, «Mining frequent patterns without candidate generation», SIGMOD'00
- M. J. Zaki and Hsiao, «CHARM: An Efficient Algorithm for Closed Itemset Mining», SDM'02
- J. Wang, J. Han, and J. Pei, «CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets», KDD'03
- C. C. Aggarwal, M.A., Bhuiyan, M. A. Hasan, «Frequent Pattern Mining Algorithms: A Survey», in Aggarwal and Han (eds.): Frequent Pattern Mining, Springer, 2014

3 Lecture 4: Pattern Evaluation

3.1 Interestingness Measures: Lift and χ^2

3.1.1 Interestingness Measure: Lift

Lift - measure of dependent/correlated events:

$$\text{lift}(B, C) = \frac{c(B \rightarrow C)}{s(C)} = \frac{s(B \cup C)}{s(B) \times s(C)}$$

Lift(B, C) may tell how B and C are correlated:

- $\text{Lift}(B, C) = 1$: B and C are independent
- $\text{Lift}(B, C) > 1$: positively correlated
- $\text{Lift}(B, C) < 1$: negatively correlated

3.1.2 Interestingness Measure: χ^2

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

General rules:

- $\chi^2 = 0$: independent
- $\chi^2 > 0$: correlated, either positive or negative, so it needs additional test

Too many null transactions may lead to invalid correlation result!

3.2 Null Invariance Measures

$$\begin{aligned}AllConf(A, B) &= \frac{s(A \cup B)}{\max\{s(A), s(B)\}} \\Jaccard(A, B) &= \frac{s(A \cup B)}{s(A) + s(B) - s(A \cup B)} \\Cosine(A, B) &= \frac{s(A \cup B)}{\sqrt{s(A) \times s(B)}} \\Kulczynsky(A, B) &= \frac{1}{2} \left(\frac{s(A \cup B)}{s(A)} + \frac{s(A \cup B)}{s(B)} \right) \\MacConf(A, B) &= \max \left\{ \frac{s(A)}{s(A \cup B)}, \frac{s(B)}{s(A \cup B)} \right\}\end{aligned}$$

3.3 Imbalance Ratio

IR (Imbalance Ratio): measure the imbalance of two itemsets A and B in rule implications:

$$IR(A, B) = \frac{|s(A) - s(B)|}{s(A) + s(B) - s(A \cup B)}$$

Kulczynski and Imbalance Ratio (IR) together present a clear picture

3.4 Recommended Readings

- C. C. Aggarwal and P. S. Yu. A New Framework for Itemset Generation. PODS'98
- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94
- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE'03
- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02
- T. Wu, Y. Chen and J. Han, Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework, Data Mining and Knowledge Discovery, 21(3):371-397, 2010

4 Lecture 4: Mining Diverse Patterns

4.1 Mining Multi- Level Associations

Items often form hierarchies. How to set min-support thresholds? **Level-reduced min-support:** items at the lower level are expected to have lower support.

Efficient mining: **shared** multi-level mining. Use the lowest min-support to pass down the set of candidates.

Redundancy¹ filtering: some rules may be redundant due to «ancestor»² relationships between items. A rule is **redundant** if:

- its support is close to the «expected» value, according to its «ancestor» rule
- it has a similar confidence as its «ancestor».

It is necessary to have customized min-support settings for different kinds of items: group-based «individualized» min-support.

4.2 Mining Multi-Dimensional Associations

Rules can be single-dimensional or multi-dimensional:

- Single-dimentional: $buys(X, \text{«milk»}) \Rightarrow buys(X, \text{«bread»})$
- Inter-dimension association rule: $age(X, \text{«18-25»}) \wedge occupation(X, \text{«student»}) \Rightarrow buys(X, \text{«coke»})$
- Hybrid-dimension association rules: $age(X, \text{«18-25»}) \wedge buys(X, \text{«popcorn»}) \Rightarrow buys(X, \text{«coke»})$

Attributes can be categorical or numerical

4.3 Mining Quantitative Associations

Methods:

- Static discretization based on predefined concept hierarchies
- Dynamic discretization based on data distribution
- Clustering: distance-based association
- Deviation analysis

4.4 Mining Negative Correlations

- Rare patterns = very low support but interesting
- Negative patterns = negatively correlated, unlikely to happen together

A support-based definition: if itemsets A and B are both frequent but rarely occur together, i.e., $\text{sup}(A \cup B) \ll \text{sup}(A) \times \text{sup}(B)$ then A and B are negatively correlated.

The support-based definition is not null-invariant!

A Kulczynski measure-based definition: if itemsets A and B are frequent but $\frac{P(A|B) + P(B|A)}{2} < \varepsilon$, where ε is a negative pattern threshold, then A and B are negatively correlated.

¹Redundancy - избыточность

²Ancestor – предок

4.5 Mining Compressed Patterns

4.5.1 Mining Compressed Patterns

Pattern distance measure:

$$Dist(P_1, P_2) = 1 - \frac{|T(P_1) \cap T(P_2)|}{|T(P_1) \cup T(P_2)|}$$

δ -clustering. For each pattern P , find all patterns which can be expressed by P and whose distance to P is within δ (δ -cover). All patterns in the cluster can be represented by P = compressed patterns.

Method for efficient, direct mining of compressed frequent patterns: Xin et al., VLDB'05.

4.5.2 Redundancy-Aware Top-k Patterns

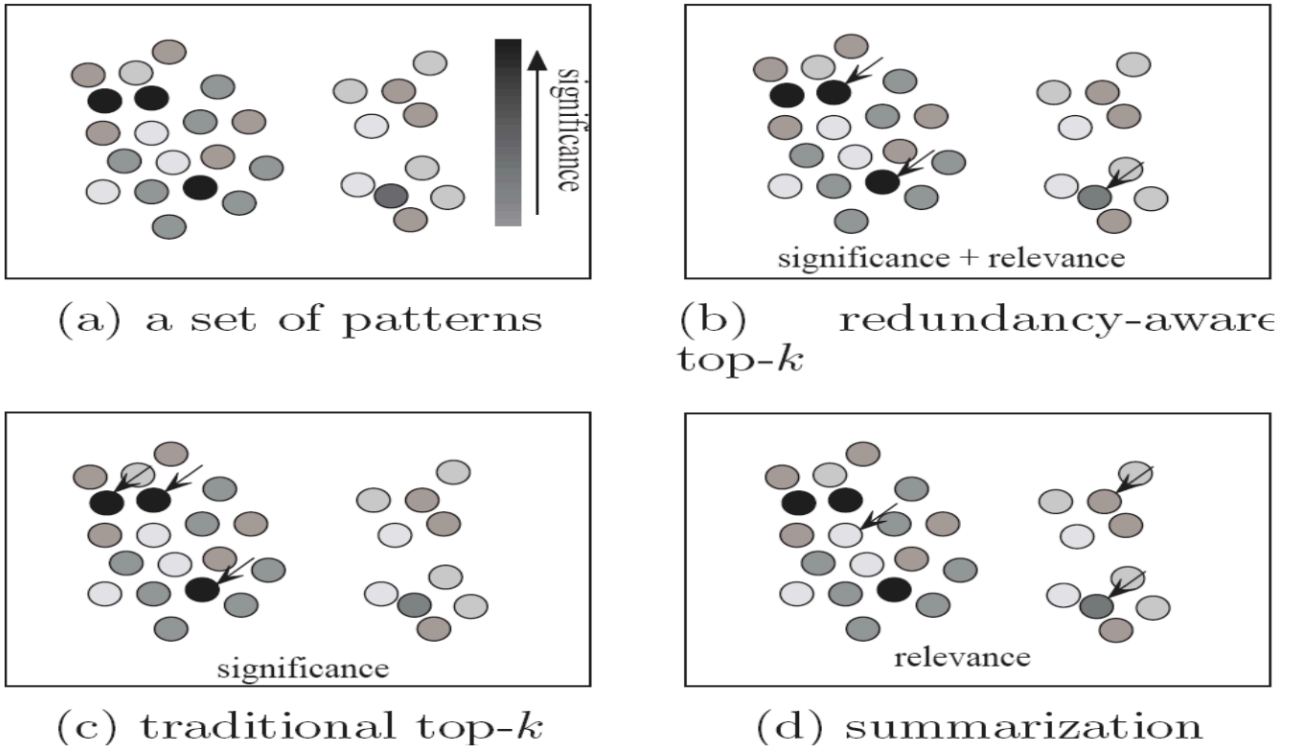


Figure 3: Desired patterns: high significance & low redundancy

Use **MMS (Maximal Marginal Significance)** for measuring the combined significance of a pattern set: Xin et al., Extracting Redundancy-Aware Top-K Patterns, KDD'06.

4.6 Mining Colossal Patterns

4.6.1 Pattern-Fusion

Pattern fusion strategy: fuse small patterns together in one step to generate new pattern candidates of significant sizes.

Subpatterns α_1 to α_k cluster tightly around the colossal pattern α by sharing a similar support. Such subpatterns are **core patterns** of α . A colossal pattern can be generated by merging a set of core patterns.

4.6.2 Robustness of Colossal Patterns

For a frequent pattern α , a subpattern β is a τ -core pattern of α if β shares a similar support set with α , i.e.,

$$\frac{|D_\alpha|}{|D_\beta|} \geq \tau, 0 < \tau \leq 1,$$

where τ is called the **core ratio**.

(d, τ)-robustness³: a pattern α is (d, τ) -robust if d is the maximum number of items that can be removed from α for the resulting pattern to remain a τ -core pattern of α . For a (d, τ) -robust pattern α , it has $\Omega(2^d)$ core patterns.

Robustness of Colossal Patterns: a colossal pattern tends to have much more core patterns than small patterns. Such core patterns can be clustered together to form «dense balls» based on pattern distance defined by

$$Dist(\alpha, \beta) = 1 - \frac{|D_\alpha \cap D_\beta|}{|D_\alpha \cup D_\beta|}$$

4.6.3 The Pattern-Fusion Algorithm

- Initialization (Creating initial pool): Use an existing algorithm to mine all frequent patterns up to a small size, e.g., 3
- Iteration (Iterative Pattern Fusion):
 - At each iteration, K seed patterns are randomly picked from the current pattern pool
 - For each seed pattern thus picked, we find all the patterns within a bounding ball centered at the seed pattern
 - All these patterns found are fused together to generate a set of super-patterns
 - All the super-patterns thus generated form a new pool for the next iteration
- Termination: when the current pool contains no more than K patterns at the beginning of an iteration

4.7 Recommended Readings

- R. Srikant and R. Agrawal, «Mining generalized association rules», VLDB'95
- Y. Aumann and Y. Lindell, «A Statistical Theory for Quantitative Association Rules», KDD'99
- D. Xin, J. Han, X. Yan and H. Cheng, «On Compressing Frequent Patterns», Knowledge and Data Engineering, 60(1): 5-29, 2007

³Robustness - прочность

- D. Xin, H. Cheng, X. Yan, and J. Han, «Extracting Redundancy-Aware Top-K Patterns», KDD'06
- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, «Mining Colossal Frequent Patterns by Core Pattern Fusion», ICDE'07
- J. Han, H. Cheng, D. Xin, and X. Yan, «Frequent Pattern Mining: Current Status and Future Directions», Data Mining and Knowledge Discovery, 15(1): 55-86, 2007

5 Constraint-Based Pattern Mining

5.1 Meta-Rule Guided Mining

In general, (meta) rules can be in the form of

$$P_1 \wedge P_2 \wedge \dots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \dots \wedge Q_r$$

Method to find meta-rules:

- Find frequent ($l + r$) predicates (based on min-support)
- Push constants deeply when possible into the mining process
- Also, push `min_conf`, `min_correlation`, and other measures as early as possible (measures acting as constraints)

5.2 Kinds of Constraints

- Pattern space pruning constraints
 - Anti-monotonic: If constraint c is violated, its further mining can be terminated
 - Monotonic: If c is satisfied, no need to check c again
 - Succinct⁴: if the constraint c can be enforced by directly manipulating the data
 - Convertible: c can be converted to monotonic or anti-monotonic if items can be properly ordered in processing
- Data space pruning constraints
 - Data succinct: Data space can be pruned at the initial pattern mining process
 - Data anti-monotonic: If a transaction t does not satisfy c , then t can be pruned to reduce data processing effort

Anti-monotonic constraints have more pruning power than monotonic constraints.

5.2.1 Pattern space pruning constraints

Constraint c is **anti-monotone**: if an itemset S violates constraint c , so does any of its superset. That is, mining on itemset S can be terminated. For example, constraint $\text{sup}(S) \geq \sigma$ is anti-monotone.

A constraint c is **monotone**: if an itemset S satisfies the constraint c , so does any of its superset. That is, we do not need to check c in subsequent mining. For example, constraints $\sum(S.\text{price}) \geq v$ or $\min(S.\text{price}) \leq v$ are monotone.

⁴Succinct - краткий

5.2.2 Data space pruning constraints

A constraint **c** is **data anti-monotone**: if a data entry **t** cannot satisfy a pattern **p** under constraint **c**, **t** cannot satisfy **p**'s superset either. That's why, data entry **t** can be pruned.

Succinctness: if the constraint **c** can be enforced by directly manipulating the data.

Convertible constraints: convert tough⁵ constraints into (anti-)monotone by proper ordering of items in transactions. For example, ordering items in value-descending order makes the constraint $avg(S.profit) > 20$ anti-monotone *if the patterns grow in the right order*.

5.3 Recommended Readings

- R. Srikant, Q. Vu, and R. Agrawal, «Mining association rules with item constraints», KDD'97
- R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang, Exploratory mining and pruning optimizations of constrained association rules», SIGMOD'98
- G. Grahne, L. Lakshmanan, and X. Wang, «Efficient mining of constrained correlated sets», ICDE'00
- J. Pei, J. Han, and L. V. S. Lakshmanan, «Mining Frequent Itemsets with Convertible Constraints», ICDE'01
- J. Pei, J. Han, and W. Wang, «Mining Sequential Patterns with Constraints in Large Databases», CIKM'02
- F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, «ExAnte: Anticipated Data Reduction in Constrained Pattern Mining», PKDD'03
- F. Zhu, X. Yan, J. Han, and P. S. Yu, «gPrune: A Constraint Pushing Framework for Graph Pattern Mining», PAKDD'07

⁵Tough - жесткий