Data Mining

A short introduction

Andrea Polini

Process Mining
MSc in Computer Science (LM-18)
University of Camerino

Summary

Summary

Data Mining

Definition

analysis of (often large) data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.

Input data is typically given as a table and the output may be rules, clusters, tree structures, graphs, equations, patterns, etc.

Motivations

Drinker	Smoker	Weight	Age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...	...	...	...

Interesting questions

- What is the effect of smoking and drinking on a person's bodyweight?
- Do people that smoke also drink?
- What factors influence a person's life expectancy the most?
- Can one identify groups of people having a similar lifestyle?

Motivations

Linear algebra	Logic	Programming	Operations research	Workflow systems	...	Duration	Result
9	8	8	9	9	...	36	cum laude
7	6	–	8	8	...	42	passed
–	–	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	–	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	–	6	6	...	52	failed
...	...	...	...	...	...	...	...

Interesting questions

- Are the marks of certain courses highly correlated?
- Which electives do excellent students (cum laude) take?
- Which courses significantly delay the moment of graduation?
- Why do students drop out?
- Can one identify groups of students having a similar study behavior?

Motivations

Cappuccino	Latte	Espresso	Americano	Ristretto	Tea	Muffin	Bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Interesting questions

- Which products are frequently purchased together?
- When do people buy a particular product?
- Is it possible to characterize typical customer groups?
- How to promote the sales of products with a higher margin?

Variables classification

- Categorical: limited set of possible values, easily to enumerate
  - ordinal: a logical order exists
  - nominal: no logical ordering
- Numerical: have an ordering and cannot be enumerated easily

- Which are the classes for the first dataset?
- what about {"*cum laude*","*failed*","*passed*"}?

Note:

- Logs can contains errors, and in any case data need to be pre-processed before mining
- In general in data mining rows of a table are not ordered in time (order of rows has no meaning). Logs can be transformed for data mining purpose (feature extraction)

Variables classification

- Categorical: limited set of possible values, easily to enumerate
  - ordinal: a logical order exists
  - nominal: no logical ordering
- Numerical: have an ordering and cannot be enumerated easily

- Which are the classes for the first dataset?
- what about {"*cum laude*", "*failed*", "*passed*"}?

Note:

- Logs can contains errors, and in any case data need to be pre-processed before mining
- In general in data mining rows of a table are not ordered in time (order of rows has no meaning). Logs can be transformed for data mining purpose (feature extraction)

Data mining techniques classification

Data Mining strategy can be classified as:

- supervised
- unsupervised

Data mining results may be both descriptive and predictive. Decision trees, association rules, regression functions say something about the data set used to learn the model. However, they can also be used to make predictions for new instances, e.g., predict the overall performance of students based on the course grades in the first semester.

Supervised learning

Supervised learning assumes labeled data – there is a response variable that labels each instance

- response variables – dependent
- predictor variables – independent

Linear algebra	Logic	Programming	Operations research	Workflow systems	...	Duration	Result
9	8	8	9	9	...	36	cum laude
7	6	–	8	8	...	42	passed
–	–	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	–	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	–	6	6	...	52	failed
...	...	...	...	...	...	...	...

Supervised learning

Supervised learning assumes labeled data – there is a response variable that labels each instance

- response variables – dependent
- predictor variables – independent

Depending on the type of response variable used we distinguish different approaches:

- Classification – uses categorical response variables
- Regression – uses numerical response variables

Linear algebra	Logic	Programming	Operations research	Workflow systems	...	Duration	Result
9	8	8	9	9	...	36	cum laude
7	6	–	8	8	...	42	passed
–	–	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	–	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	–	6	6	...	52	failed
...	...	...	...	...	...	...	...

Supervised learning

Supervised learning assumes labeled data – there is a response variable that labels each instance

- response variables – dependent
- predictor variables – independent

Depending on the type of response variable used we distinguish different approaches:

- Classification – uses categorical response variables
- Regression – uses numerical response variables

Drinker	Smoker	Weight	Age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...	...	...	...

Supervised learning

Supervised learning assumes labeled data – there is a response variable that labels each instance

- response variables – dependent
- predictor variables – independent

Depending on the type of response variable used we distinguish different approaches:

- Classification – uses categorical response variables
- Regression – uses numerical response variables

Numerical variable can be transformed in categorical one i.e. partition age into classes “young”, “middle-age”, “old”

Unsupervised learning

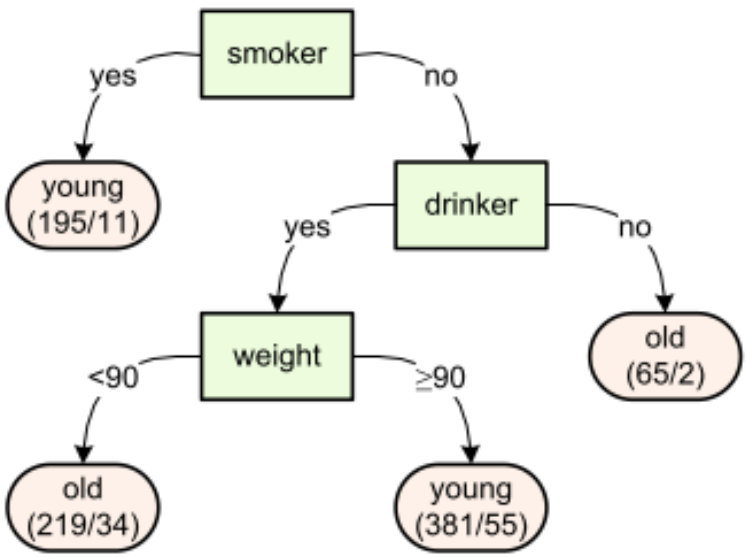
Unsupervised learning assumes unlabeled data – the variables are not split into response and predictor variables

- clustering – analysis of data to find groups of instances
- pattern discovery – generally the goal is to find rules of the form "IF X THEN Y"
- ...

Summary

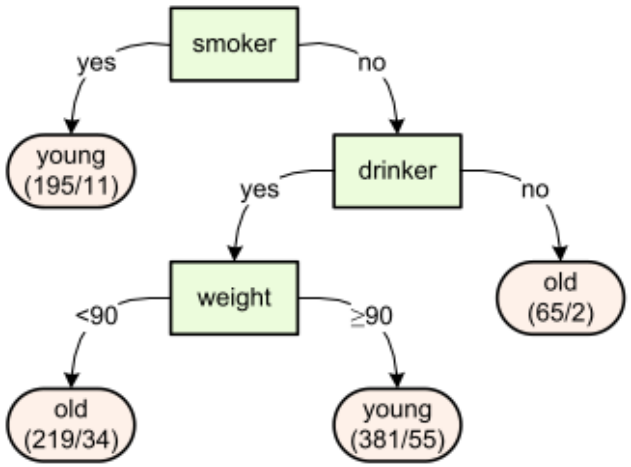
Decision Tree Learning

Decision tree learning is a supervised learning technique aiming at the classification of instances based on predictor variables. There is one categorical response variable labeling the data and the result is arranged in the form of a tree. Leaf nodes correspong to possible values of the response variable. Non-leaf nodes corresponds to predictor variables (attributes).



Decision Tree

Drinker	Smoker	Weight	Age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...	...	...	...

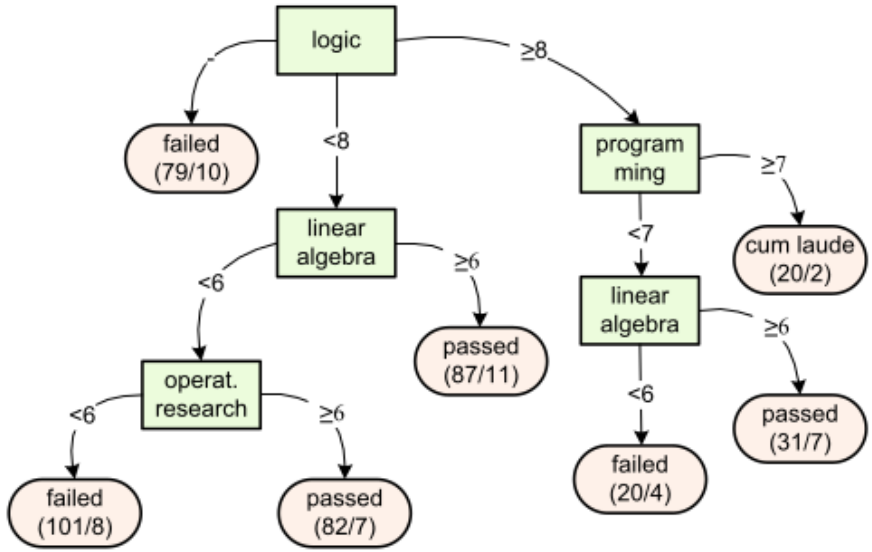


The tree classifies 860 persons into “young” (died before the age of 70) and “old” (died at 70 or later). We can notice that:

- People who smoke generally die young (195 persons of which 11 are misclassified)
- People who do not smoke and do no drink tend to live long (65 persons of which 2 are misclassified)
- People who only drink but are overweight ($\geq$ 90) also die young (381 persons of which 55 are misclassified)

Decision Tree

Linear algebra	Logic	Programming	Operations research	Workflow systems	...	Duration	Result
9	8	8	9	9	...	36	cum laude
7	6	–	8	8	...	42	passed
–	–	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	–	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	–	6	6	...	52	failed
...	...	...	...	...	...	...	...

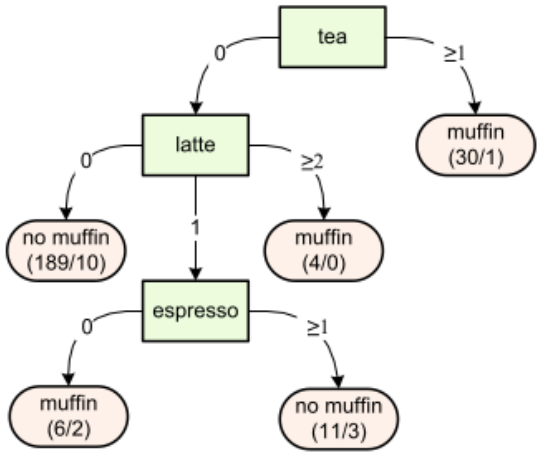


The tree classifies 420 students into “failed”, “passed”, and “cum laude” based on study results. We can notice that:

- Students that do not take the course on logic typically fail (79 students of which 10 are misclassified)
- Students that have a high mark for logic and programming, typically complete their degree cum laude (20 students of which 2 are misclassified)

Decision Tree

Cappuccino	Latte	Espresso	Americano	Ristretto	Tea	Muffin	Bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...



A decision tree derived from the table after converting response variable muffin into a Boolean.
We can notice that:

- Customers who drink tea tend to eat muffins (30 customers of which 1 is misclassified).
- Customers who do not drink tea or latte typically do not eat muffins (189 customers of which 10 are misclassified)

Learning a decision tree

Decision trees can be obtained using a variety of techniques. Most of the techniques use a recursive top-down algorithm:

1. Create the root node r and associate all instances to the root node. $X := \{r\}$ is the set of nodes to be traversed
2. If $X = \emptyset$, then return the tree with root r and end
3. Select $x \in X$ and remove it from X, i.e., $X := X \backslash \{x\}$. Determine the "score" $s^{old}(x)$ of node x before splitting, e.g., based on entropy (see next slide).
4. Determine if splitting is possible/needed. If not, go to step 2, otherwise continue with the next step.
5. For all possible attributes $a \in A$, evaluate the effects of splitting on the attribute. Select the attribute a providing the best improvement, i.e., maximize $s_a^{new(x)} - s^{old(x)}$. The same attribute should not appear multiple times on the same path from the root. Also note that for numerical attributes, so-called "cut values" need to be determined (cf. < 8 and ≥ 8).
6. If the improvement is substantial enough, create a set of child nodes Y , add Y to X (i.e., $X := X \cup Y$), and connect x to all child nodes in Y .
7. Associate each node in Y to its corresponding set of instances and go to step 2.

Entropy

Intuition

Entropy is an information-theoretic measure for the uncertainly in a **multi-set of elements**.

- If the multi-set contains many different elements and each element is unique, then variation is maximal and it takes many "bits" to encode the individual elements. Hence, the entropy is "high".
- If all elements in the multi-set are the same, then actually no bits are needed to encode the individual elements. In this case the entropy is "low".

For example, the entropy of the multi-set [a, b, c, d, e] is much higher than the entropy of the multi-set [a^5] even though both multi-sets have the same number of elements.

Entropy Measure

Assume that there is a multi-set X with n elements and there are k possible values, say $v_1, v_2, \ldots, v_k$, i.e., X is a multi-set over $V = \{v_1, v_2, \ldots, v_k\}$ with $|X| = n$. Each value v_i appears c_i times in X, i.e., $X = [(v_1)^{c_1}, (v_2)^{c_2}, \ldots, (v_k)^{c_k}]$. Without loss of generality, we can assume that $c_i \geq 1$ for all i, because values that do not appear in X can be removed from V upfront. The proportion of elements having value v_i is p_i , i.e., $p_i = c_i/n$. The entropy of X is measured in bits of information and is defined by the formula:

$$E = -\sum_{i=1}^{k} p_i \log_2 p_i$$

Alternative measures

The Gini-Simpson Index of diversity is another possible value to measure the "impurity" of a data set:

$$G = 1 - \sum_{i=1}^{k} {p_i}^2$$

For further indexes: https://en.wikipedia.org/wiki/Diversity_index

Entropy and bits

If all elements in X have the same value, i.e., $k = 1$ and $p_1 = 1$, then $E = -\log_2 1 = 0$. This means that no bits are needed to encode the value of an individual element. If all elements in X are different, i.e., $k = n$ and $p_i = 1/k$, then $E = -\sum_{i=1}^{k} (1/k)\log_2(1/k) = \log_2 k$. For instance, if there are 4 possible values, then 2 bits are needed to encode each individual element.

Consider the case of a stream in which the set of values to be communicated is $V = \{a, b, c, d\}$ with respective probability of $0.5, 0.25, 0.125, 0.125$. Which is the entropy? What could it means for an hypothetical communication protocol?

Let's apply the approach

Exercise: Lifestyle and life expectancy

In our population (860) people can drink, smoke and being overweight, and we would like to predict if these habits have influence on their life expectancy.

- 546 died before 70 ("young")
- Out of the total 195 are smokers and 184 of them died before 70
- among the non smokers, 600 drink alcohol and among them 360 died before 70, while among the 65 non drinkers only 2 died young
- among the 600 drinkers there were 219 people weighting less than 90Kg. 185 of them died after 70. Instead among the 381 over 90, 316 died before 70.

Let's learn the decision tree assessing if there is an information gain splitting the nodes of the tree. Where the overall entropy of a decision tree is given by the weighted average for the entropy of its leaf nodes

Exercise: samples classification

John Smith		**Jack Brawn**		**Peter White**	
Drinker:	No	Drinker:	Yes	Drinker:	No
Smoker:	No	Smoker:	No	Smoker:	No
Weight:	95	Weight:	105	Weight:	80
Died:	68	Died:	65	Died:	95

K-Means Clustering

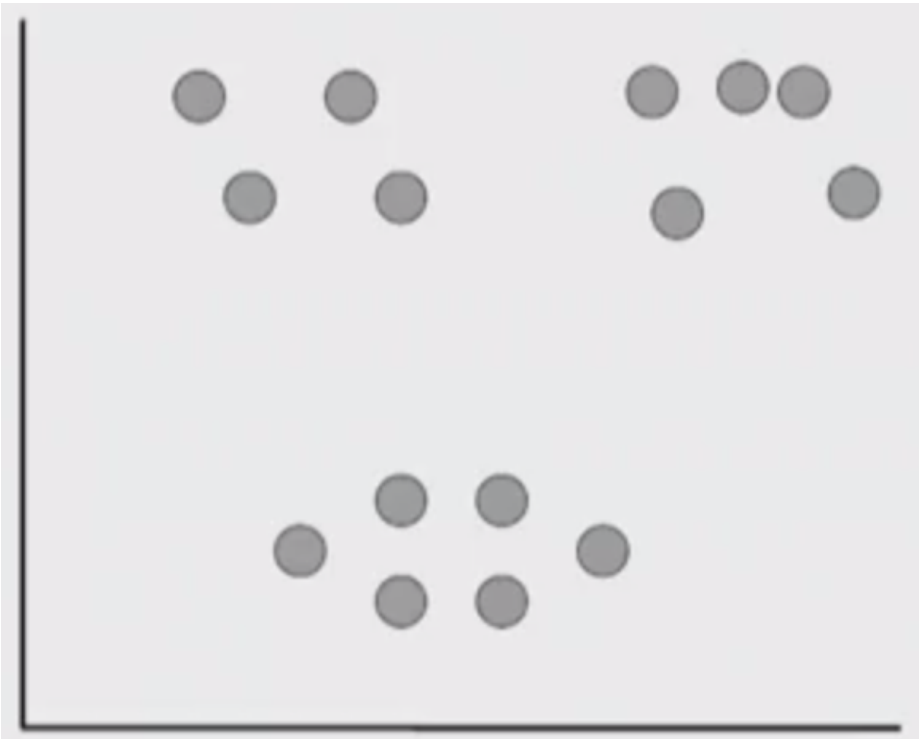
Clustering is concerned with grouping instances into clusters. Instances in one cluster should be similar to each other and dissimilar to instances in other clusters. Clustering uses unlabeled data and, hence, requires an unsupervised learning technique.

Info

Clustering is only indirectly related to process discovery. Nevertheless, clustering can be used as a preprocessing step for process mining. By grouping similar cases together it may be possible to construct partial process models that are easier to understand. If the process model discovered for all cases is too complex to comprehend, then it may be useful to first identify clusters and then discover simpler models per cluster.

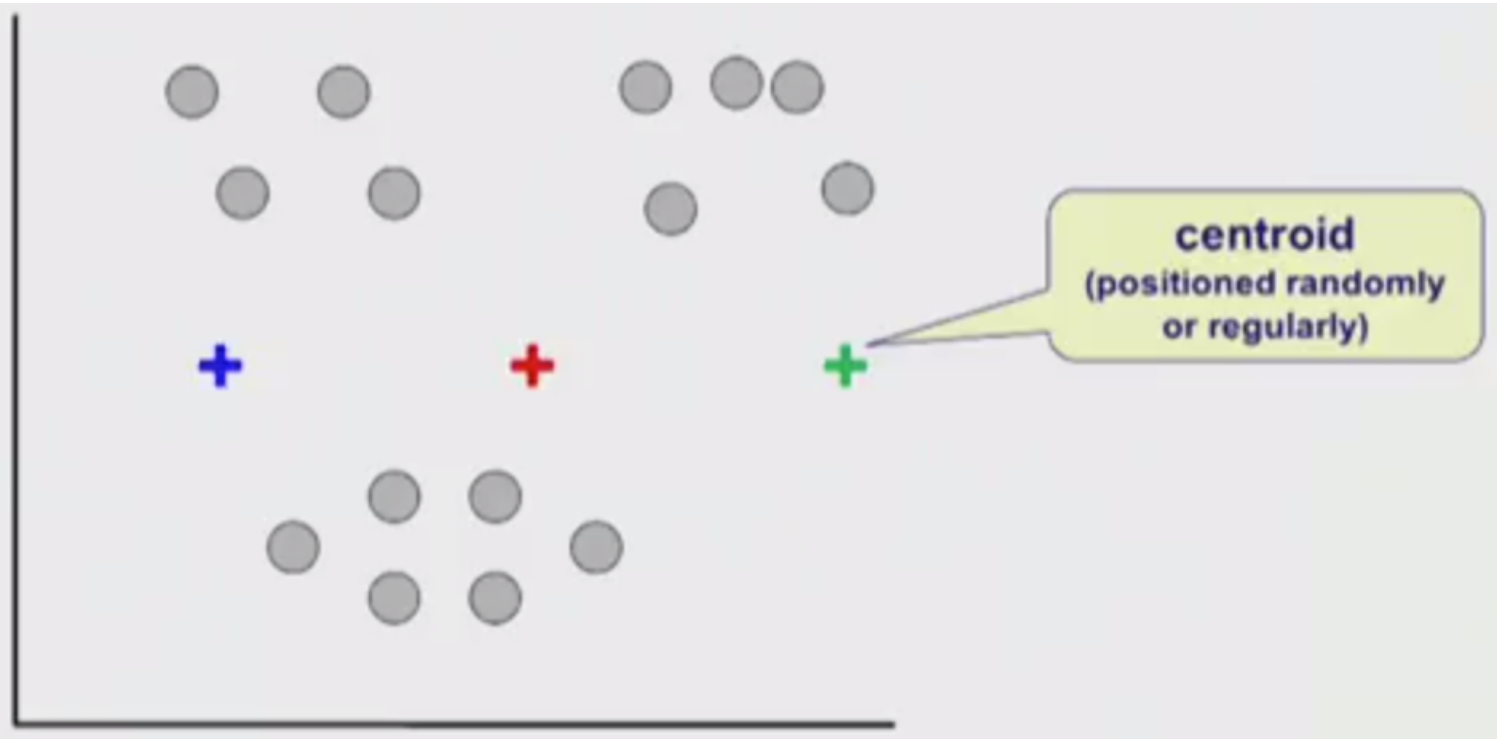
Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



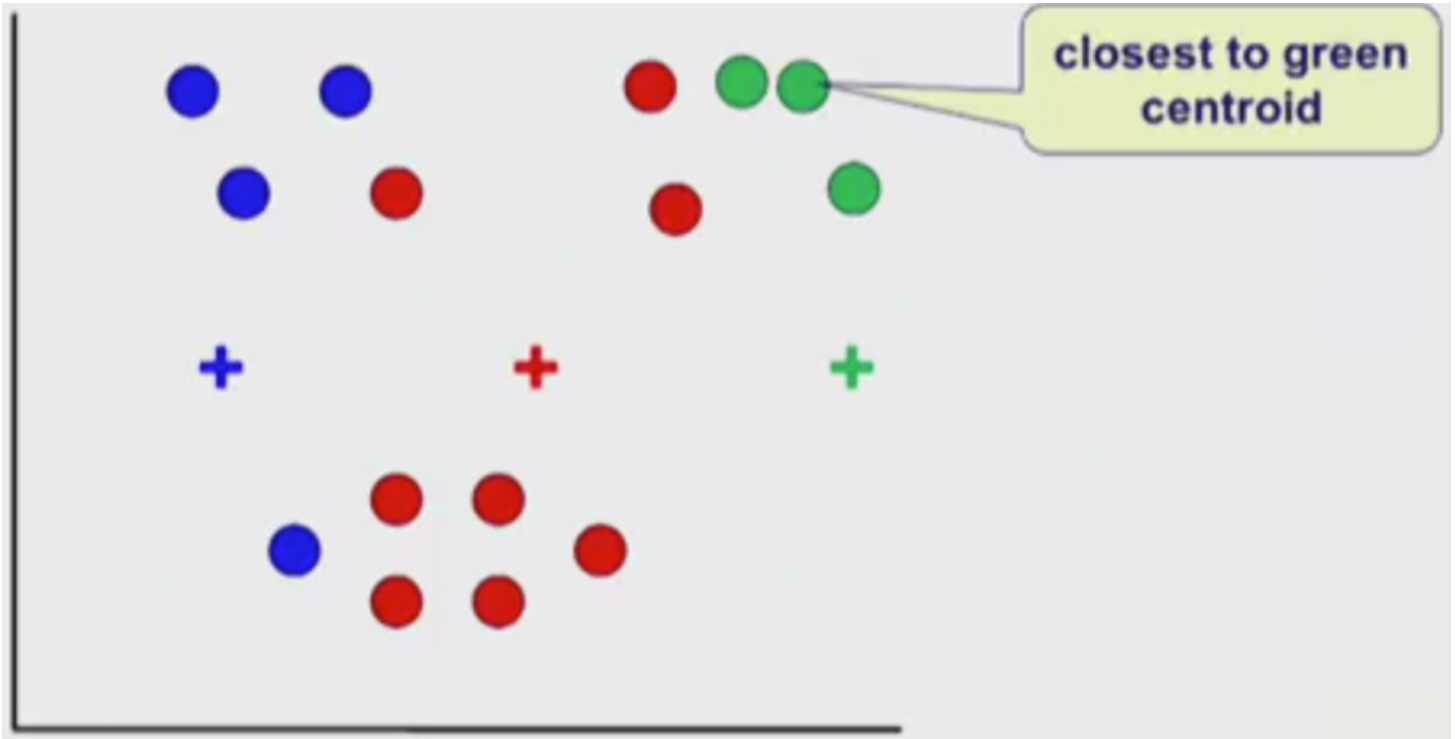
Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



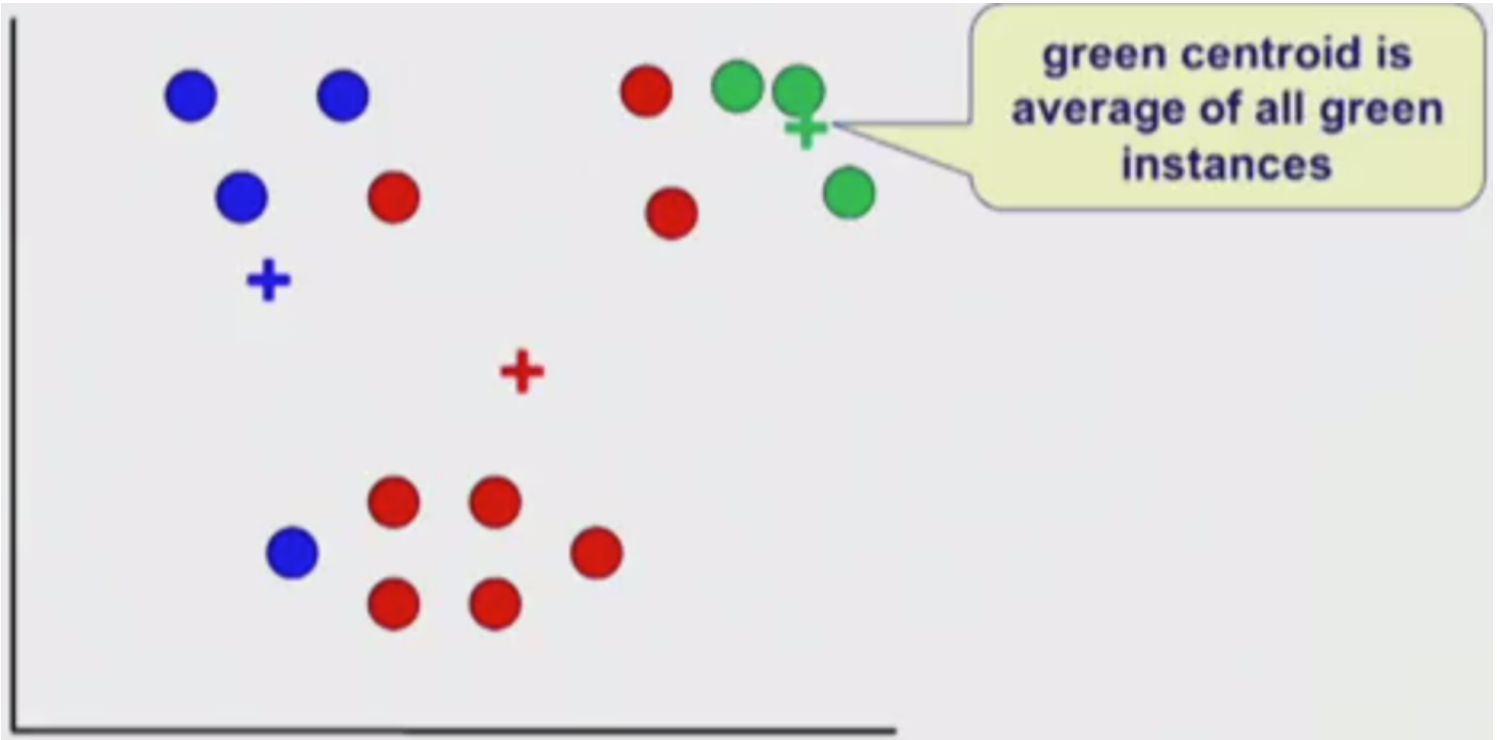
Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



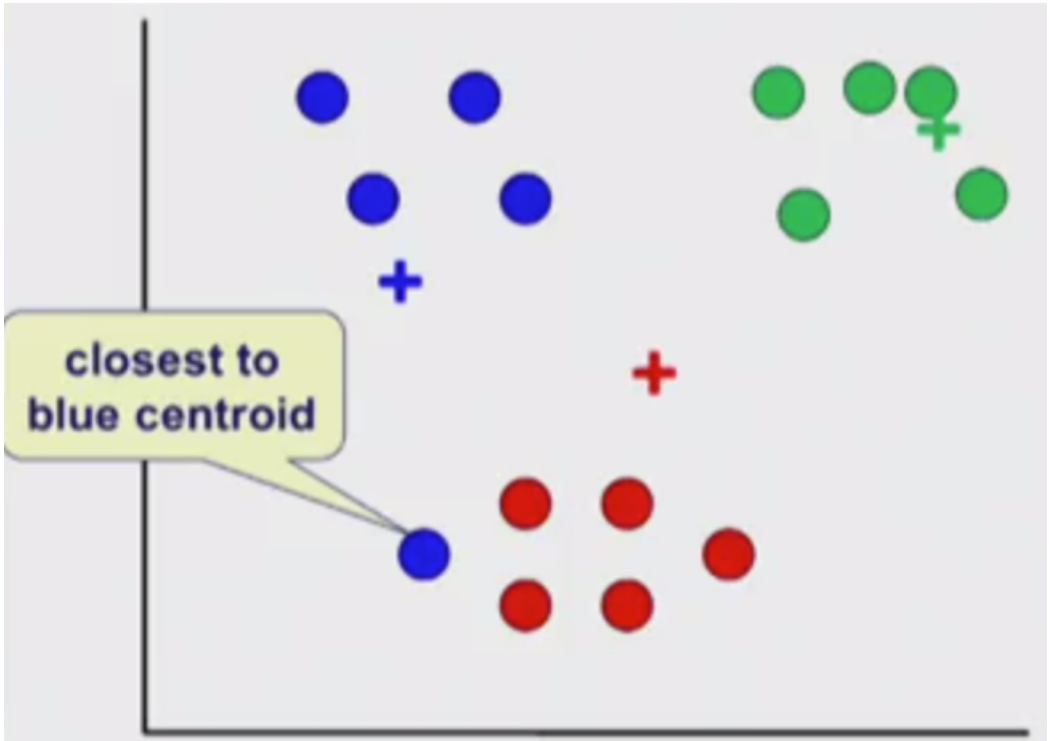
Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



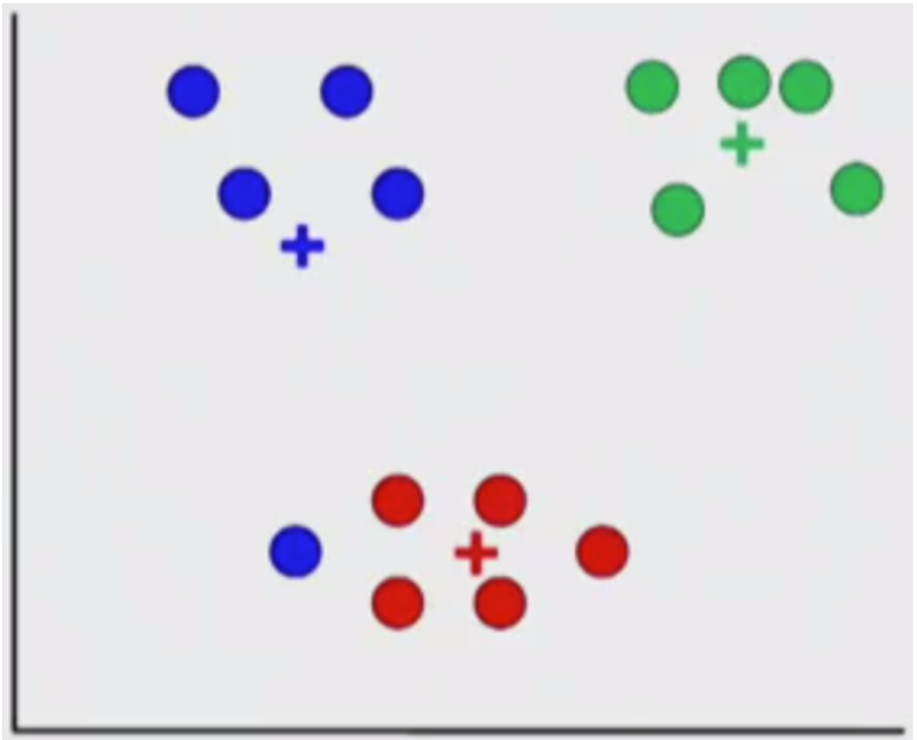
Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



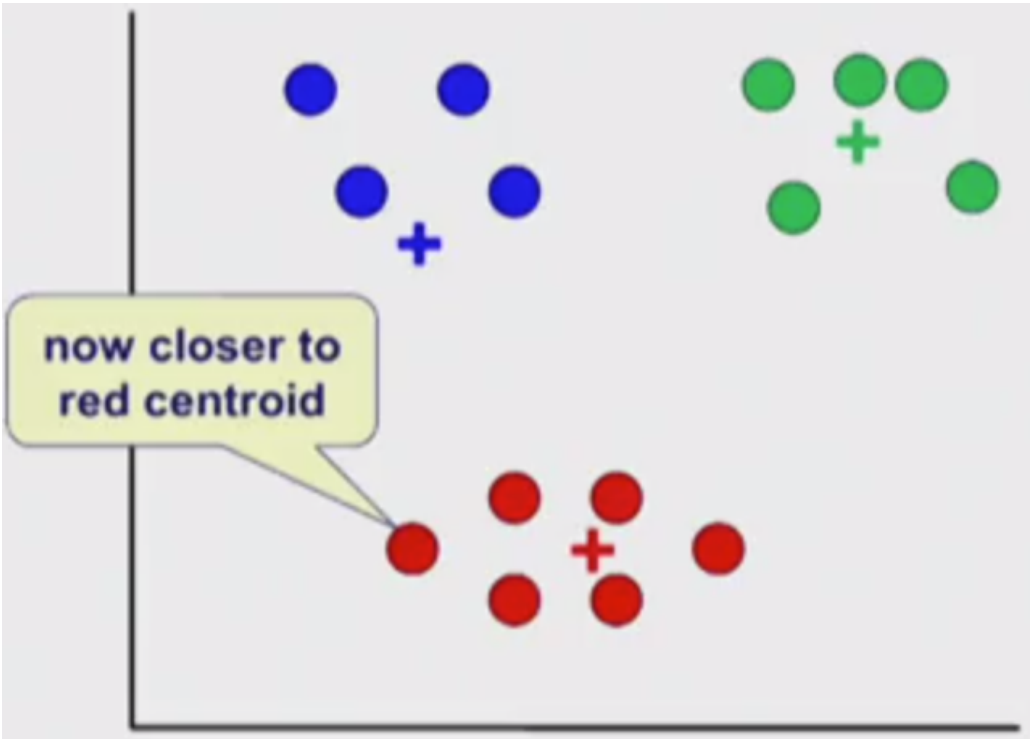
Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



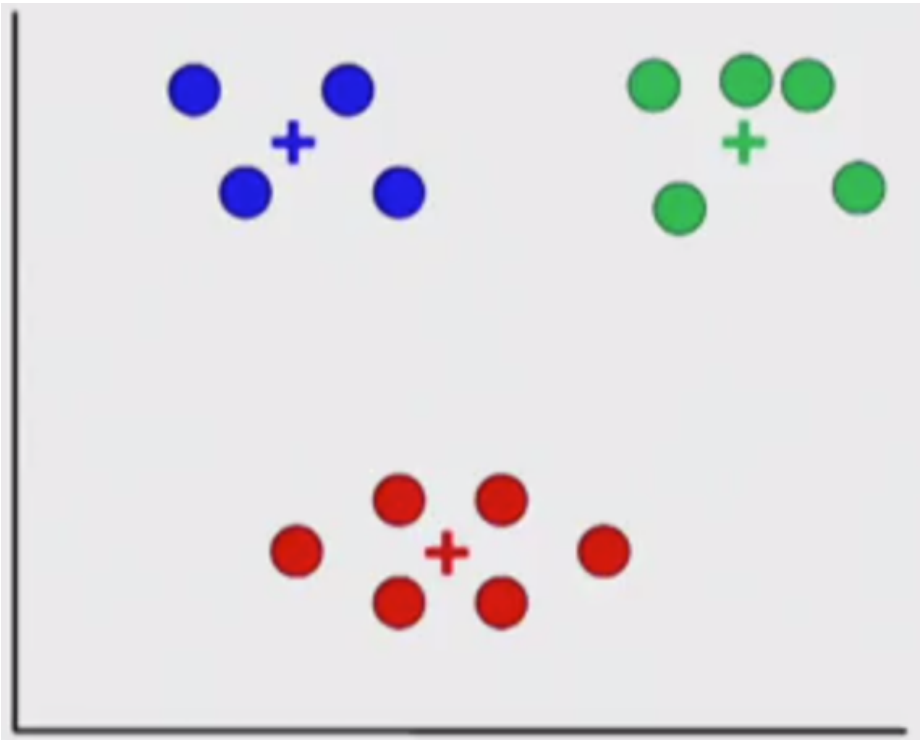
Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



Idea

Data points are placed on a n-dimensional cartesian hyperplan and euclidean distance is used.



- A fixed point has been reached. Nothing changes anymore.
- Non determinism generated from the initial random placement of controids. Experiments may be repeated many times.
- Main issue of k-means strategy is the up-front selection of the value k.

Association Rule Learning

Decision trees can be used to predict the value of some response variable that has been identified as being important. Driven by the response variable, rules like “people who drink and smoke die before 70” can be found.

Objective

Association rule learning aims at finding similar rules, with respect to decision tree, but now without focusing on a particular response variable. The goal is to find rules of the form “IF X THEN Y” where X is called the antecedent and Y the consequent (denoted as $X \Rightarrow Y$). X and Y can be any conjunction of “variable = value” terms. The only requirement is that X and Y are nonempty and any variable appears at most once in X and Y.

Association Rule Learning – Metrics

Metrics

Let N_X be the number of instances for which X holds. N_Y is the number of instances for which Y holds. $N_{X \wedge Y}$ is the number of instances for which both X and Y hold. N is the total number of instances. The following metrics are defined:

- support – indicates the applicability of the approach:

$$support(X \Rightarrow Y) = N_{X \wedge Y}/N$$

- confidence – indicates the reliability of the rule:

$$confidence(X \Rightarrow Y) = N_{X \wedge Y}/N_X$$

- lift – indicates the correlation of X and Y:

$$lift(X \Rightarrow Y) = \frac{N_{X \wedge Y}/N}{(N_X/N)(N_Y/N)} = \frac{N_{X \wedge Y} N}{N_X N_Y}$$

Association Rule Learning

Drinker	Smoker	Weight	Age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...	...	...	...

Rewriting it in terms of a item set we get:
$\{\{drinker, smoker\}, \{\}, \{drinker\}, \{drinker, smoker\}, \{smoker\}, \{\}, \ldots\}$

Association Rule Learning

Cappuccino	Latte	Espresso	Americano	Ristretto	Tea	Muffin	Bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Association Rule Learning

Cappuccino	Latte	Espresso	Americano	Ristretto	Tea	Muffin	Bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Rewriting it in terms of a item set we get:

$\{\{cappuccino, muffin\}, \{latte, muffin, bagel\}, \{espresso\},$
$\{cappuccino\}, \{tea, muffin\}, \{americano, ristretto\}, \ldots\}$

Association Rule Learning

Cappuccino	Latte	Espresso	Americano	Ristretto	Tea	Muffin	Bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Rewriting it in terms of a item set we get:

$\{\{cappuccino, muffin\}, \{latte, muffin, bagel\}, \{espresso\},$
$\{cappuccino\}, \{tea, muffin\}, \{americano, ristretto\}, \ldots\}$

Now assume to have a list of 240 orders distributed in the following manner:

$N_{tea} = 50$ $N_{latte} = 40$ $N_{muffin} = 40$

$N_{tea \wedge latte} = 50$ $N_{tea \wedge muffin} = 25$ $N_{tea \wedge latte \wedge muffin} = 15$

Association Rule Learning

Cappuccino	Latte	Espresso	Americano	Ristretto	Tea	Muffin	Bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Rewriting it in terms of a item set we get:
$\{\{cappuccino, muffin\}, \{latte, muffin, bagel\}, \{espresso\},$
$\{cappuccino\}, \{tea, muffin\}, \{americano, ristretto\}, \ldots\}$

Now assume to have a list of 240 orders distributed in the following manner:

$N_{tea} = 50$ $N_{latte} = 40$ $N_{muffin} = 40$
$N_{tea \wedge latte} = 50$ $N_{tea \wedge muffin} = 25$ $N_{tea \wedge latte \wedge muffin} = 15$

Let's derive metrics for $tea \wedge latte \Rightarrow muffin$, and $tea \Rightarrow muffin$

Generation of Association Rules

To systematically generate association rules, one typically defines two parameters: *minsup* and *minconf*. The support of any rule $X \Rightarrow Y$ should be above the thresholds (i.e. $support(X \Rightarrow Y) \geq minsup$, $confidence(X \Rightarrow Y) \geq minconf$).
Association rules can now be generated as follows:

1. Generate all frequent item-sets, i.e., all sets Z such that $N_Z/N \geq minsup$ and $|Z| \geq 2$
2. For each frequent item-set Z consider all partitionings of Z into two non-empty subsets X and Y . If $confidence(X \Rightarrow Y) \geq minconf$, then keep the rule $X \Rightarrow Y$. If $confidence(X \Rightarrow Y) < minconf$, then discard the rule
3. Output the rules found

Generation of Association Rules

To systematically generate association rules, one typically defines two parameters: *minsup* and *minconf*. The support of any rule $X \Rightarrow Y$ should be above the thresholds (i.e. $support(X \Rightarrow Y) \geq minsup$, $confidence(X \Rightarrow Y) \geq minconf$).
Association rules can now be generated as follows:

1. Generate all frequent item-sets, i.e., all sets Z such that $N_Z/N \geq minsup$ and $|Z| \geq 2$
2. For each frequent item-set Z consider all partitionings of Z into two non-empty subsets X and Y . If $confidence(X \Rightarrow Y) \geq minconf$, then keep the rule $X \Rightarrow Y$. If $confidence(X \Rightarrow Y) < minconf$, then discard the rule
3. Output the rules found

Unfortunately the algorithm is computationally prohibitive and generates many uninteresting rules

once you have, $tea \Rightarrow (latte \wedge muffin)$, no point in listing, $(tea \wedge latte) \Rightarrow muffin$

Apriori algorithm

Apriori algorithm

It permits to generate efficiently frequent item-set on the base of two main observation:

1. If an item-set is frequent then all of its non-empty subsets are also frequent. Formally, for any pair of non-empty item-sets X, Y : if $Y \subseteq X$ and $N_X/N \geq minsup$, then $N_Y/N \geq minsup$.
2. If, for any k, $\mathcal{I}_k$ is the set of all frequent item-sets with cardinality k and $\mathcal{I}_l = \emptyset$ for some l, then $\mathcal{I}_k = \emptyset$ for all $k \geq l$

Apriori algorithm

Apriori algebraic specification

1. Create $\mathcal{I}_1$. This is the set of singleton frequent item-sets
2. $k := 1$
3. If $\mathcal{I}_k = \emptyset$, then output $\cup_{i=1}^{k} \mathcal{I}_i$ and end. If $\mathcal{I}_k \neq \emptyset$ continue with the next step
4. Create $\mathcal{C}_{k+1}$ from $\mathcal{I}_k$. $\mathcal{C}_{k+1}$ is the candidate set containing item-sets of cardinality $k+1$. Note that one only needs to consider elements that are the union of two item-sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathcal{I}_k$ such that $|\mathcal{A} \cap \mathcal{B}| = k$ and $|\mathcal{A} \cup \mathcal{B}| = k+1$
5. For each candidate frequent item-set $c \in \mathcal{C}_{k+1}$: examine all subsets of c with k elements; delete c from $\mathcal{C}_{k+1}$ if any of the subsets is not a member of $\mathcal{I}_k$
6. For each item-set c in the pruned candidate frequent item-set $\mathcal{C}_{k+1}$, check whether c is indeed frequent. If so, add c to $\mathcal{I}_{k+1}$. Otherwise, discard c.
7. $k := k+1$ and return to step 3

Apriori algorithm

Apriori algebraic specification

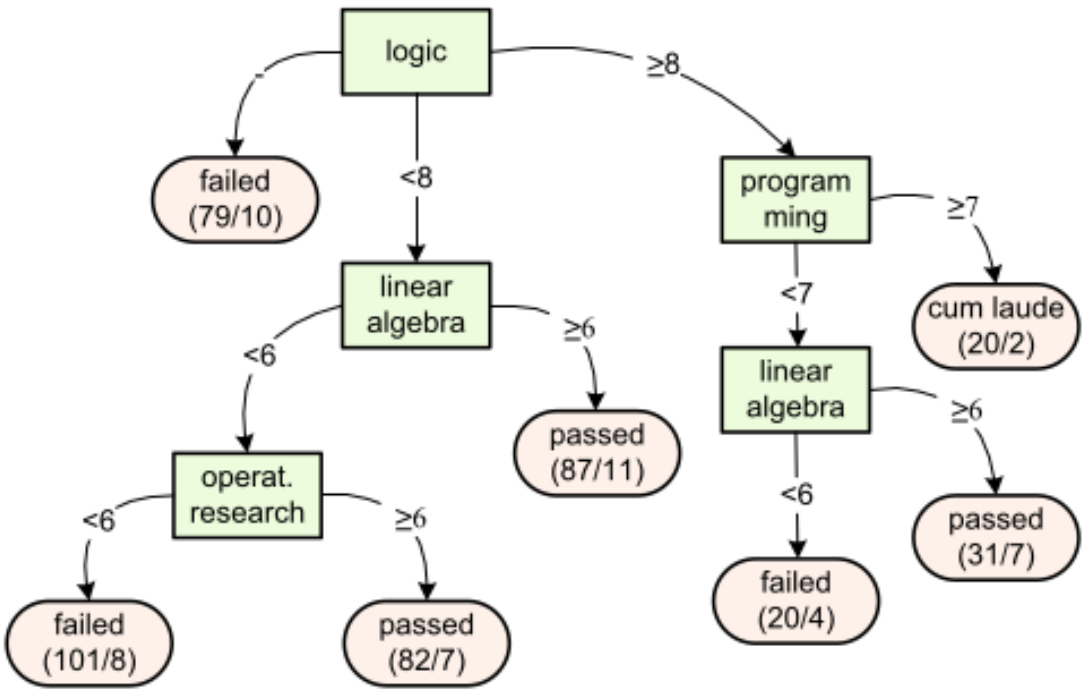
1. Create $\mathcal{I}_1$. This is the set of singleton frequent item-sets
2. $k := 1$
3. If $\mathcal{I}_k = \emptyset$, then output $\cup_{i=1}^{k} \mathcal{I}_i$ and end. If $\mathcal{I}_k \neq \emptyset$ continue with the next step
4. Create $\mathcal{C}_{k+1}$ from $\mathcal{I}_k$. $\mathcal{C}_{k+1}$ is the candidate set containing item-sets of cardinality $k+1$. Note that one only needs to consider elements that are the union of two item-sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathcal{I}_k$ such that $|\mathcal{A} \cap \mathcal{B}| = k$ and $|\mathcal{A} \cup \mathcal{B}| = k+1$
5. For each candidate frequent item-set $c \in \mathcal{C}_{k+1}$: examine all subsets of c with k elements; delete c from $\mathcal{C}_{k+1}$ if any of the subsets is not a member of $\mathcal{I}_k$
6. For each item-set c in the pruned candidate frequent item-set $\mathcal{C}_{k+1}$, check whether c is indeed frequent. If so, add c to $\mathcal{I}_{k+1}$. Otherwise, discard c.
7. $k := k+1$ and return to step 3

Many other proposal for Association rule learning are available in the literature

Performance of a Classifier

Given a data set consisting of N instances we know for each instance what the actual class is and what the predicted class is. For example, for a particular person that smokes, we may predict that the person will die young (predicted class is “young”), even though the person dies at age 104 (actual class is “old”). This can be visualized using a so-called confusion matrix.

Confusion Matrix



		predicted class: failed	predicted class: passed	predicted class: cum laude
actual class	failed	178	22	0
	passed	21	175	2
	cum laude	1	3	18

Elements on the diagonals are those correctly predicted (88%)

Confusion matrix and indexes

- t_p is the number of true positives, i.e., instances that are correctly classified as positive.
- f_n is the number of false negatives, i.e., instances that are predicted to be negative but should have been classified as positive.
- f_p is the number of false positives, i.e., instances that are predicted to be positive but should have been classified as negative.
- t_n is the number of true negatives, i.e., instances that are correctly classified as negative.

Performance measures for classifiers

- Error: $(f_p + f_n)/N$
- Accuracy: $(t_p + t_n)/N$
- t_p-rate: $t_p/(t_p + f_n)$
- f_p-rate: $f_p/(f_p + t_n)$
- Precision: $t_p/(t_p + f_p)$
- Recall: $t_p/(t_p + f_n)$
- F1 score:

$$\frac{2 \times precision \times recall}{precision + recall}$$

Performance measures for classifiers

- Error: $(f_p + f_n)/N$
- Accuracy: $(t_p + t_n)/N$
- t_p-rate: $t_p/(t_p + f_n)$
- f_p-rate: $f_p/(f_p + t_n)$
- Precision: $t_p/(t_p + f_p)$
- Recall: $t_p/(t_p + f_n)$
- F1 score:

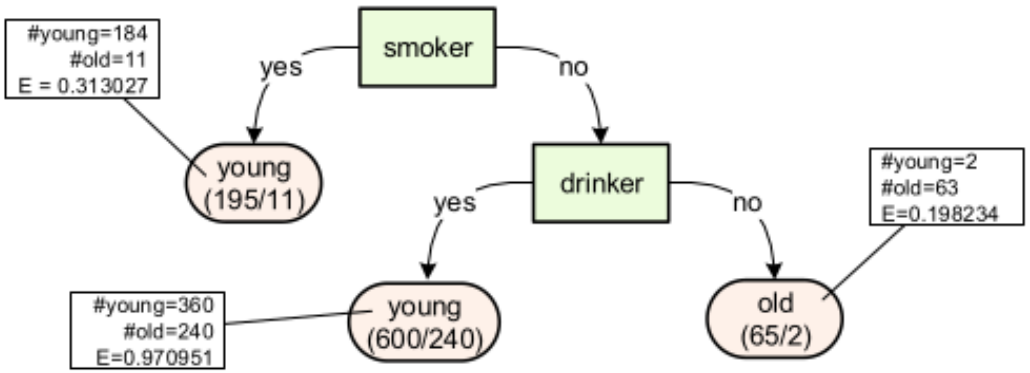
$$\frac{2 \times precision \times recall}{precision + recall}$$

		predicted class: young	predicted class: old
actual class	young	546	0
	old	314	0

Performance measures for classifiers

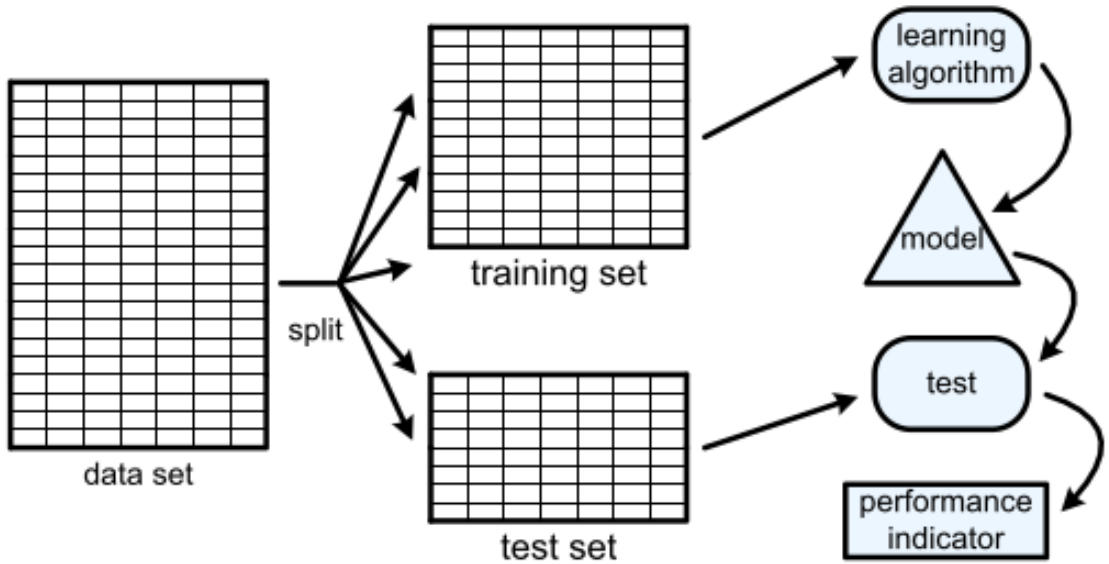
- Error: $(f_p + f_n)/N$
- Accuracy: $(t_p + t_n)/N$
- t_p-rate: $t_p/(t_p + f_n)$
- f_p-rate: $f_p/(f_p + t_n)$
- Precision: $t_p/(t_p + f_p)$
- Recall: $t_p/(t_p + f_n)$
- F1 score:

$$\frac{2 \times precision \times recall}{precision + recall}$$

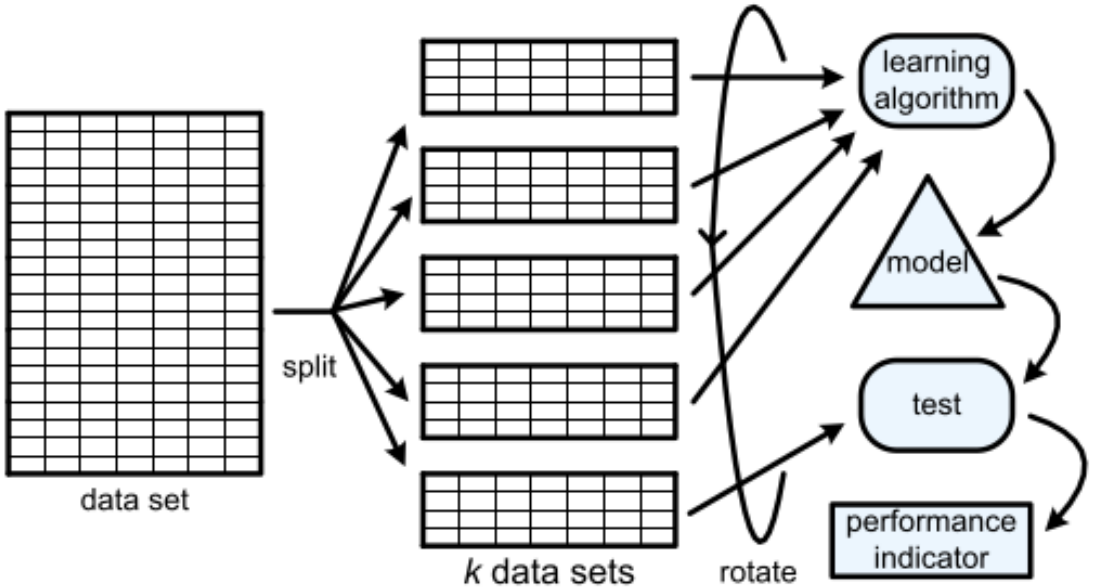


		predicted class: young	predicted class: old
actual class	young	544	2
	old	251	63

Cross-Validation



k-fold Cross Validation



Quality of results

Evaluating the quality of data mining results is far from trivial. We now discuss some additional complications that are also relevant for process mining.

- Consider the sequence 2,3,5,7,9,11, ...

Quality of results

Evaluating the quality of data mining results is far from trivial. We now discuss some additional complications that are also relevant for process mining.

- Consider the sequence 2,3,5,7,9,11, . . . What is the next number?

Quality of results

Evaluating the quality of data mining results is far from trivial. We now discuss some additional complications that are also relevant for process mining.

- Consider the sequence 2,3,5,7,9,11, . . . What is the next number?
- Inductive bias: refers to a preference for one solution rather than another which cannot be determined by the data itself but which is driven by external factors
- Representational bias: refers to choices that are implicitly made by selecting a particular representation
- Learning bias refers to strategies used by the algorithm that give preference to particular solutions

Occam's Razor

a principle attributed to the 14th-century English logician William of Ockham. The principle states that "one should not increase, beyond what is necessary, the number of entities required to explain anything"

- look for the "simplest model" that can explain what is observed in the data set
- finding a natural balance between overfitting and underfitting.