Machine Learning

Association Rule Learning

Dr. Sherif Saad

Learning Objectives

Introduce association rules learning

Understand how to extract association rules form a set of observation

Apply Apriori Algorithm to extract frequent itemsets

Generate association rules from frequent itemsets.

Association Rules Learning

Association rules learning is another key unsupervised machine learning method, after clustering, that finds interesting associations (relationships, dependencies) in large sets of data items.

It is the most common machine learning approach in data mining to extract knowledge and insights from data.

The discovery of interesting associations provides a source of information often used by businesses for decision making.

The most common applications of association rules learning are: Market-basket Data Analysis, Recommendation Systems.

Association Rules Learning

In general, association rules learning is categorized under rule-based machine learning methods (ML methods that relies on a set of rules). It is also a rule induction method.

Rule-based machine learning applies some form of learning algorithm to automatically identify useful rules, rather than a human needing to apply prior domain knowledge to manually construct rules and curate a rule set.

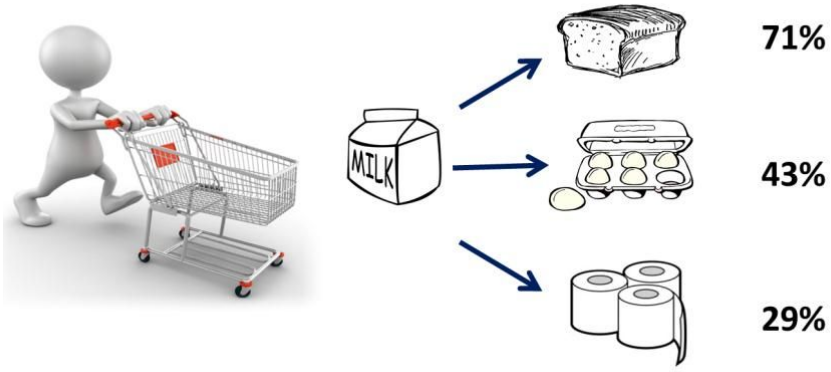
Rule induction is an area of machine learning in which formal rules are extracted from a set of observations. Rule inductions algorithms could apply supervised learning or unsupervised learning.

Association Rules Learning

Transaction	Items Set
T1	Bread, Milk, Coke
T2	Coke, Coffee, Tobacco
T3	Bread, Milk, Egg, Toilet Paper, Coffee
T4	Toilet Paper, Bread, Egg
T5	Milk, Egg, Toilet Paper, Coke
T6	MilK, Bread, Egg
T7	Coffee, Milk, Diaper
T8	Tobacco, Milk, Egg, Toilet Paper

Association Rules Learning

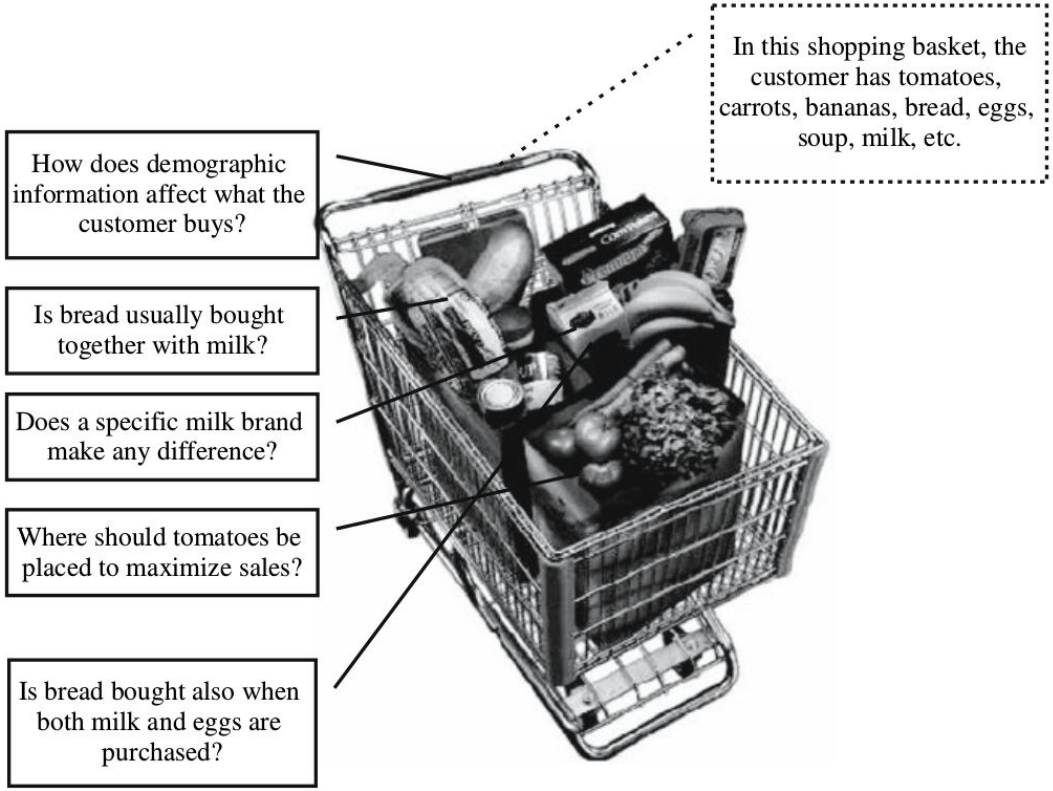
Market-basket analysis, one of the most intuitive applications of association rules, strives to analyze customer buying patterns by finding associations between items that customers put into their baskets. For instance, one can discover that customers buy milk and bread together, and even that some particular brands of milk are more often bought with certain brands of bread.



Of transactions that included milk:

- 71% included bread
- 43% included eggs
- 29% included toilet paper

Association Rules Learning



Association Rules Learning

The items are stored in the form of transactions that can be generated by an external process, or extracted from relational databases or any other datastore.

In a market-basket analysis, we represent each product in a store as a Boolean variable, which represents whether an item is present or absent. Each customer's basket is represented as a Boolean vector, denoting which items are purchased.

The vectors are analyzed to find which products are frequently bought together (by different customers)

Association Rules Learning

In general, we can think of association rules as a set of if-then rules.

Each extracted association rule is structured as follows:

LHS ⇒ RHS

Where the ***left-hand side implies the right-hand side***, with a given value of support and confidence.

Support and confidence are used to measure the quality of a given rule, in terms of its usefulness (strength) and certainty.

Association Rules Learning

To measure the quality of the rules or their usefulness we use two metrics, support and confidence.

Support tells how many examples (transactions) from a data set that was used to generate the rule include items from both LHS and RHS.

Confidence expresses how many examples (transactions) that include items from LHS also include items from RHS.

An association rule is considered interesting if it satisfies minimum values of confidence and support.

Association Rules Learning

For any given rule we can calculate the support and the confidence for that rule as follows.

$$Rule : X \Rightarrow Y$$

$$Support(X, Y) = \frac{frq(X, Y)}{N}$$

$$Confidence(X, Y) = \frac{frq(X, Y)}{frq(X)}$$

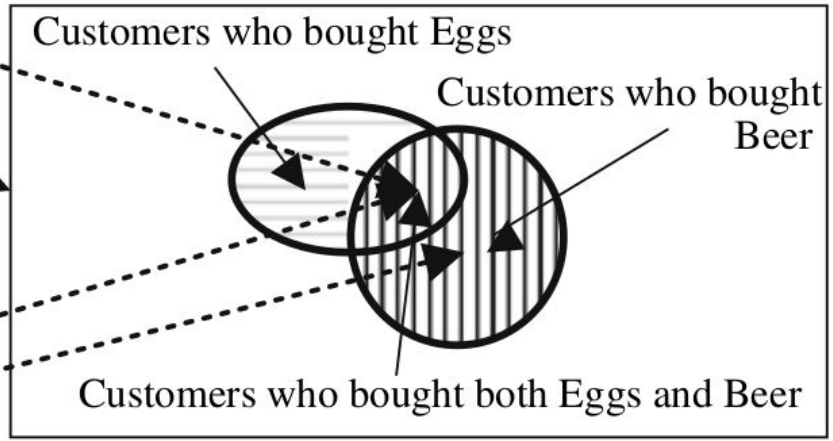
Association Rules find all sets of items (itemsets) that have ***support*** greater than the minimum support and then using the large itemsets to generate the desired rules that have ***confidence*** greater than the minimum confidence.

Association Rules Learning

How can we interpret the support and the confidence of an association rule

$$s\,(A \Rightarrow B) = \frac{\|\{T \in D \mid A \cup B \subseteq T\}\|}{\|D\|}$$

$$c\,(A \Rightarrow B) = \frac{\|\{T \in D \mid A \cup B \subseteq T\}\|}{\|\{T \in D \mid A \subseteq T\}\|}$$



Association Rules Learning

Examples of association rules

$$\text{buys}\ (x, \text{milk}) \Rightarrow \text{buys}\ (x, \text{bread})\ [25\%, 60.0\%]$$

Milk and bread are bought together in 25% of store purchases (transactions), and 60% of the baskets that include milk also include bread.

What does the following rules say about PhD computer engineering students

$$\text{major}(x, \text{Computer Engineering})\ \text{AND}\ \text{takes\_course}(x, \text{Advanced Data Analysis and Decision Making}) \Rightarrow \text{level}(x, \text{PhD})\ [1\%, 75\%]$$

Association Rules Learning

Types of association rules:

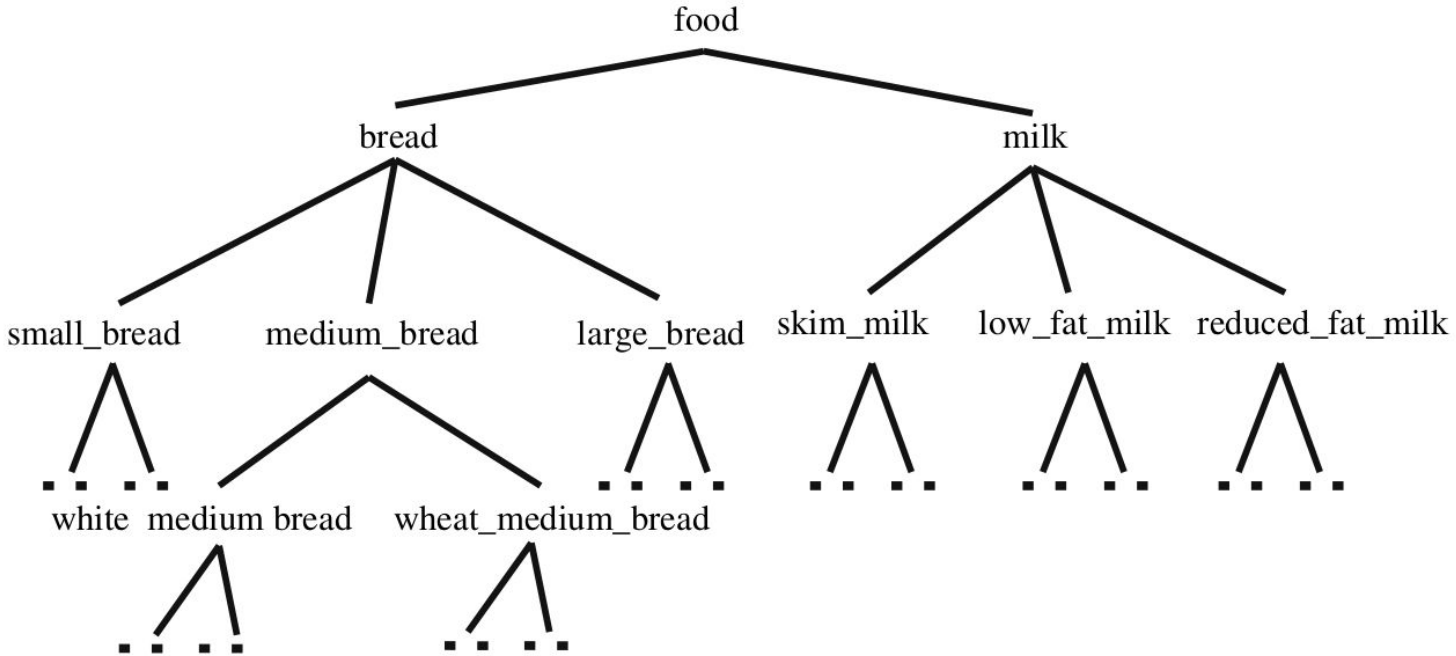
Single-dimensional and multidimensional rules; this based on the number of events in the rules e.g buy, major, takes_course, level, etc

Single-level and Multilevel; the single operate on a single level of abstraction, while the latter are based on items that can be expressed at different levels in a hierarchy. For example a multi-level rule

$$\text{buys (x, skim\_milk)} \Rightarrow \text{buys (x, large\_white\_bread)} \; [2.5\%, 60.0\%]$$

Association Rules Learning

Multi-level association rules relies on a taxonomy structure of the items



Association Rules and Transactional Data

Strong Association Rules, rules that satisfy both the minimum support threshold and the minimum confidence threshold.

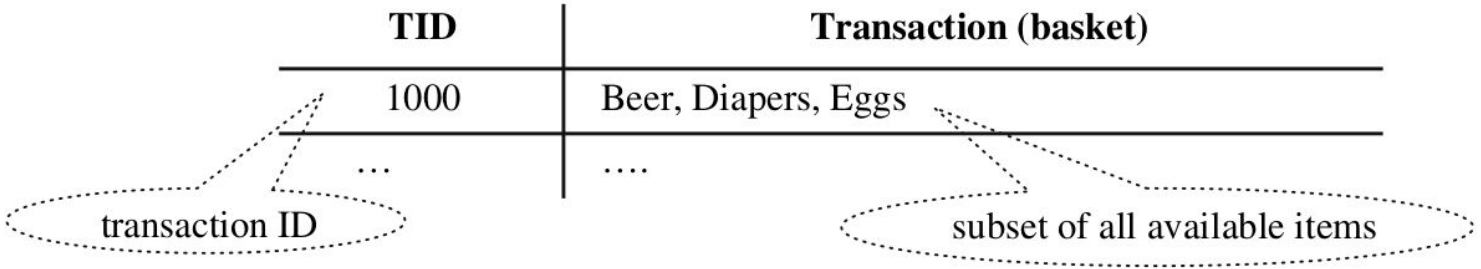
A set of items is referred to as an itemset, an itemset that contains k items is referred to as a k-itemset. For instance, {Beer, Eggs} is a 2-itemset.

Support count (also known as frequency, occurrence frequency, or count) of an itemset is the number of transactions in **D** that contain the itemset.

A frequent itemset is an itemset that satisfies a minimum support level

The number of transactions required for an itemset to satisfy minimum support is called the minimum support count.

Association Rules and Transactional Data



TID	Transaction (basket)
1000	Apples, Celery, Diapers
2000	Beer, Celery, Eggs
3000	Apples, Beer, Celery, Eggs
4000	Beer, Eggs

Association Rules and Transactional Data

For the given transactions, find all rules such that LHS $= \{A, B\}$, RHS $= \{C\}$, with minimum support $= 50\%$ and minimum confidence $= 50\%$.

TID	Transactions
1000	A, B, C
2000	A, C
3000	A, D
4000	B, E, F

Support is the probability that a transaction contains $\{A, B, C\}$, and *confidence* is the conditional probability that a transaction containing $\{A, B\}$ also contains C.

Rule $A \wedge B \Rightarrow C$ [support 25%, confidence 100%] does not satisfy the minimum confidence. Two (shorter) strong association rules are generated as:

$A \Rightarrow C$ [support 50%, confidence 66.6%]

$C \Rightarrow A$ [support 50%, confidence 100%]

Association Rules and Transactional Data

For the given transactions

TID	Transactions
1000	Apples, Celery, Diapers
2000	Beer, Celery, Eggs
3000	Apples, Beer, Celery, Eggs
4000	Beer, Eggs

find I, T for TID = 2000, *support* (Beer $\Rightarrow$ Eggs), and *confidence* (Beer $\Rightarrow$ Eggs)

$I = \{\text{Apples, Beer, Celery, Diapers, Eggs}\}$

$T = \{\text{Beer, Celery, Eggs}\}$

support (Beer $\Rightarrow$ Eggs) = 75%

confidence (Beer $\Rightarrow$ Eggs) = 100%

Association Rules Learning Algorithms

The following four steps are used to generate single-dimensional association rules:

1. Prepare input data in the transactional format.
2. Choose items of interest, i.e., itemsets.
3. Compute support counts to evaluate whether selected itemsets are frequent, i.e., whether they satisfy minimum support.
4. Given the frequent itemsets, generate strong association rules that satisfy the minimum confidence by computing the corresponding conditional probabilities (counts).

Association Rules Learning Algorithms

The simplest way to compute frequent itemsets is to consider all possible itemsets, compute their support, and check whether they are higher than the minimum support threshold.

A naïve algorithm for generation of frequent itemsets will takes $\boldsymbol{O(2^m N)}$, where m is the size of the itemsets and n is the number of transactions.

Several algorithms have been proposed to efficiently learn strong association rules and reduce the runtime complexity

Apriori Algorithm

Key idea: A subset of a frequent itemset must also be a frequent itemset

For example if {Bread, Milk, Egg} is a frequent itemset then {Bread, Milk}, {Bread, Egg}, {Milk, Egg} must be frequent itemsets.

It uses an iterative approach to find all frequent itemsets with a given support.

It generates the association rules by extracting all frequent itemsets.

The algorithm uses an incremental approach where k-itemsets are used to explore [k+1]-itemsets

Apriori Algorithm

In each iteration the algorithm generate a list of candidate itemsets based on the verified frequent itemsets from the previous iteration.

The candidate itemsets of the current iteration are checked first for pruning operation to eliminate infrequent candidate itemsets.

Only candidate itemsets that survived the pruning operation are tested or verified against the database.

The algorithm generate the frequent itemset that can be used to generate association rules.

Apriori Algorithm

Given the following transactional data set find all itemsets with support 2 or more and generate association rules with confidence = 65%

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E
500	A, C, E

Apriori Algorithm

Iteration 1: We generate the candidate itemsets of size 1 and find the sets that satisfy the support value

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E
500	A, C, E

Database

Itemset	Support
A	3
B	3
C	4
D	1
E	4

Candidate Itemsets

Itemset	Support
A	3
B	3
C	4
E	4

Frequent Itemsets

Apriori Algorithm

Iteration 2: we extend the frequent itemsets from the previous iteration by one additional item

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E
500	A, C, E

Database

Itemset	Support
A, B	1
A, C	3
A, E	2
B, C	2
B, E	3
C, E	3

Candidate Itemsets

Itemset	Support
A, C	3
A, E	2
B, C	2
B, E	3
C, E	3

Frequent Itemsets

Apriori Algorithm

Iteration 3: we extend the frequent itemsets from the previous iteration by one additional item

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E
500	A, C, E

Database

Itemset	Support
A, C, E	2
A, B, C	?
A, B, E	?
B, C, E	2

Candidate Itemsets

Itemset	Support
A, C, E	2
B, C, E	2

Frequent Itemsets

Apriori Algorithm

Iteration 4: we extend the frequent itemsets from the previous iteration by one additional item

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E
500	A, C, E

Database

Itemset	Support
A, B, C, E	1

Candidate Itemsets

Itemset	Support
{empty}	

Frequent Itemsets

Itemset	Support
A, C, E	2
B, C, E	2

Final Output

Generating Association Rule from Frequent Set

Given the frequent itemsets generated by the Apriori algorithm we generate the association rules by:

1. Generate all nonempty subsets from each frequent itemset
2. Let **L** be the frequent itemset and **s** is any subset of **L** then an association rule ***α*** is a strong rule if ***support(L-s)/support(s) >= target confidence***.

Itemset	Support
A, C, E	2
B, C, E	2

Final Output

{A, C, E} = {A, C}, {A, E}, {C, E}, {A}, {C}, {E}

{B, C, E} = {B, C}, {B, E}, {C, E}, {B}, {C}, {E}

Generating Association Rule from Frequent Set

- R1: buy(A, C) ⇒ buy(E)
  - support(A, C, E)/support(A, C)= 2/3
- R2: buy(A, E) ⇒ buy(C)
  - support(A, E, C)/support(A, E)= 2/2
- R3: buy(E,C) ⇒ buy(A)
  - support(E, C, A)/support(E, C)= 2/3
- R4: buy(A) ⇒ buy(E, C)
  - support(A, E, C)/support(A)= 2/4
- R5: buy(C) ⇒ buy(A, E)
  - support(C, A, E)/support(C)= 2/4
- R6: buy(E) ⇒ buy(A,C)
  - support(E, A, C)/support(E)= 2/4

{A, C, E} = {A, C}, {A, E}, {C, E}, {A}, {C}, {E}

{B, C, E} = {B, C}, {B, E}, {C, E}, {B}, {C}, {E}

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E
500	A, C, E

Database

Itemset	Support
A, C, E	2
B, C, E	2

Final Output

Questions