Data Mining II / Adv. Topics in Data Science

Association Rules

Rita P. Ribeiro

2023/2024

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

DEPARTAMENTO DE CIÊNCIA DE COMPUTADORES FACULDADE DE CIÊNCIAS DA UNIVERSIDADE DO PORTO

Summary

Mining Association Rules

Problem Definition

- Given:
  - data set of transactions D
  - minimal support *minsup*
  - minimal confidence *minconf*
- Obtain:
  - **all** association rules

    $$X \rightarrow Y \quad (s = Sup, c = Conf)$$

    such that

    $$Sup \geq minsup \text{ and } Conf \geq minconf$$

Apriori Algorithm

The **Apriori Algorithm** [Agrawal and Srikant, 1994] works in two steps:

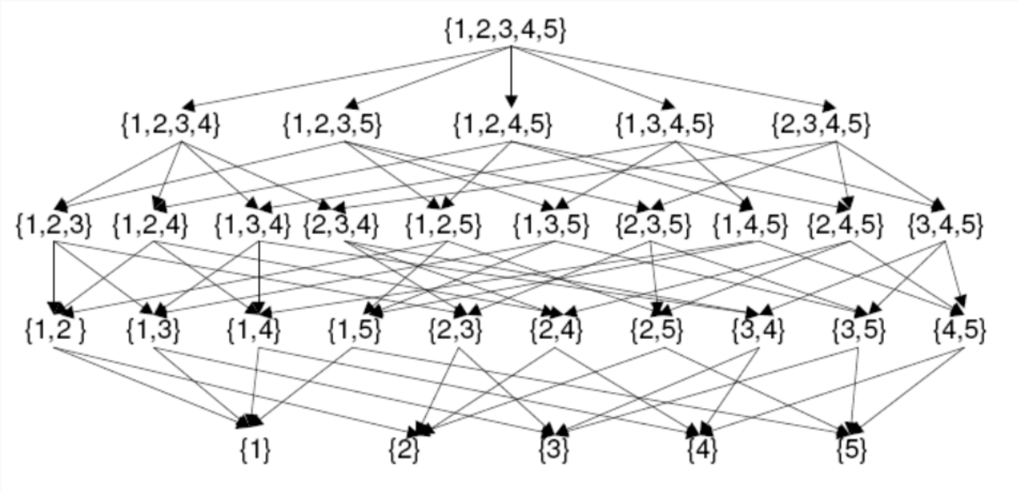
1. **Frequent itemset generation**
   - itemsets with *support* $\geq$ *minsup*
2. **Rule generation**
   - generate all confident association rules from the frequent itemsets, i.e. rules with *confidence* $\geq$ *minconf*

Apriori Algorithm (cont.)

- Problem:
  - there is a very large number of candidate frequent itemsets!
    - for transactions with k items, there are $2^k - 1$ distinct subsets.
- Downward Closure Property
  - every subset of a frequent itemset must also be frequent.
    - ex: if $\{A1, A2, A4\}$ is frequent, so is $\{A1, A2\}$ because every transaction containing $\{A1, A2, A4\}$ also contains $\{A1, A2\}$.
  - thus, every superset of an infrequent itemset is also infrequent.
    - ex: if $\{A1, A2\}$ is infrequent, so is $\{A1, A2, A4\}$.
- Apriori Pruning Principle:
  - if an itemset is below the minimal support, discard all its supersets.

Example - 1

Search Space for 5 items



Example - 1 (cont.)

- Apriori enumerates and counts the support of patterns with increasing length.
- Starts looking for frequent itemsets of size 1 (F_1), assuming *minsup* $= 50\%$ (2 transactions)
- $C_1 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$

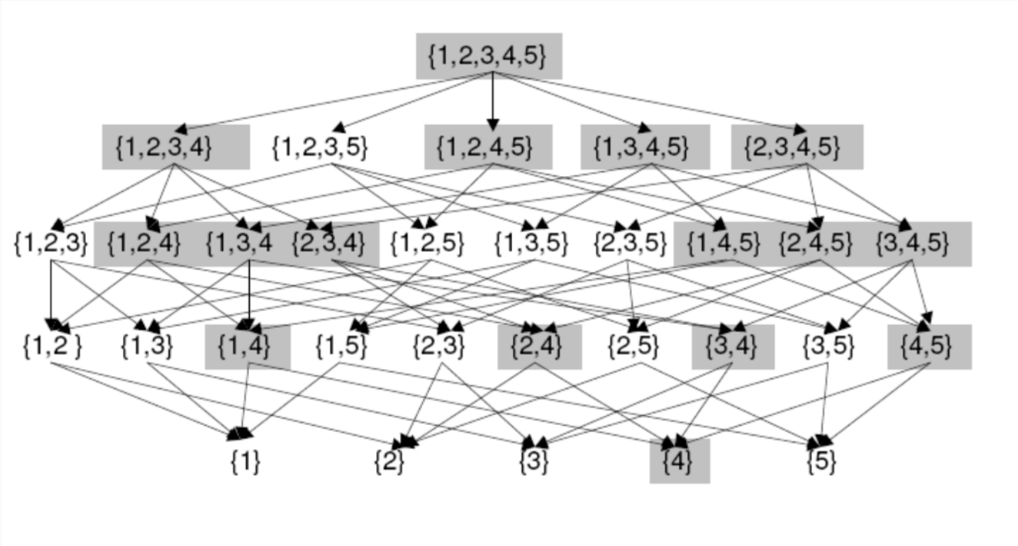
TID	ITEM-SET
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

ITEM-SET	Support
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

- $F_1 = \{\{1\}, \{2\}, \{3\}, \{5\}\}$

Example - 1 (cont.)

- Filtered Search Space for 5 items (after removing item "4")



Example - 1 (cont.)

- Looks for frequent itemsets of size 2 (F_2) from frequent itemsets of size 1 (F_1)
- Candidates $C_2 = \{\{a, b\} | \{a\} \in F_1 \wedge \{b\} \in F_1\}$
- $C_2 = \{\{1, 2\}, \{1, 3\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 5\}\}$

ITEM-SET	Support
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

- $F_2 = \{\{1, 3\}, \{2, 3\}, \{2, 5\}, \{3, 5\}\}$

Example - 1 (cont.)

- Looks for frequent itemsets of size 3 (F_3) from frequent itemsets of size 2 (F_2)
- Generation:
  $C0_3 = \{\{a, b, c\} | \{a, b\} \in F_2 \wedge \{a, c\} \in F_2\}$
- Filter:
  $C_3 = \{\{a, b, c\} | \{a, b, c\} \in C0_3 \wedge \forall x \in \{a, b, c\}\ S - \{x\} \in F_2\}$
- $C_3 = \{\{2, 3, 5\}\}$

ITEM-SET	Suporte
{2,3,5}	2

- $F_3 = \{\{2, 3, 5\}\}$
- There are no frequent itemsets of size 4

Example - 2

A	B	C	D
1			
1	1	1	
		1	
1	1	1	1
	1		
1			1
1	1	1	
		1	1
1	1	1	

Pass 1

- $minsup = 40\%$
- $C_1 = \{\{A\}, \{B\}, \{C\}, \{D\}\}$
- $F_1 = \{\{A\}, \{B\}, \{C\}\}$

Example - 2 (cont.)

A	B	C	D
1			
1	1	1	
		1	
1	1	1	1
	1		
1			1
1	1	1	
		1	1
1	1	1	

Pass 2

- *minsup* $= 40\%$
- $C_2 = \{\{A, B\}, \{A, C\}, \{B, C\}\}$
- $F_2 = \{\{A, B\}, \{A, C\}, \{B, C\}\}$

Example - 2 (cont.)

A	B	C	D
1			
1	1	1	
		1	
1	1	1	1
	1		
1			1
1	1	1	
		1	1
1	1	1	

Pass 3

- *minsup* $= 40\%$
- $C_3 = \{\{A, B, C\}\}$
- $F_3 = \{\{A, B, C\}\}$

Example - 2 (cont.)

Output

- frequent itemsets ($minsup = 40\%$)

$\{A\}$ 66%
$\{A, B\}$ 44%
$\{A, B, C\}$ 44%
$\{B\}$ 55%
$\{A, C\}$ 44%
$\{C\}$ 66%
$\{B, C\}$ 44%

- rules ($minconf = 80\%$)

$\{B\} \rightarrow \{A\}$	(sup = 44%, conf = 80%)
$\{B\} \rightarrow \{C\}$	(sup = 44%, conf = 80%)
$\{B, C\} \rightarrow \{A\}$	(sup = 44%, conf = 100%)
$\{B, A\} \rightarrow \{C\}$	(sup = 44%, conf = 100%)
$\{B\} \rightarrow \{A, C\}$	(sup = 44%, conf = 80%)

Step 1 - identifying frequent itemsets

- It is a **level-wise** algorithm
  - it traverses the itemset lattice one level at a time, from frequent 1-itemsets to the maximum size of frequent itemsets.
- It employs a **generate-and-test** strategy for finding frequent itemsets
  - at each iteration, new candidate itemsets are generated from the frequent itemsets found in the previous iteration; the support for each candidate itemset is then counted and tested against minsup.

Step 1 - identifying frequent itemsets (cont.)

- Candidate generation (Self-Join step)
  - generates new candidate k-itemsets based on the frequent (k-1)-itemsets found in the previous iteration.
- Candidate pruning (Prune step)
  - eliminates some of the candidate k-itemsets using the support-based pruning strategy.

Step 1 - identifying frequent itemsets (cont.)

- Self-Join Example:

  Given the size k candidates
  $\{A, B, C\}$
  $\{A, B, D\}$
  $\{A, C, D\}$
  $\{B, C, D\}$
  $\{A, B, E\}$
  $\{B, C, E\}$
  and assuming that in each itemset the items are lexicographically sorted
- Which are the candidates of size $k + 1$?
- What is the most efficient way of finding them (without repetitions)?

Step 1 - identifying frequent itemsets (cont.)

- Look for pairs of sets with the same prefix of size $k-1$
  $\{A, B, C\}$ and $\{A, B, D\}$
- Combine both, keeping the prefix
  $\{A, B, C, D\}$
- This way
  - No frequent set is unnoticed
  - No candidate is generated more than once

Step 1 - identifying frequent itemsets (cont.)

- Prune Example:

  $F_3 = \{\{A, B, C\}, \{A, B, D\}, \{A, C, D\}, \{A, C, E\}, \{B, C, D\}\}$

  $C_4 = \{\{A, B, C, D\}, \{A, C, D, E\}\}$

  but $\{A, C, D, E\}$ can be pruned away

  because $\{A, D, E\} \notin F_3$

- Note:
  - Prune maintains the completeness of the process

Step 2 - rule generation

- Given a frequent set $\{A, B, C, D\}$
- Which are the possible rules?
  - $\{A, B, C\} \rightarrow \{D\}$
  - $\{A, B, D\} \rightarrow \{C\}$
  - $\{A, B\} \rightarrow \{C, D\}$
- How to generate them systematically?
- How to reduce the search space?

Step 2 - rule generation (cont.)

- The rules are generated as follows:
  - generates all non-empty subsets s of each frequent itemset l
  - for each subset s computes the confidence of the rule $(l - s) \rightarrow s$
  - selects the rules whose confidence is higher than *minconf*

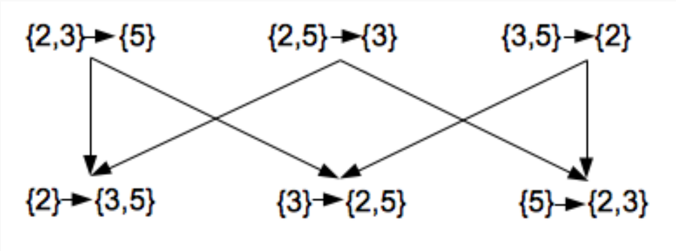
Step 2 - rule generation (cont.)

Consider again

Cliente (TID)	Itens (Item-set)
100	1, 3, 4
200	2, 3, 5,
300	1, 2, 3, 5,
400	2, 5,

and $I = \{2, 3, 5\} (= F_3)$

- Rules generated from the frequent itemset $\{2, 3, 5\}$



- Select rules $(I - a) \rightarrow a$, where $a \subseteq I$, with $minconf = 1$

$$conf((I - a) \rightarrow a) = \frac{sup(I)}{sup(I - a)}$$

Step 2 - rule generation (cont.)

- Rules with 1 consequent

$\{2, 3\} \rightarrow \{5\}$ (conf= 2/2)
$\{2, 5\} \rightarrow \{3\}$ (conf= 2/3) eliminated because $minconf = 1$
$\{3, 5\} \rightarrow \{2\}$ (conf= 2/2)

- Rules with 2 consequents

$\{3\} \rightarrow \{2, 5\}$ (conf= 2/3) eliminated because $minconf = 1$

- we don't need to worry about rules with item 3 in the consequent, because any rule obtained from $\{2, 5\} \rightarrow \{3\}$ will have a $conf < 2/3$

> Moving items from the antecedent to the consequent never changes support and never increases confidence.

Number of DB scans

- 1 to count frequencies of C_1
- C_2 built in memory
- 2 to count frequencies of C_2
- ...
- n to count frequencies of C_n
- Rule generation does not need to scan DB
- Number of scans is n
  - if the size of the largest frequent set is n or $n-1$

Complexity factors

- Number of items
- Number of transactions
- Minimal support
- Average size of transactions
- Number of frequent sets
- Average size of a frequent size
- Number of DB scans
  - k or $k+1$, where k is the size of the largest frequent set

Exercises

1. Consider the following set of transactions:

$$\{\{A, B, C\}, \{A, C\}, \{B, D\}, \{B, C, D\}, \{A\}\}$$

   Using the Apriori algorithm with *minsup* $= 40\%$ and *minconf* $= 70\%$

   - find the frequent itemsets
   - find the set of relevant rules

Exercises (cont.)

2. Consider the following set of transactions:

   Using the Apriori algorithm with *minsup* $= 30\%$ and *minconf* $= 80\%$

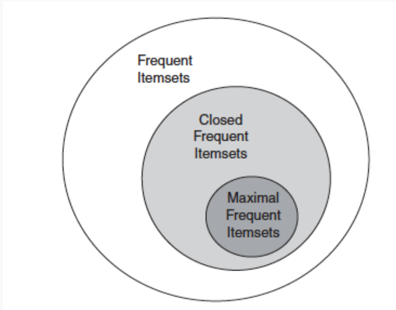
   - find the frequent itemsets
   - find the set of relevant rules

TID	Itemset
1	A D E
2	B C D
3	A C E
4	A C D E
5	A E
6	A C D
7	B C
8	A C D E
9	B C E
10	A D E

Compact Representation of Itemsets

- The number of frequent itemsets produced from a transaction data set can be very large.
- It is useful to identify a small representative set of itemsets from which all other frequent itemsets can be derived.

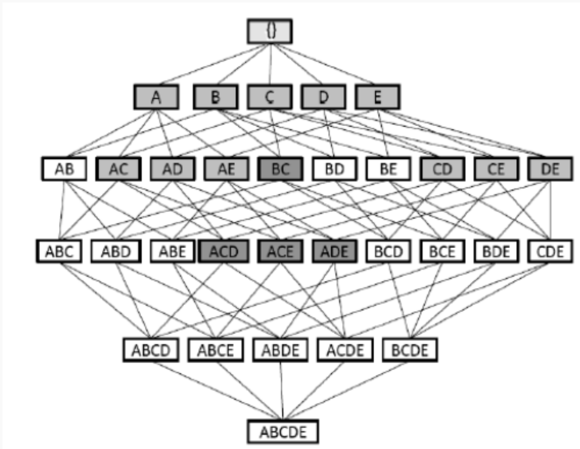
- Two such representations are:
  - closed
  - maximal



Compact Representation of Itemsets (cont.)

- s is a **closed frequent itemset** if it is a frequent itemset that has no frequent supersets with the same support.
- Example: find closed frequent itemsets with $minsup = 30\%$

TID	Itemset
1	A D E
2	B C D
3	A C E
4	A C D E
5	A E
6	A C D
7	B C
8	A C D E
9	B C E
10	A D E



closed frequent itemsets are:
$\{A\}, \{C\}, \{D\}, \{E\}, \{A, C\}, \{A, D\}, \{A, E\}, \{B, C\}, \{C, D\}, \{C, E\}, \{A, C, D\}, \{A, C, E\}, \{A, D, E\}$

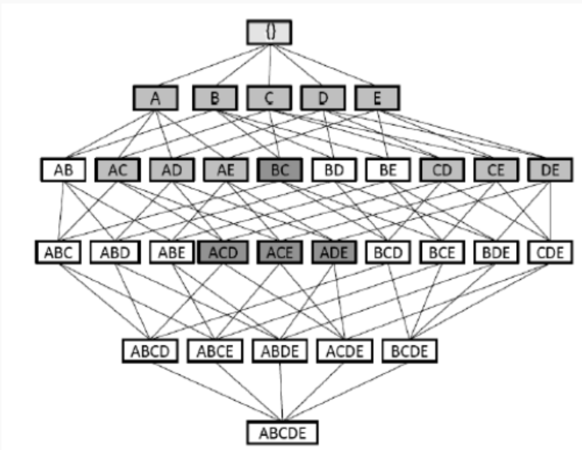
Compact Representation of Itemsets (cont.)

- The set of all **closed frequent itemsets** preserves the knowledge about the support values of all frequent itemsets.
  - $\{D, E\}$ is a non closed frequent itemset. What is its support?
  - As it is not closed, its support must be equal to one of its immediate supersets.
  - look for the most frequent closed itemset that contains $\{D, E\}$: $\{A, D, E\}$
  - $sup(\{D, E\}) = sup(\{A, D, E\})$

Compact Representation of Itemsets (cont.)

- s is a **maximal frequent itemset** if it is a frequent itemset for which none of its supersets is frequent.
- Example: find maximal frequent itemsets with $minsup = 30\%$

TID	Itemset
1	A D E
2	B C D
3	A C E
4	A C D E
5	A E
6	A C D
7	B C
8	A C D E
9	B C E
10	A D E



maximal frequent itemsets are:

$\{B, C\}, \{A, C, D\}, \{A, C, E\}, \{A, D, E\}$

Compact Representation of Itemsets (cont.)

- From the **maximal itemsets** is possible to derive all frequent itemsets (**not their support**) by computing all non-empty intersections.
  - subsets of the maximal frequent itemset $\{A, C, D\}$ are frequent itemsets
  - $\{A\}$, $\{C\}$, $\{D\}$, $\{A, C\}$, $\{A, D\}$, $\{C, D\}$
- There are algorithms that take advantage of this compact representation of frequent itemsets.

References

References

Aggarwal, C. C. (2015).
Data Mining, The Texbook.
Springer.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I. (1996).
Fast discovery of association rules.
In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. American Association for Artificial Intelligence.

Agrawal, R. and Srikant, R. (1994).
Fast algorithms for mining association rules in large databases.
In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499. Morgan Kaufmann Publishers Inc.

Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997).
Dynamic itemset counting and implication rules for market basket data.
In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, volume 26, pages 255–264. ACM.

Domingo, C., Gavalda, R., and Watanabe, O. (1998).
On-line sampling methods for discovering association rules.

References (cont.)

Gama, J. (2016).
Association rules.
Slides.

Gama, J., Oliveira, M., Lorena, A. C., Faceli, K., and de Leon Carvalho, A. P. (2015).
Extração de Conhecimento de Dados - Data Mining.
Edições Sílabo, 2nd edition.

Han, J., Kamber, M., and Pei, J. (2011).
Data Mining: Concepts and Techniques.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

Han, J., Pei, J., Yin, Y., and Mao, R. (2004).
Mining frequent patterns without candidate generation: A frequent-pattern tree approach.
Data Mining and Knowledge Discovery, 8(1):53–87.

Jorge, A. (2016).
Association rules.
Slides.

Liu, B. (2011).
Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data.
Springer, 2nd edition.

References (cont.)

Savasere, A., Omiecinski, E., and Navathe, S. B. (1995).
An efficient algorithm for mining association rules in large databases.
In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 432–444. Morgan Kaufmann Publishers Inc.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005).
***Introduction to Data Mining*.**
Addison Wesley.

Toivonen, H. (1996).
Sampling large databases for association rules.
In *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB '96, pages 134–145. Morgan Kaufmann Publishers Inc.

Torgo, L. (2017).
***Data Mining with R: Learning with Case Studies*.**
Chapman and Hall/CRC, 2nd edition.