Queen Mary
University of London

ECS766P Data Mining

Week 8: Association analysis

Emmanouil Benetos
emmanouil.benetos@qmul.ac.uk

November 2021

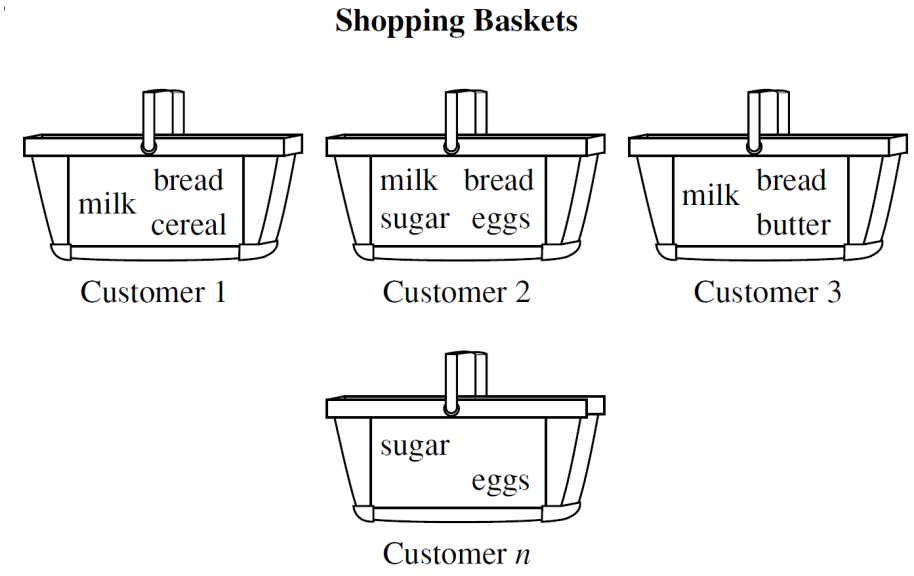
School of EECS, Queen Mary University of London

Week 6: Classification and Clustering

1. Classification
    - Classification task
    - K-nearest neighbours classifier
    - Model evaluation
    - Model selection
    - Common classifiers
2. Clustering
    - Clustering task
    - K-means clustering
    - Model evaluation and selection
    - Agglomerative hierarchical clustering

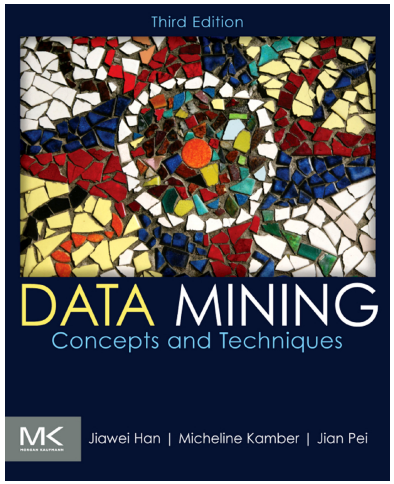
This week's lecture

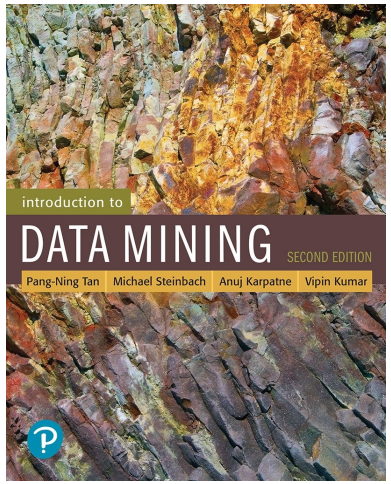
1. Frequent itemsets - basic concepts
2. Frequent itemset mining methods
3. Association rule mining
4. Pattern evaluation methods



Reading

- **Chapter 6 of J. Han, M. Kamber, J. Pei, "Data Mining: Concepts and Techniques", 3rd edition, Elsevier/Morgan Kaufmann, 2012**
- Chapter 5 of P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar, "Introduction to Data Mining", 2nd edition, Pearson, 2019





Frequent itemsets - Basic Concepts

Motivation: online shopping

Frequently bought together

+

Total price: £76.42

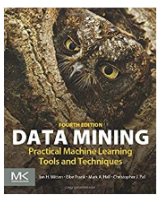
Add both to Basket

This item: Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) by Jiawei Han Hardcover £44.99

Data Mining: Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in Data… by Ian H. Witten Paperback £31.43

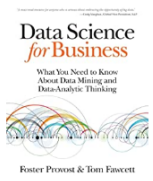
Customers who viewed this item also viewed

Page 1 of 6

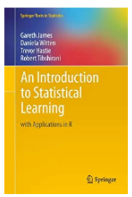


Data Mining: Practical Machine Learning Tools and Techniques (Morgan…
› Ian H. Witten
36
Paperback
£31.43

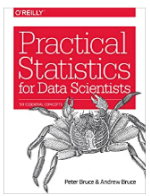
Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data…
› Jiawei Han
14
Hardcover
11 offers from £8.24



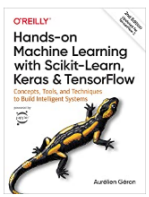
Data Science for Business: What you need to know about data mining and…
› Foster Provost
311
Paperback
£25.41



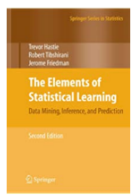
An Introduction to Statistical Learning: with Applications in R…
› Gareth James
522
Hardcover
£53.99



Practical Statistics for Data Scientists
› Peter Bruce
183
Paperback
£39.99



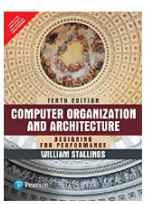
Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow:…
Aurelien Geron
404
Paperback
£35.91



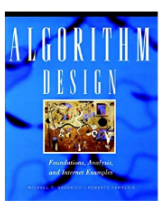
The Elements of Statistical Learning (Springer Series in Statistics)
Trevor Hastie
318
Hardcover
£53.95

Customers who bought this item also bought

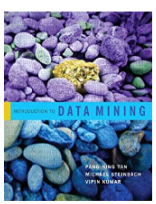
Page 1 of 12



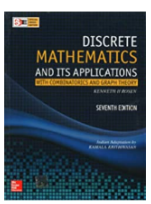
Computer Organization And Architecture
WILLIAM…
34
Paperback
21 offers from £15.85



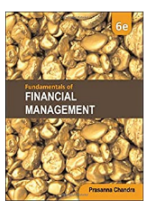
Algorithm Design: Foundations, Analysis, and Internet Examples
› Michael T. Goodrich
30
Paperback
9 offers from £46.43



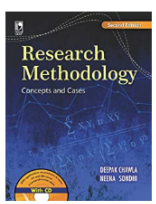
Introduction to Data Mining
› Pang-Ning Tan
108
Hardcover
7 offers from £49.99



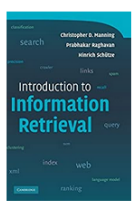
Discrete Mathematics and it's Applications
ROSEN
136
Paperback
11 offers from £10.46



Fundamentals Of Financial Management, 6Ed
CHANDRA
22
Paperback
5 offers from £15.81



Research Methodology [Paperback] [Jan 01, 2016] Deepak Chawla Neena Sondhi
16
Paperback
2 offers from £34.46



Introduction to Information Retrieval
Hinrich Manning
66
Unknown Binding
9 offers from £67.02

Motivation

- catalog design, or planning shelf-space (frequently co-purchased items can be placed near each other to increase their sales, or alternatively, placing them at opposite ends of the store to entice customers to pick other profitable items along the way.
- targeted advertisement (“based on your search/purchase history/items in your cart, you might also be interested in ...”)
- cross-marketing (instead of putting product A on sale, it may be more profitable to put product B on sale which also increases the sales of product A if they are frequently bought together),

Motivation

- analysing logs of web visits or click-streams (which sequence of pages do users frequently visit)
- warehouse management/mail orders
- maintenance (which parts break-down together, or which equipment/spares are typically required together, say before sending a dispatch)
- generating FAQ for customer service
- autocomplete suggestions

Transaction dataset

- Consider a set of items $\mathcal{I} = \{1, 2, \ldots, m\}$
- A subset $\mathcal{T} \subseteq \mathcal{I}$ is called an itemset
- A transaction dataset $\mathcal{D} = \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$ is a sequence of itemsets called transactions
- A k-itemset contains k items

TID	Items
1	Bread, eggs, milk
2	Juice, bread
3	Juice, eggs, butter, milk
4	Juice, bread, butter, milk
5	Eggs, butter, milk

Support count

- The support count $N_{\mathcal{A}}$ of an itemset $\mathcal{A} \subseteq \mathcal{I}$ is the number of transactions in a transaction dataset $\mathcal{D}$ that contain $\mathcal{A}$

$$N_{\mathcal{A}} = \sum_{i=1}^{N} \mathbb{I}\left[\mathcal{A} \subseteq \mathcal{T}_i\right],$$

  where $\mathbb{I}[e] = 1$ if e is true, and $\mathbb{I}[e] = 0$ otherwise
- This is also known as the occurrence frequency of an itemset.

Support

- The support $S_{\mathcal{A}}$ of an itemset $\mathcal{A} \subseteq \mathcal{I}$ is the fraction of transactions in a transaction dataset $\mathcal{D}$ that contain $\mathcal{A}$

$$S_{\mathcal{A}} = \frac{N_{\mathcal{A}}}{N} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\mathcal{A} \subseteq \mathcal{T}_i\right],$$

  where $\mathbb{I}[e] = 1$ if e is true, and $\mathbb{I}[e] = 0$ otherwise
- The support can also be interpreted as the probability that a transaction drawn from $\mathcal{D}$ contains $\mathcal{A}$.

Frequent pattern mining

- If the support $S_{\mathcal{A}}$ of an itemset $\mathcal{A} \subseteq \mathcal{I}$ is equal or above some predefined minimum support threshold $\tau_S \geq 0$, then $\mathcal{A}$ is considered a frequent itemset.
- Frequent pattern mining is the process of finding and analysing frequent patterns (such as frequent itemsets).
- A major challenge in mining frequent itemsets from a large data set is the fact that such mining often generates a huge number of itemsets satisfying the minimum support threshold.

Support of subsets

- Consider an itemset $\mathcal{B} \subseteq \mathcal{I}$ and an itemset $\mathcal{A} \subseteq \mathcal{B}$, and recall that

$$N_{\mathcal{B}} = \sum_{i=1}^{N} \mathbb{I}\left[\mathcal{B} \subseteq \mathcal{T}_i\right]$$

- Because $\mathcal{A} \subseteq \mathcal{T}_i$ whenever $\mathcal{B} \subseteq \mathcal{T}_i$, it can be shown that $N_{\mathcal{A}} \geq N_{\mathcal{B}}$ and $S_{\mathcal{A}} \geq S_{\mathcal{B}}$.

Frequent Itemset Mining Methods

Apriori Algorithm - Introduction

- Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets.
- The name of the algorithm is based on the fact that the algorithm uses **prior knowledge** of frequent itemset properties.
- Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore $(k+1)$-itemsets.
- To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property is used to reduce the search space.

Apriori property

Apriori property

All nonempty subsets of a frequent itemset must also be frequent.

- Consider an itemset $\mathcal{B} \subseteq \mathcal{I}$ and an itemset $\mathcal{A} \subseteq \mathcal{B}$
- Recall that $\mathcal{B}$ is frequent if $S_{\mathcal{B}} \geq \tau_S$ for some predefined threshold $\tau_S \geq 0$
- If $\mathcal{B}$ is frequent, then $\mathcal{A}$ is also frequent, since $S_{\mathcal{A}} \geq S_{\mathcal{B}} \geq \tau_S$
- Similarly, if $\mathcal{A}$ is not frequent, then $\mathcal{B}$ is also not frequent, since $S_{\mathcal{B}} \leq S_{\mathcal{A}} \leq \tau_S$
- Note that the itemset $\mathcal{B}$ has exactly $2^{|\mathcal{B}|}$ subsets, since each element of $\mathcal{B}$ may or may not be present in each of its subsets

Apriori algorithm - basic process

- The set of frequent k-itemsets is denoted by L_k
- The set of frequent itemsets L is given by

$$L = \bigcup_{k=1}^{m} L_k$$

- The Apriori algorithm finds the sequence of frequent k-itemsets $L_1, L_2, \ldots, L_m$
- Apriori starts by finding the set of frequent 1-itemsets L_1
- Subsequently, it uses L_{k-1} to find L_k for every $k \geq 2$

Finding L_1

- In order to find the set of frequent 1-itemsets L_1, the Apriori algorithm computes the frequency of each item $j \in \mathcal{I}$:

$$L_1 = \left\{ \{j\} \subseteq \mathcal{I} \mid \frac{N_{\{j\}}}{N} = S_{\{j\}} \geq \tau_S \right\}$$

- For $k \geq 2$, every frequent k-itemset must be a superset of an itemset in L_1.

Finding L_k given L_{k-1}

- The process of finding L_k given L_{k-1} can be divided into two steps:
  1. The join step generates a set of candidates C_k from L_{k-1}
  2. The prune step eliminates candidates from C_k to derive L_k

Join operation

- Consider two itemsets $\mathcal{A}_1 \subseteq \mathcal{I}$ and $\mathcal{A}_2 \subseteq \mathcal{I}$, each with $k-1$ items
- Let $\{l_1[1], l_1[2], \ldots, l_1[k-1]\}$ denote the result of **sorting** the items in itemset $\mathcal{A}_1$ in lexicographic order.
- Let $\{l_2[1], l_2[2], \ldots, l_2[k-1]\}$ denote the result of **sorting** the items in itemset $\mathcal{A}_2$ in lexicographic order.
- $\mathcal{A}_1$ and $\mathcal{A}_2$ are joinable if $l_1[j] = l_2[j]$ for every $j < k-1$ and $l_1[k-1] < l_2[k-1]$
- If $\mathcal{A}_1$ and $\mathcal{A}_2$ are joinable, then $\mathcal{A}_1 \bowtie \mathcal{A}_2$ is given by

$$\mathcal{A}_1 \bowtie \mathcal{A}_2 = \{l_1[1], l_1[2], \ldots, l_1[k-1], l_2[k-1]\}$$

- In other words, **if $\mathcal{A}_1$ and $\mathcal{A}_2$ are joinable, then $\mathcal{A}_1 \bowtie \mathcal{A}_2$ simply contains $\mathcal{A}_1$ and the *last* element of $\mathcal{A}_2$**

Finding L_k given L_{k-1}: join step

- The join step obtains the set of candidates C_k by **joining every joinable pair** of itemsets from L_{k-1}:

$$C_k = \{\mathcal{A}_1 \bowtie \mathcal{A}_2 \mid \mathcal{A}_1 \in L_{k-1}, \mathcal{A}_2 \in L_{k-1}, \ \mathcal{A}_1 \text{ and } \mathcal{A}_2 \text{ are joinable } \}$$

- The Apriori property guarantees that $L_k \subseteq C_k$: every frequent k-itemset is in C_k, although some elements of C_k may not be frequent.

Finding L_k given L_{k-1}: prune step

- The prune step reduces the size of C_k by eliminating candidate itemsets.
- The set of candidates C_k is **pruned** in two stages:
  1. Consider an itemset $\mathcal{B} \in C_k$. If a subset $\mathcal{A} \subseteq \mathcal{B}$ with $k-1$ elements is not in L_{k-1}, then $\mathcal{B}$ is not frequent.
  2. Consider an itemset $\mathcal{B} \in C_k$. If $S_{\mathcal{B}} = N_{\mathcal{B}}/N < \tau_S$, then $\mathcal{B}$ is not frequent.
- After the second stage, all infrequent itemsets from C_k can be eliminated to derive L_k

Apriori algorithm: example (1/4)

- Consider the transaction dataset $\mathcal{D}$ presented below. Suppose that the minimum support count required is 2.
- Since there are 9 transactions in the dataset, the support threshold $\tau_S = 2/9 = 22.\bar{2}\%$.

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Apriori algorithm: example (2/4)

Scan D for count of each candidate

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count

L_2

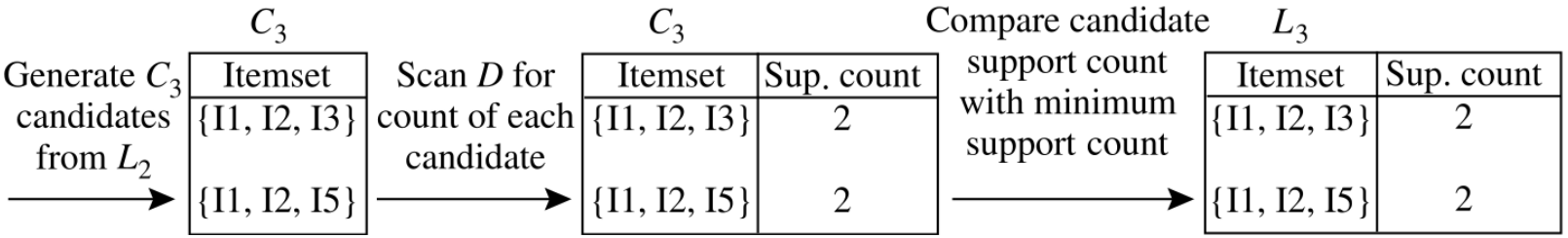
Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Apriori algorithm: example (3/4)

- The set of candidates C_3 is obtained by joining every joinable pair of itemsets from L_2:

$$C_3 = \{\{1,2,3\},\{1,2,5\},\{1,3,5\},\{2,3,4\},\{2,3,5\},\{2,4,5\}\}$$

- However, only $\{1,2,3\}$ and $\{1,2,5\}$ have all their 2-*subsets* in L_2 and are kept after pruning.



Apriori algorithm: example (4/4)

- The set of candidates C_4 is obtained by joining every joinable pair of itemsets from L_3:

$$C_4 = \{\{1, 2, 3, 5\}\}$$

- However, the 3-itemset $\{2, 3, 5\}$ is not frequent, since $\{2, 3, 5\} \notin L_3$. The Apriori property guarantees that $\{1, 2, 3, 5\}$ is also not frequent
- Therefore, $C_4 = \varnothing$ and $L_4 = \varnothing$.
- Thus, the algorithm terminates, having found all frequent itemsets.

Apriori algorithm (1/2)

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- D, a database of transactions;
- *min_sup*, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

(1) L_1 = find_frequent_1-itemsets(D);
(2) **for** ($k = 2; L_{k-1} \neq \phi; k$++) {
(3) C_k = apriori_gen(L_{k-1});
(4) **for each** transaction $t \in D$ { // scan D for counts
(5) C_t = subset(C_k, t); // get the subsets of t that are candidates
(6) **for each** candidate $c \in C_t$
(7) c.count++;
(8) }
(9) $L_k = \{c \in C_k | c.count \geq min\_sup\}$
(10) }
(11) **return** $L = \cup_k L_k$;

Apriori algorithm (2/2)

```
procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
(1)     for each itemset l_1 ∈ L_{k-1}
(2)         for each itemset l_2 ∈ L_{k-1}
(3)             if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2])
                    ∧... ∧ (l_1[k-2] = l_2[k-2]) ∧ (l_1[k-1] < l_2[k-1]) then {
(4)                 c = l_1 ⋈ l_2; // join step: generate candidates
(5)                 if has_infrequent_subset(c, L_{k-1}) then
(6)                     delete c; // prune step: remove unfruitful candidate
(7)                 else add c to C_k;
(8)             }
(9)     return C_k;

procedure has_infrequent_subset(c: candidate k-itemset;
          L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
(1)     for each (k-1)-subset s of c
(2)         if s ∉ L_{k-1} then
(3)             return TRUE;
(4)     return FALSE;
```

Frequent pattern analysis

- The set of frequent itemsets $L = \cup_k L_k$ may be too large to allow convenient analysis.
- An itemset $\mathcal{A} \subseteq \mathcal{I}$ is closed in a transaction dataset $\mathcal{D}$ if there is no itemset $\mathcal{B} \subseteq \mathcal{I}$ such that $\mathcal{A} \subseteq \mathcal{B}$, $\mathcal{A} \neq \mathcal{B}$, and $S_{\mathcal{A}} = S_{\mathcal{B}}$.
- Frequent pattern mining may be simplified by considering closed frequent itemsets.
- The Apriori algorithm can be improved by additional data structures and heuristics.
- There are other algorithms for frequent itemset mining. Some of them are capable of finding all closed frequent itemsets without first finding the set of frequent itemsets L.

Association rule mining

Association rules - Introduction

- In **market basket analysis**, we are not only interested in finding groups of frequent items, but on finding associations between different items.
- These patterns can be represented in the form of association rules.
- For example, the information that customers who purchase computers also tend to buy antivirus software at the same time can be represented by the following association rule:

$$\text{computer} \Rightarrow \text{antivirus software} \; [\text{support}{=}2\%, \text{confidence}{=}60\%]$$

where rule support and confidence are two measures of rule interestingness.

Association rules

- Consider a set of items $\mathcal{I} = \{1, 2, \ldots, m\}$
- A subset $\mathcal{T} \subseteq \mathcal{I}$ is called an itemset
- A transaction dataset $\mathcal{D} = \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$ is a sequence of itemsets called transactions
- An association rule between an itemset $\mathcal{A} \subseteq \mathcal{I}$ and an itemset $\mathcal{B} \subseteq \mathcal{I}$ such that $\mathcal{A} \cap \mathcal{B} = \varnothing$ is written as

$$\mathcal{A} \Rightarrow \mathcal{B}$$

and can be interpreted as stating that $\mathcal{A} \subseteq \mathcal{T}$ **frequently** implies that $\mathcal{B} \subseteq \mathcal{T}$ for any transaction $\mathcal{T}$ in the transaction dataset $\mathcal{D}$

Association rules: support

- The support $S_{\mathcal{A} \Rightarrow \mathcal{B}}$ of an association rule $\mathcal{A} \Rightarrow \mathcal{B}$ is equal to the support $S_{\mathcal{A} \cup \mathcal{B}}$ of the itemset $\mathcal{A} \cup \mathcal{B}$

$$S_{\mathcal{A} \Rightarrow \mathcal{B}} = S_{\mathcal{A} \cup \mathcal{B}} = \frac{N_{\mathcal{A} \cup \mathcal{B}}}{N} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[(\mathcal{A} \cup \mathcal{B}) \subseteq \mathcal{T}_i\right],$$

  where $\mathbb{I}[e] = 1$ if e is true, and $\mathbb{I}[e] = 0$ otherwise

- The support $S_{\mathcal{A} \Rightarrow \mathcal{B}}$ of an association rule $\mathcal{A} \Rightarrow \mathcal{B}$ can be interpreted as the probability that a transaction drawn from $\mathcal{D}$ contains $\mathcal{A} \cup \mathcal{B}$

Association rules: confidence

- The confidence $V_{\mathcal{A}\Rightarrow\mathcal{B}}$ of an association rule $\mathcal{A} \Rightarrow \mathcal{B}$ is given by

$$V_{\mathcal{A}\Rightarrow\mathcal{B}} = \frac{S_{\mathcal{A}\cup\mathcal{B}}}{S_{\mathcal{A}}} = \frac{N_{\mathcal{A}\cup\mathcal{B}}}{N_{\mathcal{A}}} = \frac{\sum_{i=1}^{N} \mathbb{I}\left[(\mathcal{A}\cup\mathcal{B}) \subseteq \mathcal{T}_i\right]}{\sum_{i=1}^{N} \mathbb{I}\left[\mathcal{A} \subseteq \mathcal{T}_i\right]},$$

  where $\mathbb{I}[e] = 1$ if e is true, and $\mathbb{I}[e] = 0$ otherwise

- The confidence $V_{\mathcal{A}\Rightarrow\mathcal{B}}$ of an association rule $\mathcal{A} \Rightarrow \mathcal{B}$ can be interpreted as the probability that a transaction $\mathcal{T}$ drawn from $\mathcal{D}$ contains $\mathcal{B}$ given that $\mathcal{T}$ contains $\mathcal{A}$

Association rule mining

- An association rule $\mathcal{A} \Rightarrow \mathcal{B}$ is called strong if $S_{\mathcal{A} \Rightarrow \mathcal{B}} \geq \tau_S$ and $V_{\mathcal{A} \Rightarrow \mathcal{B}} \geq \tau_V$, where $\tau_S > 0$ is a support threshold and $\tau_V \geq 0$ is a confidence threshold.
- Association rule mining is the process of finding and analysing strong association rules.
- In general, association rule mining can be viewed as a two-step process:
  1. Find all frequent itemsets
  2. Generate strong association rules from the frequent itemsets

Generating association rules from frequent itemsets

- If an association rule $\mathcal{A} \Rightarrow \mathcal{B}$ is strong for a given support threshold τ_S, then $\mathcal{A} \cup \mathcal{B}$ is a frequent itemset for the same threshold τ_S.
- Therefore, only frequent itemsets need to be considered in order to derive strong association rules.
- Consider the set of frequent itemsets $L = \cup_k L_k$ for a given dataset $\mathcal{D}$ and support threshold τ_S, which may be obtained by the Apriori algorithm.
- Generating association rules: For every non-empty subset $\mathcal{A} \subseteq \mathcal{B}$ of a frequent itemset $\mathcal{B} \in L$, the association rule

$$\mathcal{A} \Rightarrow (\mathcal{B} - \mathcal{A})$$

is strong if $V_{\mathcal{A} \Rightarrow (\mathcal{B} - \mathcal{A})} = \frac{N_{\mathcal{B}}}{N_{\mathcal{A}}} \geq \tau_V$.

Generating association rules: example (1/2)

- Consider the transaction dataset $\mathcal{D}$ presented below and a support threshold $\tau_S = 2/9 = 22.\bar{2}\%$

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Generating association rules: example (2/2)

- Using the Apriori algorithm, we found that $\{1, 2, 5\} \in L_3$ is a frequent itemset.
- The nonempty subsets are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}.
- The following association rules can be derived based on this frequent itemset:

$$
\begin{aligned}
\{\mathrm{I1}, \mathrm{I2}\} &\Rightarrow \mathrm{I5}, && \mathit{confidence} = 2/4 = 50\% \\
\{\mathrm{I1}, \mathrm{I5}\} &\Rightarrow \mathrm{I2}, && \mathit{confidence} = 2/2 = 100\% \\
\{\mathrm{I2}, \mathrm{I5}\} &\Rightarrow \mathrm{I1}, && \mathit{confidence} = 2/2 = 100\% \\
\mathrm{I1} &\Rightarrow \{\mathrm{I2}, \mathrm{I5}\}, && \mathit{confidence} = 2/6 = 33\% \\
\mathrm{I2} &\Rightarrow \{\mathrm{I1}, \mathrm{I5}\}, && \mathit{confidence} = 2/7 = 29\% \\
\mathrm{I5} &\Rightarrow \{\mathrm{I1}, \mathrm{I2}\}, && \mathit{confidence} = 2/2 = 100\%
\end{aligned}
$$

- For a confidence threshold $\tau_V = 70\%$, only three of these association rules are considered strong.

Pattern evaluation methods

Interesting association rules

- A strong association rule may or may not be interesting for a specific application.
- Some strong association rules can even be misleading about a relationship.
- Several measures have been developed to help evaluate association rules.

Misleading strong association rules: example

- Consider a transaction dataset $\mathcal{D}$ that contains 1000 transactions from a store
- Suppose that 600 transactions contain *games*, and that 750 contain *videos*
- Furthermore, suppose that 400 transactions contain both *games* and *videos*
- The association rule $\{\text{games}\} \Rightarrow \{\text{videos}\}$ has support $40\% = 400/1000$ and confidence $400/600 = 66.\bar{6}\%$
- This rule would be strong for $\tau_S = 40\%$ and $\tau_V = 66.\bar{6}\%$
- However, the probability that a transaction drawn from $\mathcal{D}$ contains videos is $750/1000 = 75\%$, which is higher than the confidence for the rule
- In fact, games and videos are **negatively associated** because the purchase of one of these items actually decreases the likelihood of purchasing the other.
- This shows that a rule $\mathcal{A} \Rightarrow \mathcal{B}$ can be considered strong even if $\mathcal{A}$ and $\mathcal{B}$ are negatively associated

Kulczynski measure

- The Kulczynski measure $K_{\mathcal{A},\mathcal{B}} \in [0,1]$ of the itemsets $\mathcal{A} \subseteq \mathcal{I}$ and $\mathcal{B} \subseteq \mathcal{I}$ such that $\mathcal{A} \cap \mathcal{B} = \varnothing$ is given by

$$K_{\mathcal{A},\mathcal{B}} = \frac{V_{\mathcal{A}\Rightarrow\mathcal{B}} + V_{\mathcal{B}\Rightarrow\mathcal{A}}}{2}$$

- The Kulczynski measure $K_{\mathcal{A},\mathcal{B}}$ can be interpreted as the average between the confidence that $\mathcal{A} \Rightarrow \mathcal{B}$ and the confidence that $\mathcal{B} \Rightarrow \mathcal{A}$
- If $K_{\mathcal{A},\mathcal{B}} = 0$, then $\mathcal{A} \subseteq \mathcal{T}$ implies that $\mathcal{B} \not\subseteq \mathcal{T}$ for any transaction $\mathcal{T}$
- If $K_{\mathcal{A},\mathcal{B}} = 1$, then $\mathcal{A} \subseteq \mathcal{T}$ implies that $\mathcal{B} \subseteq \mathcal{T}$ for any transaction $\mathcal{T}$
- Note that the Kulczynski measure is symmetric: $K_{\mathcal{A},\mathcal{B}} = K_{\mathcal{B},\mathcal{A}}$

Imbalance ratio

- The imbalance ratio $I_{\mathcal{A},\mathcal{B}} \in [0, 1]$ of the itemsets $\mathcal{A} \subseteq \mathcal{I}$ and $\mathcal{B} \subseteq \mathcal{I}$ is given by

$$I_{\mathcal{A},\mathcal{B}} = \frac{|N_{\mathcal{A}} - N_{\mathcal{B}}|}{N_{\mathcal{A}} + N_{\mathcal{B}} - N_{\mathcal{A}\cup\mathcal{B}}}$$

- The imbalance ratio $I_{\mathcal{A},\mathcal{B}}$ can be interpreted as the ratio between the absolute difference between the support count of $\mathcal{A}$ and the support count of $\mathcal{B}$ and the number of transactions that contain $\mathcal{A}$, $\mathcal{B}$, or both $\mathcal{A}$ and $\mathcal{B}$
- If $I_{\mathcal{A},\mathcal{B}} = 0$, then $\mathcal{A}$ and $\mathcal{B}$ have the same support
- If $I_{\mathcal{A},\mathcal{B}} = 1$, then either $\mathcal{A}$ or $\mathcal{B}$ has zero support
- Note that the imbalance ratio is symmetric: $I_{\mathcal{A},\mathcal{B}} = I_{\mathcal{B},\mathcal{A}}$

Summary

- **Frequent itemsets** are sets of items that satisfy a **minimum support threshold**.
- The **Apriori algorithm** is a seminal algorithm for mining frequent itemsets. It explores the **Apriori property** that all nonempty subsets of a frequent itemset must also be frequent.
- **Association rule mining** consists of first finding frequent itemsets, from which strong association rules are generated.
- Not all strong association rules are interesting. Therefore, the support-confidence framework should be augmented with **pattern evaluation measures**.

Questions?
also please use the forum on QM+