Chapter V: Mining frequent patterns, associations and correlations

Knowledge Discovery in Databases

Luciano Melodia M.A.
Evolutionary Data Management, Friedrich-Alexander University Erlangen-Nürnberg
Summer semester 2021

Chapter V: Mining frequent patterns, associations and correlations

Basic Concepts.

Scalable frequent-itemset-mining methods.

- Apriori: a candidate-generation-and-test approach.
- Improving the efficiency of apriori.
- FPGrowth: a frequent-pattern-growth approach.
- ECLAT: frequent-pattern mining with vertical data format.
- Mining closed itemsets and max-itemsets.

Generating association rules from frequent itemsets.

Which patterns are interesting? Pattern-evaluation methods.

Summary.

What is frequent-pattern analysis?

Frequent pattern:

- A pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a dataset.
- A pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a dataset.

Motivation: Finding inherent regularities in data:

- What products are often purchased together? Beer and diapers?!
- What are the subsequent purchases after buying a PC?
- FPGrowth: a frequent-pattern-growth approach.
  - "Who bought this has often also bought . . ."
- What kinds of DNA are sensitive to this new drug?
- Can we automatically classify Web documents?

Applications:

- Basket-data analysis, cross-marketing, catalog design, sale-campaign analysis, Web-log (click-stream) analysis, and DNA-sequence analysis.

Why is frequent-pattern mining important?

A frequent pattern is an intrinsic and important property of a dataset.

Foundation for many essential data-mining tasks:

- Association, correlation, and causality analysis.
- Sequential, structural (e.g., sub-graph) patterns.
- Pattern analysis in spatiotemporal, multimedia, time-series, and stream data.
- Classification: discriminative, frequent-pattern analysis.
- Cluster analysis: frequent-pattern-based clustering.
- Data warehousing: iceberg cube and cube gradient.
- Semantic data compression: fascicles (Jagadish, Madar, and Ng, VLDB'99).
- Broad applications.

An example

From: Martin Lindstrom: Brandwashed. Random House, 2011:

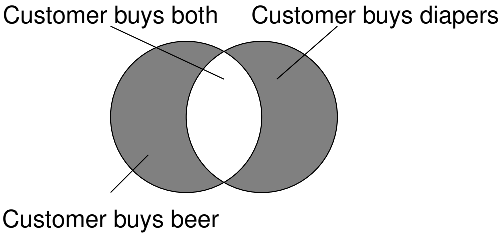
It is by crunching these numbers that the data-mining industry has uncovered some even more surprising factoids:

Did you know, for example, that at Walmart a shopper who buys a Barbie doll is 60 percent more likely to purchase one of three types of candy bars? Or that toothpaste is most often bought alongside canned tuna? Or that a customer who buys a lot of meat is likely to spend more money in a health-food store than a non-meat-eater? Or what about the data revealed to one Canadian grocery chain that customers who bought coconuts also tended to buy prepaid calling cards? At first, no one in store management could figure out what was going on. What could coconuts possibly have to do with calling cards?

Finally it occurred to them that the store served a huge population of shoppers from the Caribbean islands and Asia, both of whose cuisines use coconuts in their cooking. Now it made perfect sense that these Caribbean and Asian shoppers were buying prepaid calling cards to check in with their extended families back home.

An example

TID	Items bought
10	Beer, Nuts, Diapers
20	Beer, Coffee, Diapers
30	Beer, Diapers, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diapers, Eggs, Milk



Itemset:

A set of one or more items.
k-itemset $X = \{x_1, x_2, \ldots, x_k\}$.

(Absolute) Support, or support count of X:

Frequency or occurrence of X.

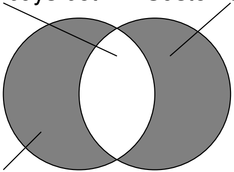
(Relative) Support s:

The fraction of the transactions that contain X.
I.e. the **probability** that a transaction contains X.

An itemset X is frequent, if X's support is no less than a `min_sup` threshold.

An example

TID	Items bought
10	Beer, Nuts, Diapers
20	Beer, Coffee, Diapers
30	Beer, Diapers, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diapers, Eggs, Milk



Find all the rules $X \rightarrow Y$ with minimum support and confidence.

Support s: probability that a transaction contains $X \cup Y$.

Confidence c: conditional probability that a transaction having X also contains Y.

Example:

Let min_sup $= 50\%$ and min_conf $= 50\%$.

Frequent itemsets:

Beer: 3, Nuts: 3,
Diapers: 4, Eggs: 3,
$\{$Beer, Diapers$\}$: 3.

Association rules:

Beer $\rightarrow$ Diapers (60%, 100%).
Diapers $\rightarrow$ Beer (60%, 75%).

Basic concepts: association rules (2)

Implication of the form $A \rightarrow B$:

where $A \neq \emptyset$, $B \neq \emptyset$ and $A \cap B = \emptyset$.

Strong rule:

Satisfies both min_sup and min_conf

$$\text{support}(A \rightarrow B) = P(A \cup B), \tag{1}$$

$$\text{confidence}(A \rightarrow B) = P(B|A) \tag{2}$$

$$= \frac{\text{support}(A \cup B)}{\text{support}(A)} \tag{3}$$

$$= \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}. \tag{4}$$

I.e. confidence of rule can be easily derived from the support counts of A and $A \cup B$.

Association-rule mining:

Find all frequent itemsets.

Generate strong association rules from the frequent itemsets.

Closed itemsets and max-itemsets

A long itemset contains a combinatorial number of sub-itemsets.

E.g. $\{a_1, a_2, \ldots, a_{100}\}$ contains

$$\binom{100}{1} + \binom{100}{2} + \cdots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 \cdot 10^{30} \text{ sub-itemsets!} \tag{5}$$

Solution:

Mine closed itemsets and max-itemsets instead.

An itemset X is closed, if X is frequent and there exists no super-itemset $X \subset Y$ with the same support as X.

Proposed by (Pasquier et al., ICDT'99).

An itemset X is a max-itemset, if X is frequent and there exists no frequent super-itemset $X \subset Y$.

Proposed by (Bayardo, SIGMOD'98).

Closed itemset is a lossless "compression" of frequent itemsets.

Reducing the number of itemsets (and rules).

Closed itemsets and max-itemsets (II)

Example:

DB $= \{\langle a_1, a_2, \ldots, a_{100}\rangle, \langle a_1, a_2, \ldots, a_{100}\rangle\}$.
I.e. just two transactions.
min_sup $= 1$.

What are the closed itemsets?

$\langle a_1, a_2, \ldots, a_{100}\rangle$: 1,
$\langle a_1, a_2, \ldots, a_{50}\rangle$: 2,
Number behind the colon: support_count.

What are the max-itemsets?

$\langle a_1, a_2, \ldots, a_{100}\rangle$: 1.

What is the set of all frequent itemsets?

Chapter V: Mining frequent patterns, associations and correlations

- Basic Concepts.
- **Scalable frequent-itemset-mining methods.**
  - **Apriori: a candidate-generation-and-test approach.**
  - Improving the efficiency of apriori.
  - FPGrowth: a frequent-pattern-growth approach.
  - ECLAT: frequent-pattern mining with vertical data format.
  - Mining closed itemsets and max-itemsets.
- Generating association rules from frequent itemsets.
- Which patterns are interesting? Pattern-evaluation methods.
- Summary.

The downward-closure property and scalable mining methods

The downward-closure property of frequent patterns:

Any subset of a frequent itemset must also be frequent.

If $\{$Beer, Diapers, Nuts$\}$ is frequent, so is $\{$Beer, Diapers$\}$.

I.e. every transaction having $\{$Beer, Diapers, Nuts$\}$ also contains $\{$Beer, Diapers$\}$.

Scalable mining methods: three major approaches.

A priori (Agrawal & Srikant, VLDB'94).

Frequent-pattern growth (FPgrowth) (Han, Pei & Yin, SIGMOD'00).

Vertical-data-format approach (CHARM) (Zaki & Hsiao, SDM'02).

A priori: a candidate generation & test approach

A priori pruning principle:

If there is any itemset which is infrequent,
its supersets should not be generated/tested!
(Agrawal & Srikant, VLDB'94; Mannila et al., KDD'94)

Method:

Initially, scan DB once to get frequent 1-itemsets.
Generate length-$(k+1)$ candidate itemsets from length-k frequent itemsets.
Test the candidates against DB, discard those that are infrequent.
Terminate when no further candidate or frequent itemset can be generated.

A priori algorithm - an example

Database TBD

TID	Items
10	A,C,D
20	B,C,E
30	A,B,C,E
40	B,E

min_sup = 2

1st scan → C_1

Itemset	sup
$\{A\}$	2
$\{B\}$	3
$\{C\}$	3
$\{D\}$	1
$\{E\}$	3

L_1

Itemset	sup
$\{A\}$	2
$\{B\}$	3
$\{C\}$	3
$\{E\}$	3

C_2

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

2nd scan → C_2

Itemset	sup
$\{A, B\}$	1
$\{A, C\}$	2
$\{A, E\}$	1
$\{B, C\}$	2
$\{B, E\}$	3
$\{C, E\}$	2

L_2

Itemset	sup
$\{A, C\}$	2
$\{B, C\}$	2
$\{B, E\}$	3
$\{C, E\}$	2

C_3

Itemset
$\{B, C, E\}$

3rd scan → L_3

Itemset	sup
$\{B, C, E\}$	2

A priori algorithm (pseudo code)

C_k: candidate itemsets of size k
L_k: frequent itemsets of size k

$L_1 = \{$frequent items$\}$;

for $(k = 1; L_k \neq \emptyset; k{+}{+})$ **do begin**
    $C_{k+1} =$ candidates generated from L_k;
    for each transaction t in database **do**
        increment the count of all candidates in C_{k+1} that are contained in t;
    $L_{k+1} =$ candidates in C_{k+1} with min_sup;
end;
return $\bigcup_k L_k$;

Implementation of a priori

How to generate candidates?

Step 1: self-joining L_k (or joining L_k with L_1).
Step 2: pruning.

Example of candidate generation:

$L_3 = \{abc, abd, acd, ace, bcd\}$.
Self-joining: $L_3 \bowtie L_3$:

abcd from *abc* and *abd*.
acde from *acd* and *ace*.

Pruning:

acde is removed because *ade* is not in L_3.

$C_4 = \{abcd\}$.

Implementation of a priori

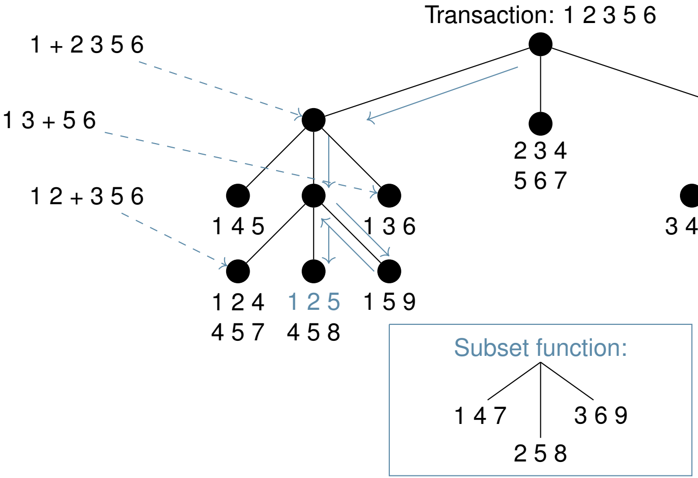
Why is counting supports of candidates a problem?

- The total number of candidates can be huge.
- One transaction may contain many candidates.

Method:

- Candidate itemsets are stored in a **hash tree**.
- Leaf node of hash tree contains a list of itemsets and counts.
- Interior node contains a hash table.
- Subset function: finds all the candidates contained in a transaction.

Counting supports of candidates using hash tree



Candidate generation: an SQL implementation

Chapter V: Mining frequent patterns, associations and correlations

- Basic Concepts.
- Scalable frequent-itemset-mining methods.
  - Apriori: a candidate-generation-and-test approach.
  - **Improving the efficiency of apriori.**
  - FPGrowth: a frequent-pattern-growth approach.
  - ECLAT: frequent-pattern mining with vertical data format.
  - Mining closed itemsets and max-itemsets.
- Generating association rules from frequent itemsets.
- Which patterns are interesting? Pattern-evaluation methods.
- Summary.

References: Basic concepts of frequent-pattern mining

(Association Rules)

R. Agrawal, T. Imielinski, and A. Swami: Mining association rules between sets of items in large databases. SIGMOD'93.

(Max-Itemset)

(Max-Itemset) R. J. Bayardo: Efficiently mining long patterns from databases. SIGMOD'98.

(Closed Itemsets)

N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal: Discovering frequent closed itemsets for association rules. ICDT'99.

(Sequential Pattern)

R. Agrawal and R. Srikant: Mining sequential patterns. ICDE'95.

References: Apriori and its improvements

R. Agrawal and R. Srikant: Fast algorithms for mining association rules. VLDB'94.

H. Mannila, H. Toivonen, and A. I. Verkamo: Efficient algorithms for discovering association rules. KDD'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.

H. Toivonen: Sampling large databases for association rules. VLDB'96.

S. Brin, R. Motwani, J. D. Ullman, and S. Tsur: Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97.

S. Sarawagi, S. Thomas, and R. Agrawal: Integrating association rule mining with relational database systems: alternatives and implications. SIGMOD'98.

References: Depth-first, projection-based FP mining

R. Agarwal, C. Aggarwal, and V. V. V. Prasad: A tree projection algorithm for generation of frequent itemsets. J. Parallel and Distributed Computing, 2002.

G. Grahne and J. Zhu: Efficiently Using Prefix-Trees in Mining Frequent Itemsets. FIMI'03.

B. Goethals and M. Zaki: An introduction to workshop on frequent itemset mining implementations. FIMI'03.

J. Han, J. Pei, and Y. Yin: Mining frequent patterns without candidate generation. SIGMOD'00.

J. Liu, Y. Pan, K. Wang, and J. Han: Mining frequent itemsets by opportunistic projection. KDD'02.

J. Han, J. Wang, Y. Lu, and P. Tzvetkov: Mining top-*k* frequent closed patterns without minimum support. ICDM'02.

J. Wang, J. Han, and J. Pei. CLOSET+: Searching for the best strategies for mining frequent closed itemsets. KDD'03.

References: Vertical format and row enumeration methods

M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li: Parallel algorithm for discovery of association rules. DAMI'97.

M. J. Zaki and C. J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. SDM'02.

C. Bucila, J. Gehrke, D. Kifer, and W. White. DualMiner: A dual-pruning algorithm for itemsets with constraints. KDD'02.

F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. Zaki. CARPENTER: Finding closed patterns in long biological datasets. KDD'03.

H. Liu, J. Han, D. Xin, and Z. Shao: Mining interesting patterns from very high dimensional data: a top-down row enumeration approach. SDM'06.

References: Mining correlations and interesting rules

S. Brin, R. Motwani, and C. Silverstein: Beyond market basket: generalizing association rules to correlations. SIGMOD'.

M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo: Finding interesting rules from large sets of discovered association rules. CIKM'94.

R. J. Hilderman and H. J. Hamilton: Knowledge Discovery and Measures of Interest. Kluwer Academic, 2001.

C. Silverstein, S. Brin, R. Motwani, and J. Ullman: Scalable techniques for mining causal structures. VLDB'98.

P.-N. Tan, V. Kumar, and J. Srivastava: Selecting the right interestingness measure for association patterns. KDD'02.

E. Omiecinski: Alternative interest measures for mining associations. TKDE'03.

T. Wu, Y. Chen and J. Han: Association mining in large databases: a re-examination of its measures. PKDD'07.

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.

Thank you for your attention.
Any questions about the fifth chapter?

Ask them now, or again, drop me a line:
luciano.melodia@fau.de.