Ve572 Lecture 10

Manuel and Jing

UM-SJTU Joint Institute

June 19, 2018

- Association analysis, a.k.a Association rule mining/learning is a method for discovering interesting relations between variables in large databases.
- It is often used by retailers, again your cat and dog, which has a very large database on transactions, known as market basket data, e.g.

TID	Items
1	{ `Bread`, `Milk` }
2	{ `Bread`, `Diapers`, `Beer`, `Eggs` }
3	{ `Milk`, `Diapers`, `Beer`, `Cola` }
4	{ `Bread`, `Milk`, `Diapers`, `Beer` }
5	{ `Bread`, `Milk`, `Diapers`, `Cola` }

- Retailers are interested in analysing the data to learn about the purchasing behaviour of their customers, and make recommendations to their customers.

Q: Can you guess the difference between clustering and association analysis?

- Association analysis is often used in

1. Bioinformatics
2. Medical diagnosis
3. Finance
4. Government monitoring

- We discuss association analysis using market basket data. Let

$$\mathcal{I} = \{z_1, z_2, \ldots, z_p\}$$

be the set of all items in a market basket data, and

$$\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$$

be the set of all transactions, where each transaction

$$t_i$$

has a unique transaction ID, and contains a subset of items from $\mathcal{I}$.

- Imagine we have a very big market basket data, that is, very large p and n in

$$\mathcal{I} = \{z_1, z_2, \ldots, z_p\} \qquad \text{and} \qquad \mathcal{T} = \{t_1, t_2, \ldots, t_n\}$$

Q: How would you make recommendations? That is, how to expand the idea

	Gillette Razors	Christian Louboutin Shoes	Air Jordan Shoes
obs. 1	1	0	1
obs. 2	0	1	0
obs. 1	1	0	1

to slightly more complicated data but vastly big

TID	Bread	Milk	Diapers	Beer	Egg	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

where 1 denotes at least 1 of that item was brought in the ith transaction.

- In association analysis, a subset of $\mathcal{I}$, $\mathcal{X} \subset \mathcal{I}$, is known as an itemset, e.g.

$$\mathcal{A} = \{ \texttt{Milk}, \texttt{Diapers}, \texttt{Beer} \}$$

- The support of an itemset $\mathcal{X}$ is defined as

$$\operatorname{supp}(\mathcal{X}) = \frac{|\{t_i \mid \mathcal{X} \subset t_i, t_i \in \mathcal{T}\}|}{|\mathcal{T}|}$$

where $|\cdot|$ denotes the number of elements in a set, the size of the set.

- For instance, the support of $\mathcal{A}$ is 0.4,

TID	Bread	Milk	Diapers	Beer	Egg	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

- An association rule is an implication of the form

$$\mathcal{X} \implies \mathcal{Y}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are itemsets of $\mathcal{I}$, and are disjoint

$$\mathcal{X} \cap \mathcal{Y} = \varnothing$$

- The confidence of a rule is defined as

$$\mathrm{conf}\left(\mathcal{X} \implies \mathcal{Y}\right) = \frac{\mathrm{supp}\left(\mathcal{X} \cup \mathcal{Y}\right)}{\mathrm{supp}\left(\mathcal{X}\right)}$$

Q: What is the confidence of the rule { `Milk`, `Diapers` } $\implies$ { `Beer` }?

TID	Bread	Milk	Diapers	Beer	Egg	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

- The support of an itemset $\mathcal{X}$ is an indication of how frequently $\mathcal{X}$ in $\mathcal{T}$.

Q: What can we interpret the confidence of a rule

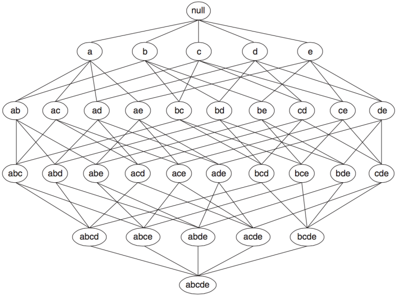
$$\operatorname{conf}(\mathcal{X} \implies \mathcal{Y}) = \frac{\operatorname{supp}(\mathcal{X} \cup \mathcal{Y})}{\operatorname{supp}(\mathcal{X})}$$

- Of course, we wish to discover rules with high support and high confidence.
- Similar to hierarchical clustering in the sense that the number of rules satisfy

$$\operatorname{supp}(\mathcal{X} \cup \mathcal{Y}) \geq s_{min} \qquad \text{and} \qquad \operatorname{conf}(\mathcal{X} \implies \mathcal{Y}) \geq c_{min}$$

  decreases as s_{min} and c_{min} increases, just like the number of clusters and d.
- A choice then can be made regarding the recommendations for customers once we have a set of rules with the highest support and confidence.
- Association rule mining is about finding all rules achieve certain s_{min} or c_{min}
- A brutal force approach is NOT a good idea since it grows really quickly.

- We will have a lot of rules with only 6 items,



- In practice, when apply the support requirement, a lot of rules are out, e.g.

$$\{\ \texttt{Beer}, \texttt{Diapers}\ \} \implies \{\ \texttt{Milk}\ \}, \{\ \texttt{Milk}\ \} \implies \{\ \texttt{Beer}, \texttt{Diapers}\ \},$$
$$\{\ \texttt{Diapers}, \texttt{Milk}\ \} \implies \{\ \texttt{Beer}\ \}, \{\ \texttt{Beer}\ \} \implies \{\ \texttt{Diapers}, \texttt{Milk}\ \},$$
$$\{\ \texttt{Beer}, \texttt{Milk}\ \} \implies \{\ \texttt{Diapers}\ \}, \{\ \texttt{Diapers}\ \} \implies \{\ \texttt{Beer}, \texttt{Milk}\ \},$$

all involve the same $\operatorname{supp}(\mathcal{X} \cup \mathcal{Y})$, where $\mathcal{X} \cup \mathcal{Y} = \{\ \texttt{Milk}, \texttt{Diapers}, \texttt{Beer}\ \}$

- Hence a common strategy used by many association rule mining algorithms is to decompose the problem into the following sequential steps:

1. Frequent Itemset Generation

$$\mathrm{supp}\left(\mathcal{X} \cup \mathcal{Y}\right) \geq s_{min}$$

2. Strong Rule Generation

$$\mathrm{conf}\left(\mathcal{X} \implies \mathcal{Y}\right) \geq c_{min}$$

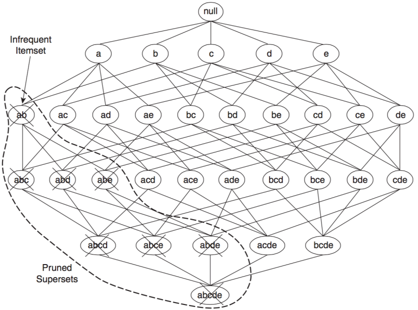
- Notice the definition of support

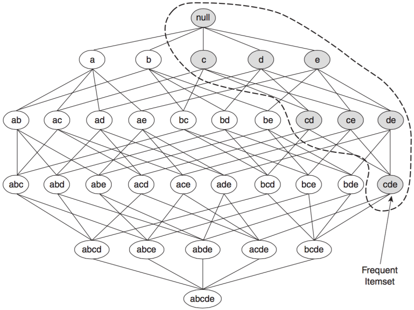
$$\mathrm{supp}\left(\mathcal{X}\right) = \frac{\left|\{t_i \mid \mathcal{X} \subset t_i, t_i \in \mathcal{T}\}\right|}{|\mathcal{T}|}$$

ensures the following key inequality is true

$$\mathrm{supp}\left(\mathcal{X} \cup \mathcal{Y}\right) \leq \mathrm{supp}\left(\mathcal{X}\right)$$

- This means if $\{a, b\}$ is infrequent, then all the supersets are infrequent,





and if $\{c, d, e\}$ is frequent, then all the subsets are frequent.

1. This leads to the sub-steps in the Frequent Itemset Generation step in the

Apriori algorithm

- Roughly, this step consists of two iterative sub-steps:

(a) Candidate generation

(b) Candidate pruning

- To give a high-level illustration, consider the following small dataset again

TID	Bread	Milk	Diapers	Beer	Egg	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

- Note we want to compute/check the support, however $|\mathcal{T}|$ is a constant and

$$\sigma(\mathcal{A}) = |\mathcal{T}| \cdot \mathrm{supp}(\mathcal{A}) = |\{t_i \mid \mathcal{A} \subset t_i, t_i \in \mathcal{T}| = \text{counts}$$

thus it is just a matter of converting this table in wide form into a long form.

(a) Candidate generation:

Generate all itemset $\mathcal{A}$ of size 1, that is $|\mathcal{A}| = 1$,

Itemset	Count
Bread	4
Milk	4
Diapers	4
Beer	3
Egg	1
Cola	2

- Suppose $s_{min} = 60\%$, which is equivalent to a minimum count equal to 3.

(b) Candidate pruning:

Remove itemsets with a low count, that is, we left with

Itemset	Count
Bread	4
Milk	4
Diapers	4
Beer	3

- Next iteration, we repeat sub-step (a) and (b), but with itemset of $\mathcal{A}$ of size

$$|\mathcal{A}| = 2$$

- Invoking the key inequality,

$$\operatorname{supp}(\mathcal{X} \cup \mathcal{Y}) \leq \operatorname{supp}(\mathcal{X})$$

the relevant itemset are the ones that do not involve Egg or Cola

TID	Bread	Milk	Diapers	Beer	Egg	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

- Thus, we have { `Bread`, `Milk` }, { `Bread`, `Diapers` }, { `Bread`, `Beer` }, { `Milk`, `Diapers` }, { `Milk`, `Beer` }, { `Diapers`, `Beer` },

(a) Candidate generation:

Itemset	Count
Bread, Milk	3
Bread, Diapers	3
Bread, Beer	2
Milk, Diapers	3
Milk, Beer	2
Diapers, Beer	3

(b) Candidate pruning:

Itemset	Count
Bread, Milk	3
Bread, Diapers	3
Milk, Diapers	3
Diapers, Beer	3

- Similarly, in the next iteration, we have

Itemset	Count
Bread, Milk, Diapers	3

- Notice every itemset of size 3 involving Beer is a superset of an infrequent itemset in the previous two iterations.

TID	Bread	Milk	Diapers	Beer	Egg	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Itemset	Count
Bread, Milk	3
Bread, Diapers	3
Bread, Beer	2
Milk, Diapers	3
Milk, Beer	2
Diapers, Beer	3

- In general, the candidate generation sub-step in Apriori algorithm merges a pair of itemsets of size $(k-1)$ only if their first $(k-2)$ items are idential.

- Let

$$\mathcal{A} = \{a_1, a_2, \ldots, a_{k-1}\} \qquad \text{and} \qquad \mathcal{B} = \{b_1, b_2, \ldots, b_{k-1}\}$$

be a pair of frequent itemsets generated in the $(k-1)$th iteration.

- In the kth iteration, $\mathcal{A}$ and $\mathcal{B}$ are merged

$$\mathcal{A} \cup \mathcal{B}$$

to form an itemset of size k if the following conditions are satisfied

$$\begin{aligned} a_i &= b_i \qquad \text{for} \quad i = 1, 2, \ldots, k-2 \\ a_{k-1} &= b_{k-1} \end{aligned}$$

- Iterating until no such pair of frequent itemsets exists, which is the end of

1. Frequent Itemset Generation

(a) Candidate generation

(b) Candidate pruning

- Now consider rule generation given a frequent itemset $\mathcal{A} = \mathcal{X} \cup \mathcal{Y}$ of size k

$$\text{conf}\left(\mathcal{X} \implies \mathcal{Y}\right) \geq c_{min}$$

- For example, we have the following as one of the frequent itemsets

$$\mathcal{A} = \{\ \texttt{Bread, Milk, Diapers}\ \}$$

- There are six candidate association rules that can be generated from $\mathcal{A}$:

$$\begin{aligned}
\{\ \texttt{Bread, Milk}\ \} &\implies \{\ \texttt{Diapers}\ \} \\
\{\ \texttt{Bread, Diapers}\ \} &\implies \{\ \texttt{Milk}\ \} \\
\{\ \texttt{Milk, Diapers}\ \} &\implies \{\ \texttt{Bread}\ \} \\
\{\ \texttt{Bread}\ \} &\implies \{\ \texttt{Milk, Diapers}\ \} \\
\{\ \texttt{Milk}\ \} &\implies \{\ \texttt{Bread, Diapers}\ \} \\
\{\ \texttt{Diapers}\ \} &\implies \{\ \texttt{Bread, Milk}\ \}
\end{aligned}$$

Q: How can we efficiently generate and prune rules given a frequent itemset?

- Note if a rule $\mathcal{C} \implies \mathcal{A}-\mathcal{C}$, where $\mathcal{C} \subset \mathcal{A}$, is low confidence rule, that is,

$$\mathrm{conf}\left(\mathcal{C} \implies \mathcal{A}-\mathcal{C}\right) < c_{min}$$

then any rule $\mathcal{C}^* \implies \mathcal{A}-\mathcal{C}^*$, where $\mathcal{C}^* \subset \mathcal{C}$, is also low confident,

$$\mathrm{conf}\left(\mathcal{C}^* \implies \mathcal{A}-\mathcal{C}^*\right) < c_{min}$$

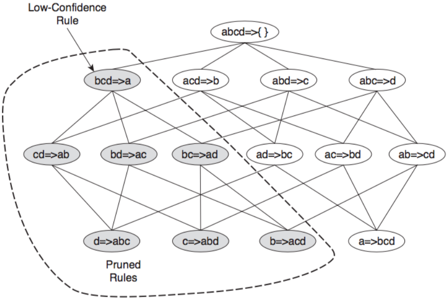
- This is clear from the definition of

$$\begin{aligned}\mathrm{conf}\left(\mathcal{C} \implies \mathcal{A}-\mathcal{C}\right) &= \frac{\mathrm{supp}\left(\mathcal{C}\cup\left(\mathcal{A}-\mathcal{C}\right)\right)}{\mathrm{supp}\left(\mathcal{C}\right)} = \frac{\mathrm{supp}\left(\mathcal{A}\right)}{\mathrm{supp}\left(\mathcal{C}\right)}\\ \mathrm{conf}\left(\mathcal{C}^* \implies \mathcal{A}-\mathcal{C}^*\right) &= \frac{\mathrm{supp}\left(\mathcal{C}^*\cup\left(\mathcal{A}-\mathcal{C}^*\right)\right)}{\mathrm{supp}\left(\mathcal{C}^*\right)} = \frac{\mathrm{supp}\left(\mathcal{A}\right)}{\mathrm{supp}\left(\mathcal{C}^*\right)}\end{aligned}$$

Since $\mathcal{C}^* \subset \mathcal{C}$, we have

$$\mathrm{supp}\left(\mathcal{C}^*\right) \geq \mathrm{supp}\left(\mathcal{C}\right)$$

- Thus we should start with the largest possible $\mathcal{C}$ for a given frequent itemset,



and precede to the next level only if both rules satisfy

$$\begin{matrix} \text{conf}\,(\mathcal{C}_1 \implies \mathcal{A} - \mathcal{C}_1) < c_{min} \\ \text{conf}\,(\mathcal{C}_2 \implies \mathcal{A} - \mathcal{C}_2) < c_{min} \end{matrix} \implies \text{conf}\,(\mathcal{C}_1 \cap \mathcal{C}_2 \implies \mathcal{A} - \mathcal{C}_1 \cap \mathcal{C}_2)$$

- Rules from 2 itemsets are not related, so are generated and pruned separately