Data Mining Algorithms: Association Rules

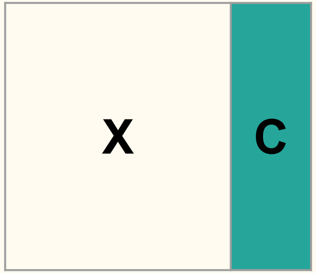
Department of Computer Science and Engineering
Kathmandu University

Supervised, unsupervised and semi-supervised learning

Supervised Learning

Given:

- A set of data that contains both the inputs (X) and the desired outputs (C)



Task:

- Build a mathematical model consistent with the training data

Goal:

- To predict the class of a new object knowing its attributes

Unsupervised Learning

Given:

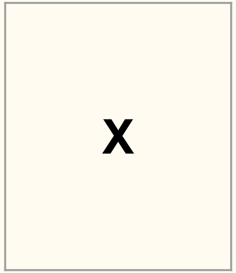
- A set of data that contains only inputs (X) and no desired outputs

Task:

- Build a mathematical model consistent with the data

Goal:

- Identify homogeneous groups in a dataset



Semi-supervised learning

Learn a mathematical model from incomplete training data, where a portion of the sample input doesn't have labels/desired outputs

Mining frequent patterns

Mining frequent patterns

Frequent pattern:

A pattern (a set of items, subsequences, subgraphs, etc.) that occurs frequently in a data set

Motivation:

Finding inherent regularities (associations) in data

Association rules

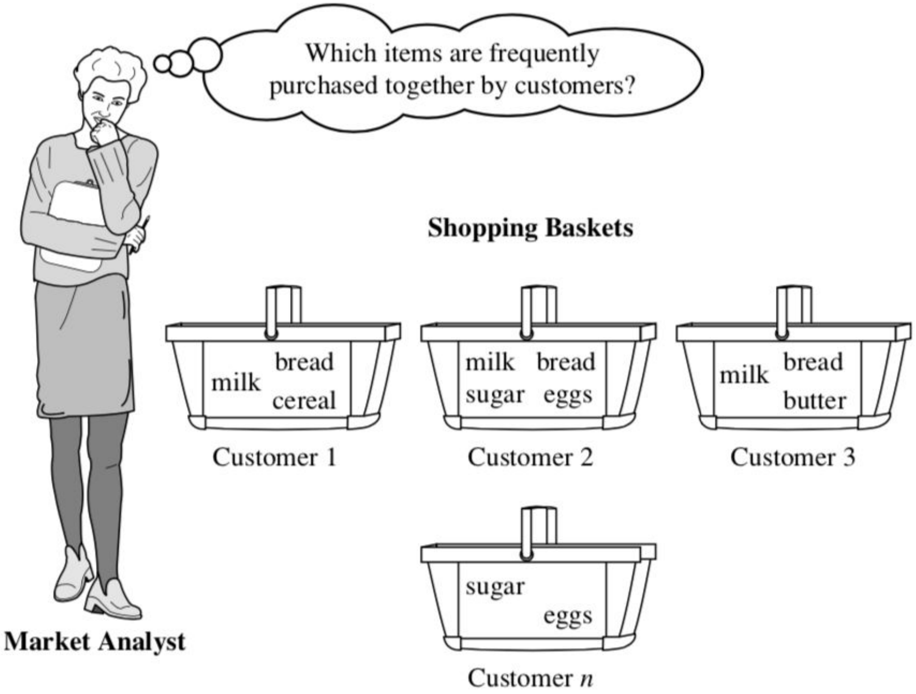
- **Rules**: Models that consist of a set of “IF Condition THEN Conclusion” rules learned from underlying data

- **Supervised rule learners:** classification and regression trees (CART, ID3, CHAID, C4.5), class association rules (CBA, CPAR, CMAR, ...)
- **Unsupervised rule learners**: Clustering based on decision trees, **association rules** (first proposed by Agrawal et al., 1993 in the context of frequent itemsets and association rule mining for market basket analysis)

Market basket analysis

Analyze customer buying habits by finding associations between the different items that customers place in their “shopping baskets”.

Help develop marketing strategies by gaining insight into which items are frequently purchased together by customers.



Market basket analysis

Market basket analysis: 5 items, 8 transactions

1. Bread
2. Butter
3. Mustard
4. Jam
5. Sausage

Detail of transactions

t1: Bread, Butter
t2: Bread, Mustard
t3: Bread, Mustard, Sausage
t4: Bread, Butter, Jam
t5: Sausage
t6: Bread, Butter, Jam
t7: Bread, Mustard, Sausage
t8: Bread, Butter, Jam

Database D

TID	Items
1	{1, 2}
2	{1, 3}
3	{1, 3, 5}
4	{1, 2, 4}
5	{5}
6	{1, 2, 4}
7	{1, 3, 5}
8	{1, 2, 4}

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Market basket analysis

Database D

TID	Items
1	{1, 2}
2	{1, 3}
3	{1, 3, 5}
4	{1, 2, 4}
5	{5}
6	{1, 2, 4}
7	{1, 3, 5}
8	{1, 2, 4}

Transactional matrix

Trans. \ Item	1	2	3	4	5	tot.
t1	1	1				2
t2	1		1			2
t3	1		1		1	3
t4	1	1		1		3
t5					1	1
t6	1	1		1		3
t7	1		1		1	3
t8	1	1		1		3
tot.	7	4	3	3	3	20

A **transactional matrix** for n transactions, p items is an n x p matrix, where

$x_{ij} = 1$ if j is bought in transaction i
$x_{ij} = 0$ otherwise

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Market basket analysis

Database D

TID	Items
1	{1, 2}
2	{1, 3}
3	{1, 3, 5}
4	{1, 2, 4}
5	{5}
6	{1, 2, 4}
7	{1, 3, 5}
8	{1, 2, 4}

Transactional matrix

Trans. \ Item	1	2	3	4	5	tot.
t1	1	1				2
t2	1		1			2
t3	1		1		1	3
t4	1	1		1		3
t5					1	1
t6	1	1		1		3
t7	1		1		1	3
t8	1	1		1		3
tot.	7	4	3	3	3	20

Interpretation:

- t4 contains 3 items
- Item 1 appears in 7 transactions
- In total, 20 items have been bought

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Terminologies

An **itemset** is a set of items: $\{i_1, i_2, ..., i_n\}$

k-itemset = a set (or a conjunction) of k distinct items
Example: {bread, butter, jam} is a *3-itemset*

Itemset support: Proportion (or number) of transactions which contain the considered itemset
Example: {Supp({bread, butter, jam}) = 3/8}

Itemsets: examples

Itemset	Length	Support
{4,5}	2	0
{3,5}	2	2
{3,4}	2	0
{2,5}	2	0
{2,4}	2	3
{2,3}	2	0
{1,5}	2	2
{1,4}	2	3
{1,3}	2	3
{1,2}	2	4
{5}	1	3
{4}	1	3
{3}	1	3
{2}	1	4
{1}	1	7
empty	0	8

Itemset	Length	Support
...	4	0
{3,4,5}	3	0
{2,4,5}	3	0
{2,3,5}	3	0
{2,3,4}	3	0
{1,4,5}	3	0
{1,3,5}	3	2
{1,3,4}	3	0
{1,2,5}	3	0
{1,2,4}	3	3
{1,2,3}	3	0

Important remark: the support of *{2,3}* is *0*, so the support of *{1,2,3}*, *{2,3,4}* and *{2,3,5}* is necessarily *0*!

Antimonotonic property of the support!

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Notations and definitions

Formalism of market basket analysis (Agrawal et al. 93, 94)

- $T = \{t_1, t_2, \ldots, t_i, \ldots, t_n\}$, a set of n transactions
- $I = \{i_1, i_2, \ldots, i_j, \ldots, i_p\}$, a set of p items
- X(n, p), the corresponding transaction-item association matrix
- A and B two itemsets having no common item

Association rules

- Association rules reveals the interactions between the items
- Let A and B be two itemsets having no common item, then an association rule is an implication of the form $A \Rightarrow B$
- A : body, antecedent,
- B : head, consequent
- If the items of A are in the basket, usually the items of B also!

Rule evaluation: support and confidence

Rule support:

- Fraction of transactions which contain all the items of A and B (i.e., $P(A \cup B)$)
- **Interpretation**: Support assesses the generality of the rule

Rule confidence:

- Fraction of transactions which contain the items of B among those which contain the items of A (i.e., $P(B \mid A)$)
- **Interpretation**: Confidence assesses the strength of the rule

Support and confidence

Example: r = {bread, butter} $\Rightarrow$ {jam}

Supp(A) = 4/8 = 0.50

Supp(B) = 3/8 = 0.375

Support:

$$Supp\left(A \rightarrow B\right) = \frac{n_{ab}}{n} = p_{ab}$$

Confidence:

$$Conf\left(A \rightarrow B\right) = \frac{n_{ab}/n}{n_a/n} = \frac{p_{ab}}{p_a}$$

Supp(A $\Rightarrow$ B)= Supp(AB)=3/8

Conf(A $\Rightarrow$ B) = Supp(AB)/Supp(A) = 3/4 = 0.75

Association rule extraction/mining

Support-confidence approach

Exponential complexity: there are 2^p-1 possible itemsets and 3^p - $2^{p+1} + 1$ possible rules! If p = 20, there are 2,097,152 possible itemsets and 3,486,784,401 possible rules ...

Problem: n and p are usually very large. How to extract the interesting rules?

Extraction of strong rules (min_supp and min_conf being two prefixed threshold):

- **Frequent rule:** support of the rule (p_{ab}) exceeds min_supp
- **Confident rule:** confidence of the rule ($p_{b/a}$) exceeds min_conf

Justification: the antimonotonicity of the support condition allows to prune the search space

Support-confidence approach

Task: Find all the rules that satisfy both a minimum support (min_sup) and a minimum confidence (min_conf) thresholds.

Steps:

1. Find all **frequent itemsets** (with support $\geq$ min_sup).
2. Generate strong rules from the frequent itemsets (conf $\geq$ min_conf)

Antimonotony of the support

- Any sub-itemset of a frequent itemset is frequent
- Any super-itemset of a non-frequent itemset is non-frequent

Example: for a support threshold of 0.375,

- {mustard, sausage} is non-frequent, so {bread, mustard, sausage} is non-frequent

- {bread, butter, jam} is frequent, so {butter, jam} is necessarily frequent

The Apriori Algorithm

The Apriori algorithm

- The baseline algorithm of support-confidence approach
- Exploits the antimonotonic property of the support to perform the first step (i.e., if there is any pattern which is infrequent, its superset should not be generated/tested!)

Step 1. Frequent itemset search (min_sup) by browsing the itemset lattice
If $A \Rightarrow B$ is a strong rule, $X = \{A \cup B\}$ is necessarily a frequent itemset produced by this step

Step 2. For each frequent itemset X, all the rules of the type $X \backslash Y \rightarrow Y$ (where $Y \subset X$) are examined and only the confident rules (min_conf) are stored

-

Apriori example

Database D

TID	Items
1	{1, 3, 4}
2	{2, 3, 5}
3	{1, 2, 3, 5}
4	{2, 5}

Transaction-items matrix

trans. \ item	1	2	3	4	5	total
1	1		1	1		3
2		1	1		1	3
3	1	1	1		1	4
4		1			1	2
total	2	3	3	1	3	12

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Apriori example

Step 1 : Frequent itemsets search
Support and confidence thresholds :
Min_{Supp} = 0.50, Min_{Conf} = 0.75

Database D

TID	Items
1	{1, 3, 4}
2	{2, 3, 5}
3	{1, 2, 3, 5}
4	{2, 5}

Scan D →

C1

Itemset	Supp
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

F1

Itemset	Sup.
{1}	2
{2}	3
{3}	3
{5}	3

Only frequent items (green) of C1 are kept in F1

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Apriori example

C2 is built from F1
Only frequent items (green) of C2 are kept in F2

C2

Itemset
{1, 2}
{1, 3}
{1, 5}
{2, 3}
{2, 5}
{3, 5}

Scan D →

Itemset	Supp
{1, 2}	1
{1, 3}	2
{1, 5}	1
{2, 3}	2
{2, 5}	3
{3, 5}	2

C2

Itemset	Supp
{1, 3}	2
{2, 3}	2
{2, 5}	3
{3, 5}	2

F2

C3 is built from F2
Only frequent items (green) of C3 are kept in F3

C3

Itemset
{1, 2, 3}
{1, 3, 5}
{2, 3, 5}

Scan D →

Itemset	Supp
{1, 2, 3}	1
{1, 3, 5}	1
{2, 3, 5}	2

C3

Itemset	Supp
{2, 3, 5}	2

F3

26

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Apriori example

List of frequent itemsets (supp ≥ 0.5)

Taille	Itemset	Support
2	{1, 3}	0,50
2	{2, 3}	0,50
2	{2, 5}	0,75
2	{3, 5}	0,50
3	{2, 3, 5}	0,50

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Apriori example

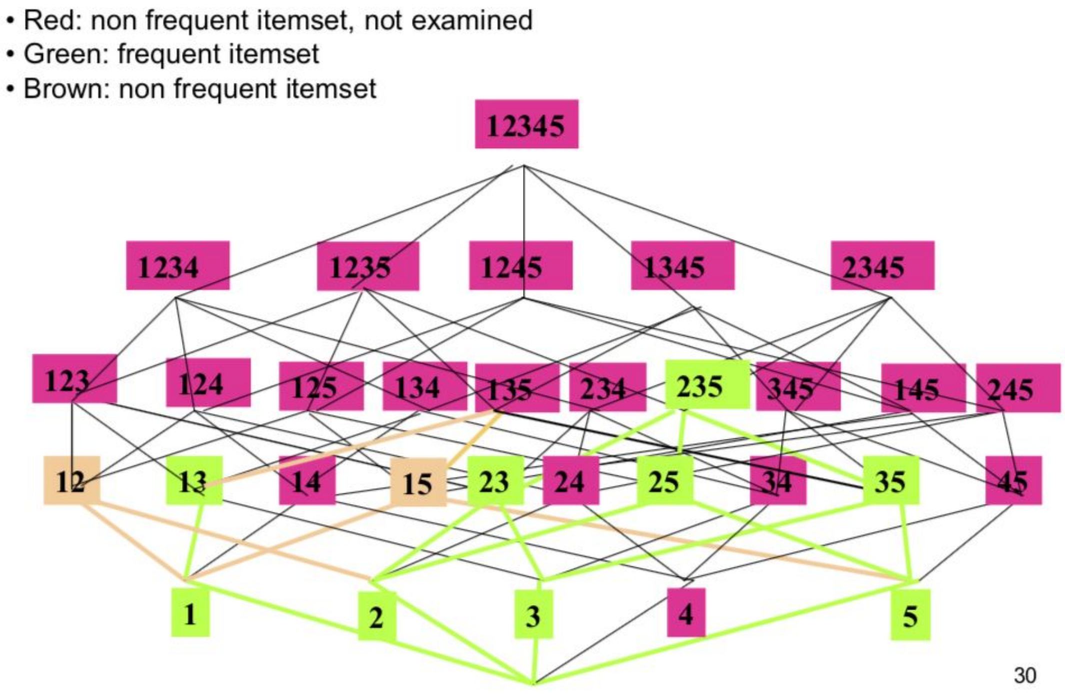
Step 2: Selection of strong rules

From each frequent itemset, all possible rules are examined, only the confident rules are retained, those whose Conf ≥ 0.75

Size	Itemset	Rule	Support	Confidence
2	{1, 3}	1 --> 3	0,5	1
		3 --> 1		0,67
2	{2, 3}	2 --> 3	0,5	0,67
		3 --> 2		0,67
2	{2, 5}	2 --> 5	0,75	1
		5 --> 2		1
2	{3, 5}	3 --> 5	0,5	0,67
		5 --> 3		0,67
3	{2, 3, 5}	35 --> 2	0,5	1
		25 --> 3		0,67
		23 --> 5		1
		2 --> 35		0,67
		3 --> 25		0,67
		5 --> 23		0,67

Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

Itemsets lattice



Source: Lallich, S. (2013). Association Rules [Lecture slides]. University of Lyon.

The Apriori algorithm

Frequent itemset generation (level-wise search):

1. Initially, scan D once to get frequent 1-itemset
2. For each level k:
   2.1. Generate length (k+1) candidates from length k frequent itemsets
   2.2. Scan D and remove the infrequent candidates
3. Terminate when no candidate set can be generated

The Apriori algorithm

Candidate generation: Generating k+1 candidates at level k

- Step 1: **Self-joining** two frequent k-itemsets if they have the same k-1 prefix
- Step 2: **Pruning**: remove a candidate if it contains any infrequent k-itemset

Example: L3={abc, abd, acd, ace, bcd}

- – Self-joining: L3*L3
    - abc and abd ⇒ abcd
    - acd and ace ⇒ acde
- Pruning:
    - acde is removed because ade is not in L3

The Apriori algorithm

Pros

- Efficient pruning
- Support and confidence → intelligible measures
- Exhaustive search

Cons

- Huge candidate sets
- Multiple scans of database
- Many rules are without any interest

Improvements:

- Database partitioning (Savasere et al. 95)
- FP-Growth algorithm (Han et al. 00)

Quality and interestingness measures

Quality and interestingness measures

Support-confidence approach neglects rules with a small support though some may have a high confidence

If the support threshold is lowered to remedy this inconvenience, even more rules are produced

The support and confidence conditions cannot detect independence. If X and Y are independent : $P(Y|X) = P(Y)$
If $P(Y)$ is high $\rightarrow$ nonsense rule with high support

Correlation Analysis

The support and confidence measures are insufficient at filtering out uninteresting association rules.

To tackle this weakness, a correlation measure can be used to augment the support–confidence framework for association rules.

Example of a correlation measure: Lift, χ^2

Lift

The **lift** between the occurrence of A and B can be measured by computing

$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}.$$

If lift(A, B) $<$ 1, then A and B are negatively correlated (i.e., the occurrence of one likely leads to the absence of the other one).

If lift(A, B) $>$ 1, then A and B are positively correlated.

If lift(A, B) $=$ 1, then A and B are independent.