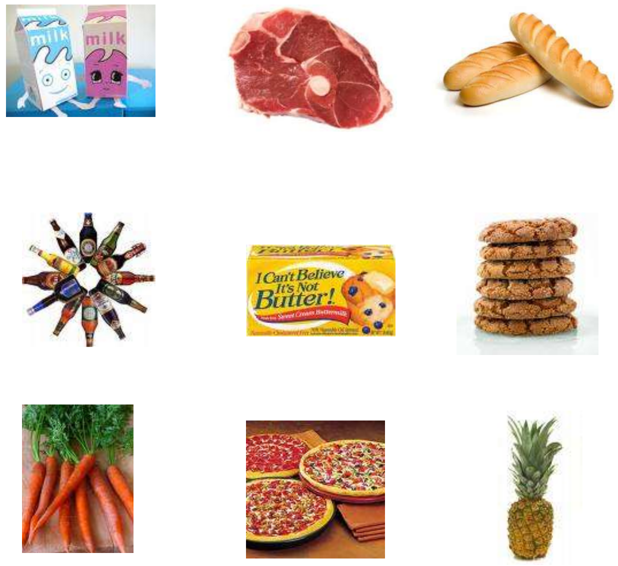
Association Rule

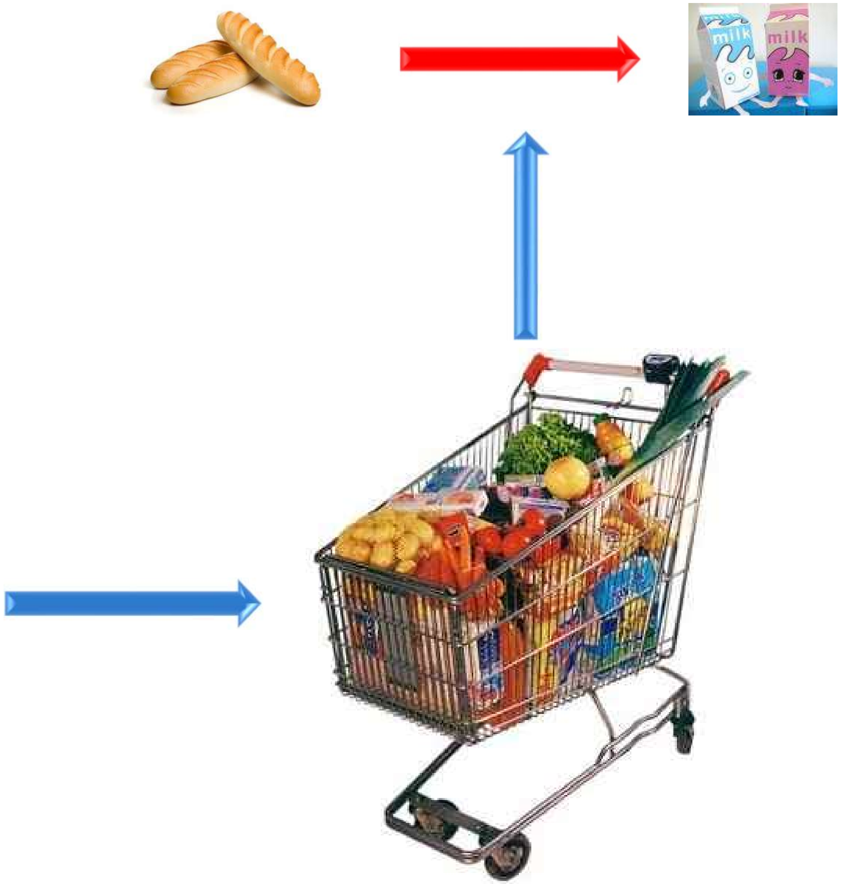
Lecturer: Dr. Bo Yuan

E-mail: yuanb@sz.tsinghua.edu.cn

Overview

- Frequent Itemsets
- Association Rules
- Sequential Patterns





A Real Example

Frequently Bought Together

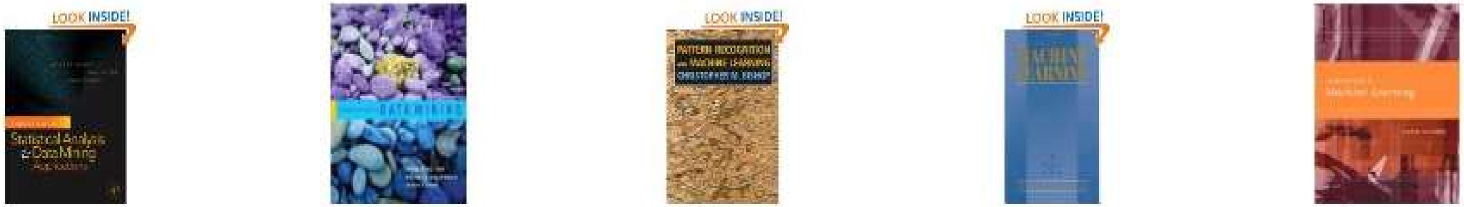
Price For All Three: $166.83

Add all three to Cart | Add all three to Wish List

Show availability and shipping details

- [x] **This item:** Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems) by Eibe Frank
- [x] The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) by Robert Tibshirani
- [x] Pattern Recognition and Machine Learning (Information Science and Statistics) by Christopher M. Bishop

Customers Who Bought This Item Also Bought



- Handbook of Statistical Analysis and Data Mining... by John Elder IV (9) $71.96
- Introduction to Data Mining by Pang-Ning Tan (15) $80.80
- Pattern Recognition and Machine Learning... by Christopher M. Bishop (47) $58.03
- Machine Learning (Mcgraw-Hill International Edit) by Tom M. Mitchell (38) $73.72
- Introduction to Machine Learning (Adaptive Comp... by Ethem Alpaydin (8) $43.79

Market-Based Problems

- Finding associations among items in a transactional database.
- Items
  - Bread, Milk, Chocolate, Butter …
- Transaction (Basket)
  - A non-empty subset of all items
- Cross Selling
  - Selling additional products or services to an existing customer.
- Bundle Discount
- Shop Layout Design
  - Minimum Distance vs. Maximum Distance
- “Baskets” & “Items”: Sentences & Words

Definitions

- A transaction is a set of items: $T=\{i_a, i_b,\ldots,i_t\}$
- T is a subset of I where I is the set of all possible items.
- The dataset D contains a set of transactions.
- An association rule is in the form of

$$P \Rightarrow Q \text{ where } P \subset I, Q \subset I \text{ and } P \cap Q = \varnothing$$

- A set of items is referred to as itemset.
- An itemset containing k items is called k-itemset.
- An itemset can be seen as a conjunction of items.

Transactions

Transactions	Items
1	Bread, Jelly, Peanut, Butter
2	Bread, Butter
3	Bread, Jelly
4	Bread, Milk, Butter
5	Chips, Milk
6	Bread, Chips
7	Bread, Milk
8	Chips, Jelly

Searching for rules in the form of: Bread → Butter

Support of an Itemset

- The support of an item (or itemset) X is the percentage of transactions in which that item (or itemset) occurs.

$$Support(X) = \frac{\#X}{n}$$

Itemset	Support	Itemset	Support
Bread	6/8	Bread, Butter	3/8
Butter	3/8	…	
Chips	2/8	Bread, Butter, Chips	0/8
Jelly	3/8	…	
Milk	3/8	Bread, Butter, Chips, Jelly	0/8
Peanut	1/8	…	
		Bread, Butter, Chips, Jelly, Milk	0/8
		…	
		Bread, Butter, Chips, Jelly, Milk, Peanut	0/8

Support & Confidence of Association Rule

- The support of an association rule X→Y is the percentage of transactions that contain X and Y.

$$Support(X \rightarrow Y) = \frac{\#\,(X \cup Y)}{n}$$

- The confidence of an association rule X→Y is the ratio of the number of transactions that contain {X, Y} to the number of transactions that contain X.

$$Confidence(X \rightarrow Y) = \frac{\#\,(X \cup Y)}{\#(X)}$$

- It can be represented equally as

$$Confidence(X \rightarrow Y) = \frac{Support(X \cup Y)}{Support(X)}$$

- Conditional probability: P(Y|X)

Support & Confidence of Association Rule

- Support measures how often the rule occurs in the dataset.
- Confidence measures the strength of the rule.

Transactions	Items
1	Bread, Jelly, Peanut, Butter
2	Bread, Butter
3	Bread, Jelly
4	Bread, Milk, Butter
5	Chips, Milk
6	Bread, Chips
7	Bread, Milk
8	Chips, Jelly

Bread → Milk

Support: 2/8

Confidence: 1/3

Milk → Bread

Support: 2/8

Confidence: 2/3

Frequent Itemsets and Strong Rules

- Support and Confidence are bounded by thresholds:
  - Minimum support σ
  - Minimum confidence Φ
- A frequent (large) itemset is an itemset with support larger than σ.
- A strong rule is a rule that is frequent and its confidence is higher than Φ.
- Association Rule Problem
  - Given I, D, σ and Φ, to find all strong rules in the form of $X \rightarrow Y$.
- The number of all possible association rules is huge.
  - Brute force strategy is infeasible.
  - A smart way is to find frequent itemsets first.

The Big Picture

- Step 1: Find all frequent itemsets.

- Step 2: Use frequent itemsets to generate association rules.
  - For each frequent itemset f
    - Create all non-empty subsets of f.
  - For each non-empty subset s of f
    - Output s → (f-s) if support (f) / support (s) > Φ

$$\{a, b, c\} \quad \left\{ \begin{array}{l} a\ b \rightarrow c \\ a\ c \rightarrow b \\ b\ c \rightarrow a \\ a \rightarrow b\ c \\ b \rightarrow a\ c \\ c \rightarrow a\ b \end{array} \right.$$

The key is to find frequent itemsets.

Myth No. 1

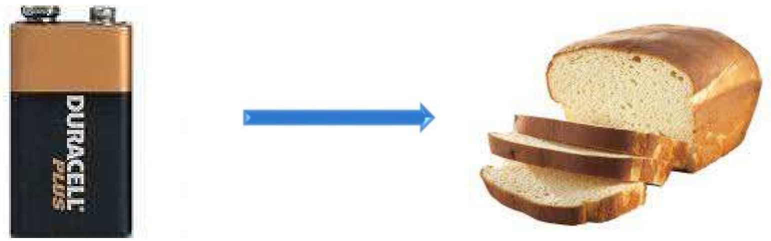
- A rule with high confidence is not necessarily plausible.
- For example:
  - |D|=10000
  - #{DVD}=7500
  - #{Tape}=6000
  - #{DVD, Tape}=4000
  - Thresholds: σ=30%, Φ=50%
  - Support(Tape → DVD)= 4000/10000=40%
  - Confidence(Tape → DVD)=4000/6000=66%

- Now we have a strong rule: Tape → DVD
  - Seems that Tapes will help promote DVDs.
  - However, P(DVD)=75% > P(DVD | Tape) !!
  - Tape buyers are less likely to purchase DVDs.

Myth No. 2

Transactions
Bread, Milk
Bread, Battery
Bread, Butter
Bread, Honey
Bread, Chips
Yogurt, Coke
Bread, Battery
Cookie, Jelly

$$P(Bread \mid Battery) = 100\% > P(Bread) = 75\%$$

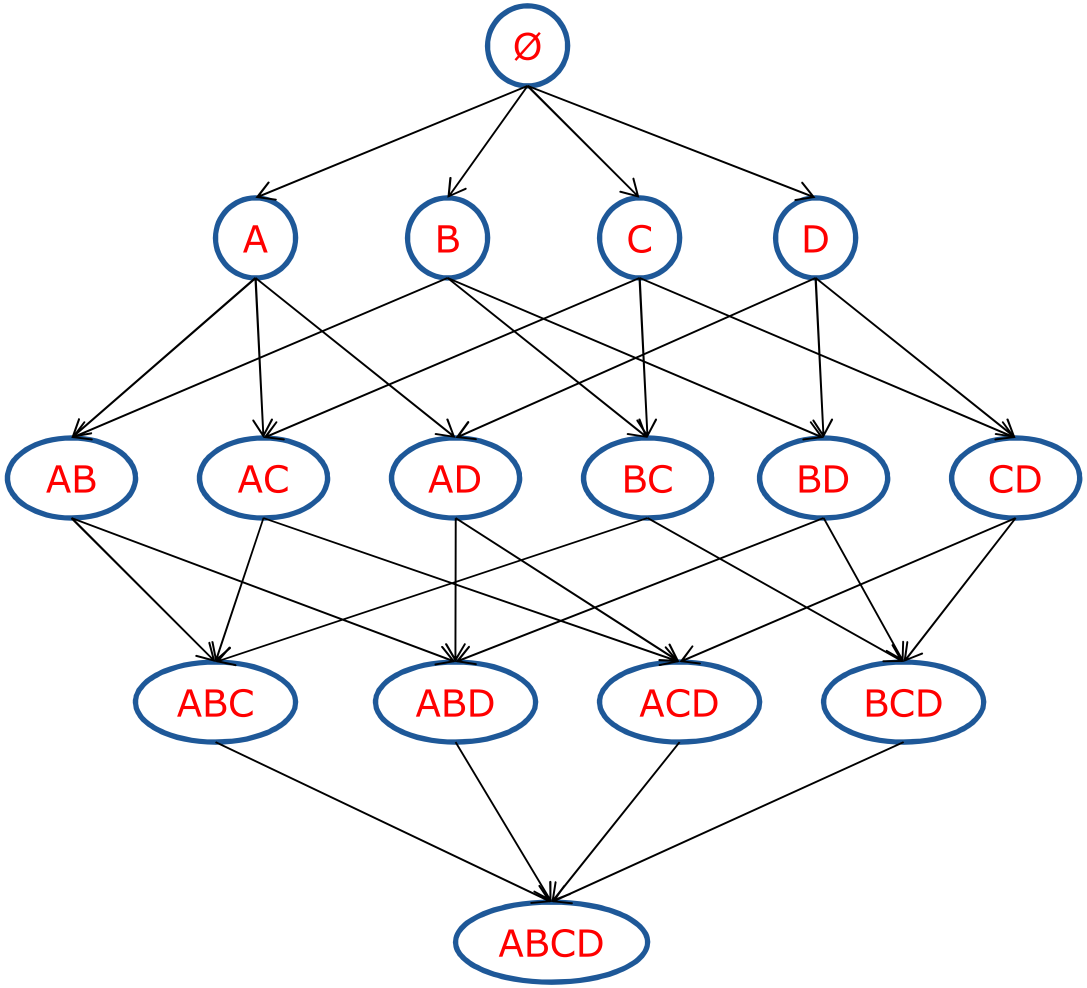


Myth No. 3

Association $\neq$ Causality

P(Y|X) is just the conditional probability.

Itemset Generation



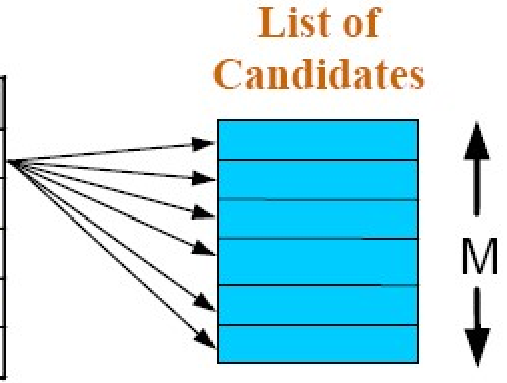
Itemset Calculation

Transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, Water, Eggs
3	Milk, Diaper, Water, Coke
4	Bread, Milk, Diaper, Water
5	Bread, Milk, Diaper, Coke

N

w



$$O(NMW)$$

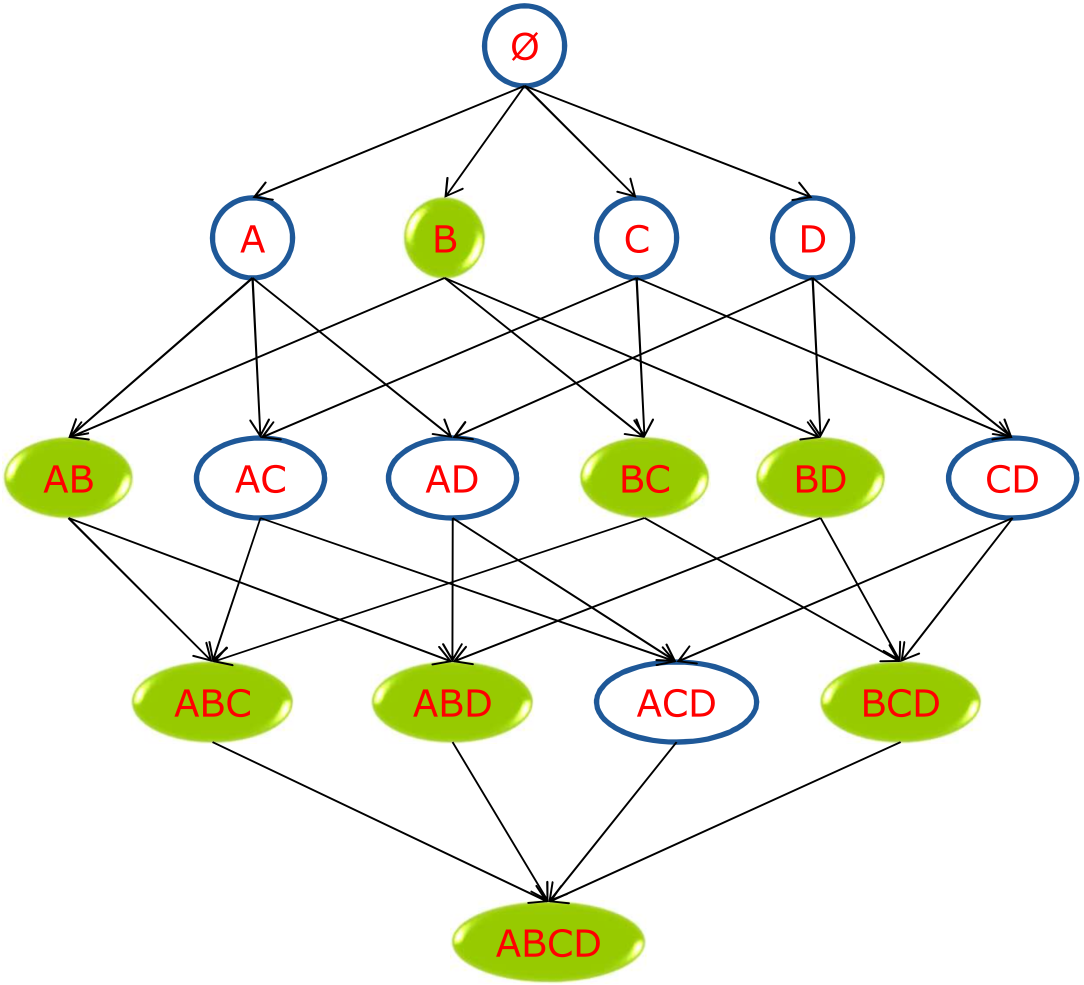
$$M = 2^d - 1$$

The Apriori Method

- One of the best known algorithms in Data Mining
- Key ideas
  - A subset of a frequent itemset must be frequent.
    - {Milk, Bread, Coke} is frequent → {Milk, Coke} is frequent
  - The supersets of any infrequent itemset cannot be frequent.
    - {Battery} is infrequent → {Milk, Battery} is infrequent

Title 1–20	Cited by	Year
Fast algorithms for mining association rules R Agrawal, R Srikant Proc. 20th int. conf. very large data bases, VLDB 1215, 487-499	19603	1994
Mining association rules between sets of items in large databases R Agrawal, T Imieliński, A Swami ACM SIGMOD Record 22 (2), 207-216	17129	1993
Mining sequential patterns R Agrawal, R Srikant Data Engineering, 1995. Proceedings of the Eleventh International Conference ...	6017	1995

Candidate Pruning



General Procedure

- Generate itemsets of a particular size.
- Scan the database once to see which of them are frequent.
- Use frequent itemsets to generate candidate itemsets of size=size+1.
- Iteratively find frequent itemsets with cardinality from 1 to k.
- Avoid generating candidates that are known to be infrequent.
- Require multiple scans of the database.
- Efficient indexing techniques such as Hash function & Bitmap may help.

Apriori Algorithm

C_k: Candidate itemset of size k

L_k: Frequent itemset of size k

$L_1 \leftarrow \{frequent\ items\}$

for (k=1; $L_k \neq \emptyset$; k++)

$C_{k+1} \leftarrow candidate(L_k)$ candidates

for each transaction t

$Q \leftarrow \{c \mid c \in C_{k+1} \wedge c \subseteq t\}$

$count[c] \leftarrow count[c] + 1, \quad \forall c \in Q$ counting

end for

$L_{k+1} \leftarrow \{c \mid c \in C_{k+1} \wedge count[c]/N \geq \sigma\}$ filtering

end for

return $\bigcup_k L_k$

$L_k \rightarrow C_{k+1}$

$$L_1 = \{1, 2, 3, 4, 5\} \qquad L_2 = \{\{1, 2\}, \{2, 3\}\}$$

$$\{X \cup p \mid X \in L_k, p \in L_1, p \notin X\}$$

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

$$\{X \cup Y \mid X, Y \in L_k, |X \cap Y| = k - 1\}$$

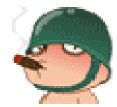
$$C_3 = \{\{1, 2, 3\}\}$$

$L_k \rightarrow C_{k+1}$

$\{X \bigcup Y_k \mid X, Y \in L_k, X_i = Y_i, \forall i \in [1, k-1], X_k \neq Y_k\}$ Ordered List

$L_2 = \{\{1,2\},\{2,3\}\}$ $C_3 = \{\}$

$L_2 = \{\{1,3\},\{2,3\}\}$ $C_3 = \{\}$



$L_2 = \{\{1,2\},\{1,3\},\{2,3\}\}$ $C_3 = \{\{1,2,3\}\}$

$L_2 = \{\{1,2\},\{1,3\}\}$ $C_3 = \{\{1,2,3\}\}$

Correctness

$$\boxed{\forall X, X \in L_{k+1} \Rightarrow X \in C_{k+1}}$$

$$\{X_1, \ldots, X_k, X_{k+1}\} \in L_{k+1}$$

$$\Downarrow$$

$$\{X_1, \ldots, X_{k-1}, X_k\} \in L_k$$

$$\{X_1, \ldots, X_{k-1}, X_{k+1}\} \in L_k$$

Join

$$\{X_1, \ldots, X_{k-1}, X_k, X_{k+1}\} \in C_{k+1}$$

Demo

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

Support > 1

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Note: {1,2,3} {1,2,5} and {1,3,5} not in C_3

Clothing Example

Apriori-Gen Algorithm – Clothing Example

- Given: 20 clothing transactions; s=20%, c=50%
- Generate association rules using the Apriori algorithm

Transaction	Items	Transaction	Items
t_1	Blouse	t_{11}	TShirt
t_2	Shoes, Skirt, TShirt	t_{12}	Blouse, Jeans, Shoes, Skirt, TShirt
t_3	Jeans, TShirt	t_{13}	Jeans, Shoes, Shorts, TShirt
t_4	Jeans, Shoes, TShirt	t_{14}	Shoes, Skirt, TShirt
t_5	Jeans, Shorts	t_{15}	Jeans, TShirt
t_6	Shoes, TShirt	t_{16}	Skirt, TShirt
t_7	Jeans, Skirt	t_{17}	Blouse, Jeans, Skirt
t_8	Jeans, Shoes, Shorts, TShirt	t_{18}	Jeans, Shoes, Shorts, TShirt
t_9	Jeans	t_{19}	Jeans
t_{10}	Jeans, Shoes, TShirt	t_{20}	Jeans, Shoes, Shorts, TShirt

- Scan1: Find all 1-itemsets. Identify the frequent ones.

  Candidates: ~~Blouse~~, Jeans, Shoes, Shorts, Skirt, Tshirt

  Support: ~~3/20~~ 14/20 10/20 5/20 6/20 14/20

  Frequent (Large): Jeans, Shoes, Shorts, Skirt, Tshirt

  Join the frequent items – combine items with each other to generate candidate pairs

Clothing Example

Clothing Example – cont.1

Jeans, Shoes, Shorts, Skirt, Tshirt

- **Scan2: 10 candidate 2-itemsets were generated. Find the frequent ones.**

{Jeans, Shoes}:7/20 {Shoes, Short}:4/20 ~~{Short, Skirt}: 0/20~~ {Skirt, TShirt}: 4/20

{Jeans, Short} :5/20 ~~{Shoes, Skirt}: 3/20~~ {Short, TShirt}: 4/20

~~{Jeans, Skirt}~~ :3/20 {Shoes, TShirt}: 10/20

{Jeans, TShirt}:9/20 4/20 *7 frequent itemsets are found out of 10.*

Scan	Candidates	Large Itemsets
1	{Blouse}, {Jeans}, {Shoes}, {Shorts}, {Skirt}, {TShirt}	{Jeans}, {Shoes}, {Shorts} {Skirt}, {Tshirt}
2	{Jeans, Shoes}, {Jeans, Shorts}, {Jeans, Skirt}, {Jeans, TShirt}, {Shoes, Shorts}, {Shoes, Skirt}, {Shoes, TShirt}, {Shorts, Skirt}, {Shorts, TShirt}, {Skirt, TShirt}	{Jeans, Shoes}, {Jeans, Shorts}, {Jeans, TShirt}, {Shoes, Shorts}, {Shoes, TShirt}, {Shorts, TShirt}, {Skirt, TShirt}
3	{Jeans, Shoes, Shorts}, {Jeans, Shoes, TShirt}, {Jeans, Shorts, TShirt}, {Jeans, Skirt, TShirt}, {Shoes, Shorts, TShirt}, {Shoes, Skirt, TShirt}, {Shorts, Skirt, TShirt}	{Jeans, Shoes, Shorts}, {Jeans, Shoes, TShirt}, {Jeans, Shorts, TShirt}, {Shoes, Shorts, TShirt}
4	{Jeans, Shoes, Shorts, TShirt}	{Jeans, Shoes, Shorts, TShirt}
5	∅	∅

Everyone is combined with each other

2 sets are joined if they have 1 item in common (i,.e. 1 item different)

2 sets are joined if they have 2 item in common (i,.e. 1 item different)

Clothing Example – cont.2

- The next step is to use the large itemsets and generate association rules
- c=50%
- The set of large itemsets is

  L={~~{Jeans}~~,~~{Shoes}~~, ~~{Shorts}~~, ~~{Skirt}~~, ~~{TShirt}~~, {Jeans, Shoes}, {Jeans, Shorts}, {Jeans, TShirt}, {Shoes, Shorts}, {Shoes, TShirt}, {Shorts, TShirt}, {Skirt, TShirt}, {Jeans, Shoes, Shorts}, {Jeans, Shoes, TShirt}, {Jeans, Shorts, TShirt},{Shoes, Shorts, TShirt}, {Jeans, Shoes, Shorts,TShirt} }

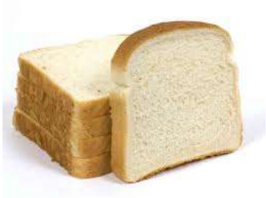
- We ignore the first 5 as they do not consists of 2 nonempty subsets of large itemsets. We test all the others, e.g.:

$$Confidence(Jeans \rightarrow Shoes) = \frac{Support(\{Jeans, Shoes\})}{Support(\{Jeans\})} = \frac{7/20}{14/20} = 50\% \geq c$$

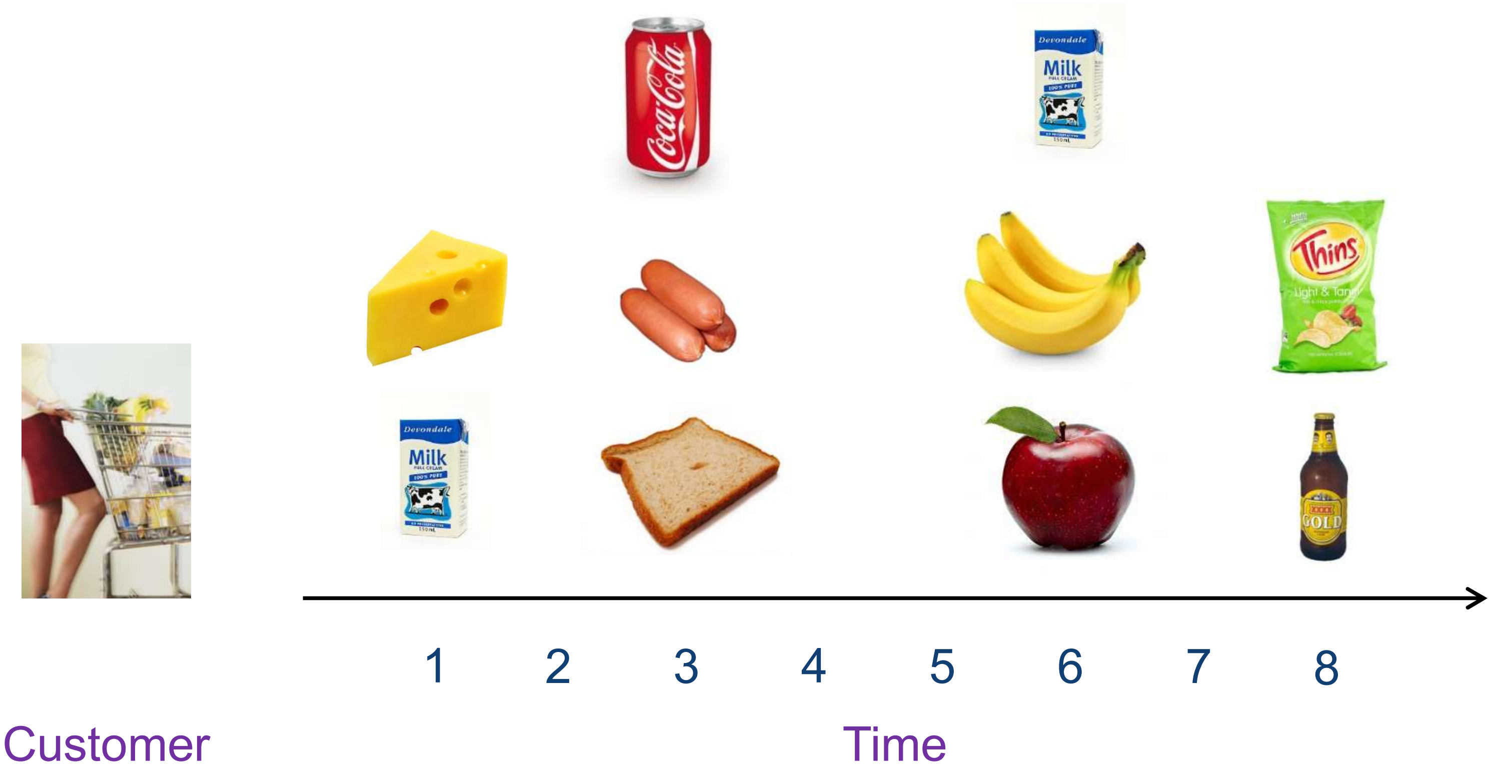
Real Examples

Effective Recommendation

这孩子，没救了..



Sequential Pattern



Sequence

- A sequence is an ordered list of elements where each element is a collection of one or more items.

- $t=<t_1\ t_2\ \ldots\ t_m>$ is a subsequence of $s=<s_1\ s_2\ \ldots\ s_n>$ if there exist integers $1 \leq j_1<j_2< \ldots <j_m \leq n$ such that $t_1 \subseteq s_{j1}$, $t_2 \subseteq s_{j2}, \ldots, t_m \subseteq s_{jm}$.

s	t	Y/N
<{2, 4} {3, 6, 5} {8}>	<{2} {3, 6} {8}>	Yes
<{2, 4} {3, 6, 5} {8}>	<{2} {8}>	Yes
<{1, 2} {3, 4}>	<{1} {2}>	No
<{2, 4} {2, 4} {2, 5}>	<{2} {4}>	Yes

Support of Sequence

CID	Time	Items
A	1	1, 2, 4
A	2	2, 3
A	3	5
B	1	1, 2
B	2	2, 3, 4
C	1	1, 2
C	2	2, 3, 4
C	3	2, 4, 5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

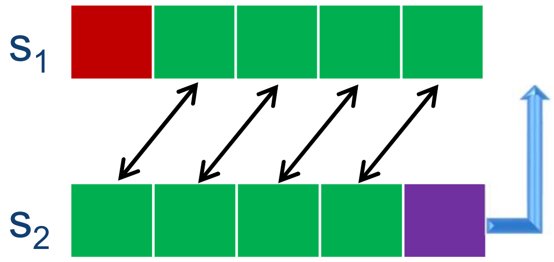
Support	
<{1, 2}>	60%
<{2, 3}>	60%
<{2, 4}>	80%
<{3} {5}>	80%
<{1} {2}>	80%
<{2} {2}>	60%
<{1} {2, 3}>	60%
<{2} {2, 3}>	60%
<{1, 2} {2, 3}>	60%

Candidate Space

- Given: {Milk} {Bread}
- 2-itemset: {Bread, Milk}
- 2-sequence:
  - <{Bread, Milk}>
  - <{Bread} {Milk}>, <{Milk} {Bread}>
  - <{Bread} {Bread}>, <{Milk} {Milk}>
- Order matters in sequences but not for itemsets.
- For 1000 items: $1000 \times 1000 + \frac{1000 \times 999}{2} = 1499500$
- The search space is much larger than before.
- How to generate candidates efficiently?

Candidate Generation

- A sequence s_1 is merged with another sequence s_2 if and only if the subsequence obtained by dropping the first item in s_1 is identical to the subsequence obtained by dropping the last item in s_2.



3-sequences
<{1} {2} {3}>
<{1} {2, 5}>
<{1} {5} {3}>
<{2} {3} {4}>
<{2, 5} {3}>
<{3} {4} {5}>
<{5} {3, 4}>

Candidate
<{1} {2} {3} {4}>
<{1} {2, 5} {3}>
<{1} {5} {3, 4}>
<{2} {3} {4} {5}>
<{2, 5} {3, 4}>

Pruning
<{1} {2, 5} {3}>

Reading Materials

- Text Book
  - J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Chapter 6, Morgan Kaufmann.

- Core Papers

  - J. Han, J. Pei, Y. Yin and R. Mao (2004) “Mining frequent patterns without candidate generation: A frequent-pattern tree approach”. *Data Mining and Knowledge Discovery*, Vol. 8(1), pp. 53-87.
  - R. Agrawal and R. Srikant (1995) “Mining sequential patterns”. In *Proceedings of the Eleventh International Conference on Data Engineering* (ICDE), pp. 3-14.
  - R. Agrawal and R. Srikant (1994) “Fast algorithms for mining association rules”. In *Proceedings of the 20th International Conference on Very Large Data Bases* (VLDB), pp. 487-499.
  - R. Agrawal, T. Imielinski, and A. Swami (1993) “Mining association rules between sets of items in large databases”. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (SIGMOD), pp. 207-216.