Data Analysis: Identifying Groups in Data

Institute of Technology Tallaght

Department of Computing

Types of machine learning

Machine learning algorithms for data analysis can be classified into two big groups:

- **supervised learning** - algorithms used for building predictive models and they involve:
  - the use of data where the values of the variable to be predicted are known for the **training models**
  - the use of other data where the values of the variable to be predicted are known for **testing and validation** of models
  - the use of models for **prediction**
- **unsupervised learning** - algorithms that
  - discover relationships or other information not immediately evident in the data
  - do not predict anything
  - for example, perform **clustering** or the discovery of **association rules**

Grouping Data

	Clustering	**Association rules**
Grouped entity type	Data instance	Value of categorical variable
Grouping criterion	Some measure of distance	Frequency of co-occurrence

The kind of groupings we are looking at here are those discovered by *unsupervised learning*, which is a machine learning term for analysis methods that:

- do not build models and hence do not involve a training stage or target variable prediction
- consist of algorithms that are applied to data in order to extract some useful information
- typically result in whole-set insights, into particular aspects of the complex distribution of attribute values in the set

	Clustering	Association rules
Grouped entity type	Data instance	Value of categorical variable
Grouping criterion	Some measure of distance	Frequency of co-occurrence

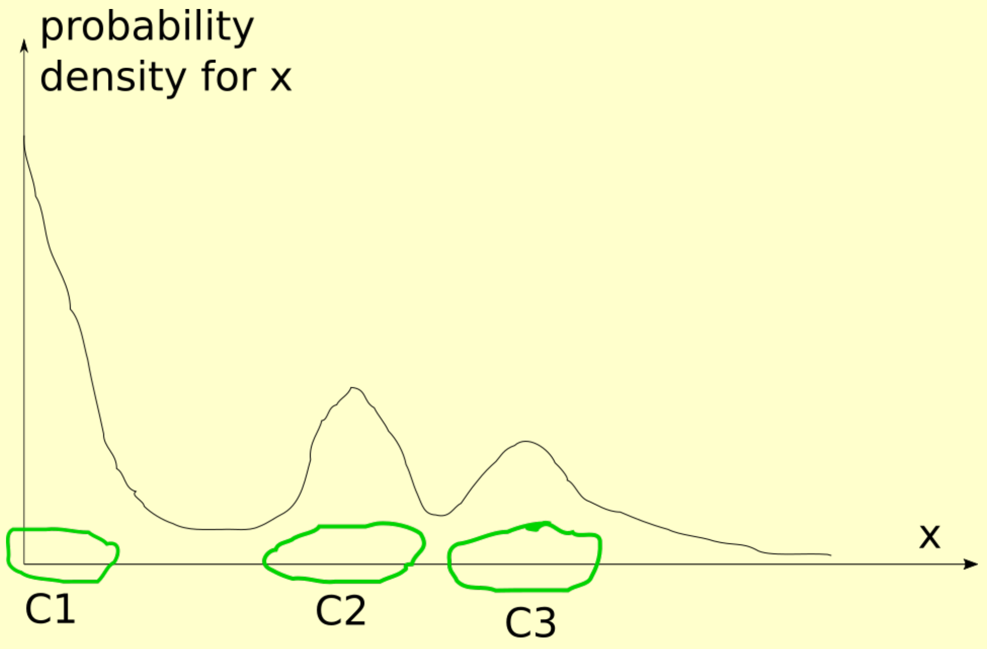
Clustering

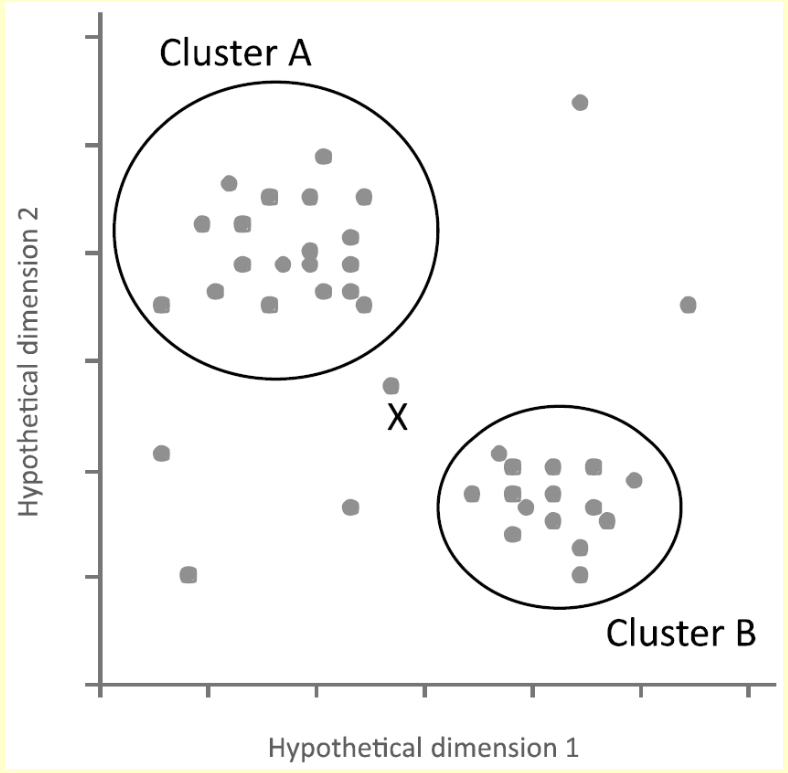
- Clustering methods discover groups of data instances in a set that are similar in terms of their attribute values
- All clustering methods can be categorized as
  - *hierarchical*, with a bottom-up approach where the data set is progressively grouped into clusters - the number of clusters does not need to be known in advance
  - *non-hierarchical*, where an algorithm divides the data set into a pre-set number of clusters - the best number of clusters is decided empirically, using measures of cluster quality for a range of cluster counts
- All clustering algorithms depend on the concept of *distance* between data instances:
  - e.g. Euclidean distance in the case of numeric attributes
  - e.g. Jaccard distance in the case of categorical attributes
  - algorithms have also been defined for calculating distances when there is a mix of numeric and categorical attributes in the data

*The picture below left shows what clusters would look like in a single-attribute data distribution graph. Here x is the only attribute and the vertical axis shows the probability density of x. The density has three 'humps' associated with ranges of **x** marked C1, C2 and C3. These are the three clusters in the data set.*

*The picture below right illustrates clusters using a data set represented by two attributes. There are two clusters, named A and B. The data instance **x** is an **outlier** as it doesn't clearly belong to either of the clusters.*

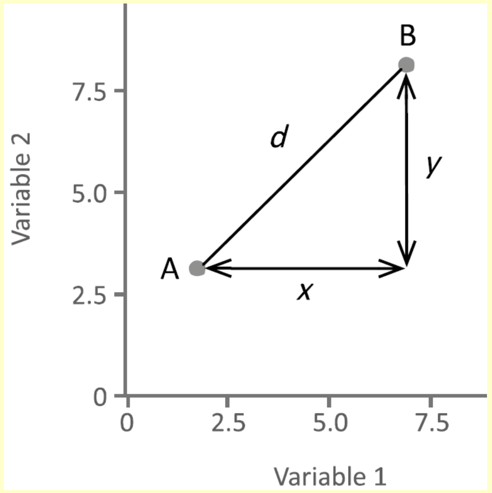
Both of the examples are trivial and meant for illustration only. With real data sets, many attributes would be used for clustering and the groups defined in a multi-dimensional plane.





Source: [MSD]

Euclidean distance



Source: [MSD]

- In a two- or three-dimensional space this is simply a straight line between two points in the space
- In the Cartesian co-ordinate system the distance can be calculated using Pythagoras' theorem
- The distance between the points in the picture, A(2, 3) and B(7, 8), can be calculated as:

$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} = \sqrt{(7-2)^2 + (8-3)^2} = \sqrt{50} = 7.07$$

	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	**...**	$\boldsymbol{x_{\mathrm{p}}}$
i_1	x_{11}	x_{21}	...	$x_{\mathrm{p}1}$
i_2	x_{12}	x_{22}	...	$x_{\mathrm{p}2}$
.	.	.		.
.	.	.		.
.	.	.		.
i_{n}	$x_{1\mathrm{n}}$	$x_{2\mathrm{n}}$	...	x_{pn}

- In the general case, the Euclidean distance between data instances can be calculated for any number of dimensions representing attributes in the data set:

$$d_{Ejk} = \sqrt{\sum_{i=1}^{p} (x_{ik} - x_{ij})^2}$$

where x_1, x_2... x_p are numeric attributes and d_{Ejk} is the Euclidean distance between instances j and k.

Jaccard distance

- The Jaccard distance is one example of how distance can be quantified between data instances based on categorical attribute values
- In order for distances to be calculated, categorical data with more than two possible values is transformed into dichotomous data, by turning each value into a dichotomous attribute (for example, if a **colour** attribute had values **red**, **green** and **blue**, it would be transformed into three dichotomous attributes, each with possible values **0** or **1**)

	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	...	$\boldsymbol{x_p}$
i_1	x_{11}	x_{21}	...	x_{p1}
i_2	x_{12}	x_{22}	...	x_{p2}
.	.	.		.
.	.	.		.
.	.	.		.
i_n	x_{1n}	x_{2n}	...	x_{pn}

	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	$\boldsymbol{x_6}$
i_j	0	1	0	1	0	1
i_k	1	1	0	0	0	0

- Now, the Jaccard distance can be defined as:

$$d_{Jjk} = \frac{C_{01jk} + C_{10jk}}{C_{01jk} + C_{10jk} + C_{11jk}}$$

where C_{XYjk} is the number of attributes that have value X in instance j and value Y in instance k and d_{Jjk} is the Jaccard distance between instances j and k. X and Y are binary values (0 or 1).

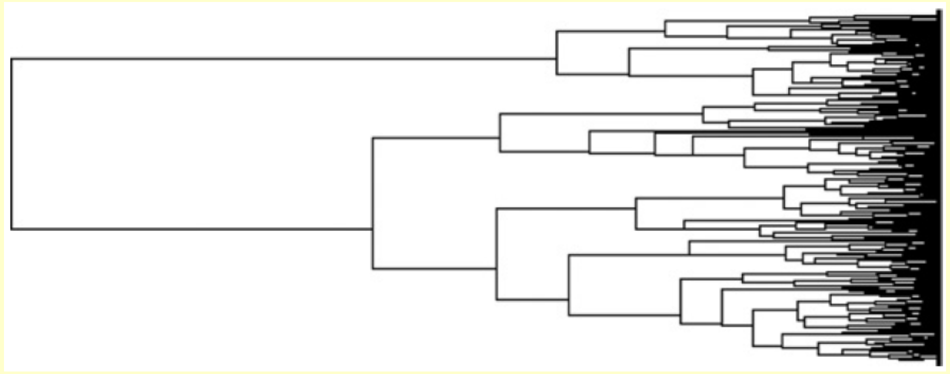
- In the example shown on the left, the Jaccard distance is:

$$d_{Jjk} = \frac{1+2}{1+2+1} = 0.75$$

Agglomerative hierarchical clustering

- Agglomerative hierarchical clustering is an example of hierarchical clustering
- It is a bottom up approach with the following high-level algorithm:
  1. initially each instance in the data set is declared to be a cluster in itself (there are n clusters, if n is the number of data instances in the data set)
  2. the distance between each pair of clusters is calculated and the two clusters that are the closest are joined into a new cluster (the number of clusters is reduced by 1)
  3. if the number of clusters is greater than 1, step 2 is repeated

  Note: Step 2 is performed $n-1$ times. An algorithm for distance to and from a cluster is used (lowest, highest or average).


Source: [MSD]

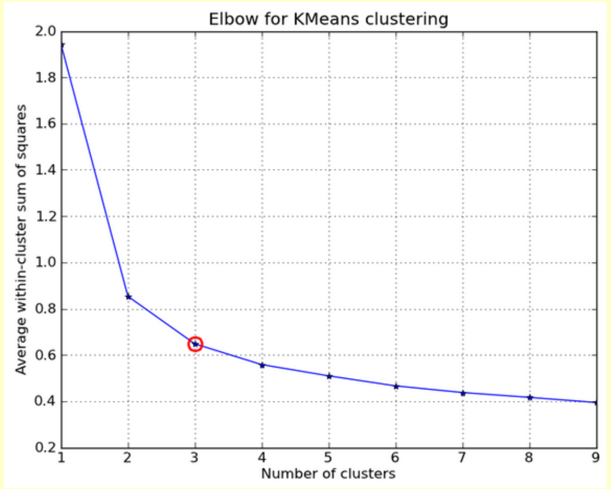
- The number of clusters not set. Division into any number of clusters between 1 and n contained in result.
- The resultant tree can be shown as a *dendrogram* (*dendron*: *tree* in Greek)

k-Means clustering

- k-Means clustering is an example of non-hierarchical clustering
- The algorithm is applied with prior knowledge of the number of clusters that should be created and can be described as follows:
  1. the number of clusters, k, is set
  2. k arbitrary points are chosen in the data-instance space (that defined by the data attributes) as initial **cluster representatives** C_1, C_2,...C_k
  3. for each instance in the data set, the closest cluster representative is determined and the instance assigned to cluster j, where C_j is the closest cluster representative
  4. if no instance changes cluster (in 2nd or subsequent iteration), the process is finished
  5. if some instances moved to a new cluster, the cluster representatives are re-calculated so that each attribute gets the value average for that attribute among the data instances in the cluster (i.e. the centroid of the instances in the cluster becomes the cluster representative)
  6. repeat from step 3

Deciding on the optimal number of clusters

- The optimal number of clusters is the highest number k for which the k clusters are significantly better in quality than the $k-1$ clusters arrived at using the same algorithm
- Different measures are used for this but they are all based around the sum of within-cluster squares: $S_{sawc} = \sum_{i=0}^{n} d^2_{iC(i)}$, where n is the number of instances in the set and $d_{iC(i)}$ is the distance between instance i and the representative of the cluster it belongs to, $C(i)$



Source: stackoverflow, user: Amro

The picture shows an elbow diagram for the well-known iris data set, containing data for 3 different types of iris. In this case we know that 3 is the right number of clusters and the elbow diagram reflects this.

- The value of the within-cluster sum of squares is plotted against k and the value of k at which the negative slope becomes significantly less steep is the optimal value of k
- Often the elbow-like shape, after which the method is named *the elbow method*, is visually distinguishable in the graph
- The method is not exact but does not need to be, since what is required is some indication of what value of k provides a good balance between model complexity (in terms of the number of clusters) and model quality (in terms of the 'compactness' of the clusters)

Association rules

- Association rules state the expectation of particular values of particular attributes ocurring together in a data instance
- The following is an example of an association rule:

```
IF customer is between 18 and 20 AND
the customer buys a towel AND
the customer buys a bath mat
THEN the customer buys a shower curtain
```

A small section of a data set for which that rule is defined, might look something like this:

TransactionId	**Customer age**	...	**Towel**	**Bath mat**	**Shower curtain**	...	**Chewing gum**	...	**Milk**	**Biscuits**	...
123456	20	...	1	1	1	...	1	...	0	0	...
123457	65	...	0	0	0	...	0	...	1	1	...
123458	26	...	0	0	0	...	1	...	1	0	...

Each data instance (line in the table) represents a retail transaction in, let's say, a supermarket. The items sold in the shop are all represented with binary attributes, set to 0 if the item is absent from a transaction and to 1 if the item is present.

- But, how are these rules arrived at? Some **measures** indicating levels of occurrence, co-ocurrence and likelihood of co-occurrence not being accidental are calculated for all the possible combinations of attributes, then the combinations with measure values indicative of association are stated in the form of association rules.

  Before we define the measures of interest, let us name the parts of an association rule:

  → The **antecedent** is the condition in the rule i.e. the part between the IF and THEN. In the example, this is:

  ```
  customer is between 18 and 20 AND the customer buys a towel
  AND the customer buys a bath mat
  ```

  → The **consequent** is the part that is true if the condition is fulfilled i.e. the part after THEN. In the example, this is:

  ```
  the customer buys a shower curtain
  ```

  The most commonly used measures are:

  ✓ **Support**: the probability of some combination of items occurring together. In particular, of interest is the *support for the rule*, i.e. the probability of all the attribute values that appear in the rule occurring together in a data instance:

$$P(x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots x_p \in S_p)$$

where $P(A)$ is the probability of A, x_1, x_2 ... x_p are the attributes of the data and S_1, S_2 ... S_p are the sets of allowed values (one for each attribute) that define the combination of attribute values present in the rule. $\wedge$ is the logical AND operator. For the example, the support is:

$$P(customer\_age \in \{18, 19, 20\} \wedge towel \in \{1\} \wedge bath\_mat \in \{1\} \wedge shower\_curtain \in \{1\} \\ \wedge chewing\_gum \in \{0, 1\} \wedge milk \in \{0, 1\} \wedge biscuits \in \{0, 1\})$$

The expression includes all the attributes, but the ones that are not of interest are let have any value (e.g. `biscuits` could be there or not but we don't care, because we are looking for a relationship between `customer_age`, presence of `towel`, presence of `bath_mat` and presence of `shower_curtain`, so `biscuits` is permitted any value i.e. 0 or 1). By leaving out the attributes that we don't care about and using more intuitive notation, the value for the support can be re-written as:

$$P((18 \leq customer\_age \leq 20) \wedge (towel = 1) \wedge (bath\_mat = 1) \wedge (shower\_curtain = 1))$$

✓ **Confidence**: the probability of the items in the combination occurring together, given the antecedent:

$$P(x_c \in S_c \,|\, x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots \boldsymbol{x_{c-1} \in S_{c-1} \wedge x_{c+1} \in S_{c+1}} \wedge \ldots x_p \in S_p) =$$

$$\frac{P(x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots x_p \in S_p)}{P(x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots \boldsymbol{x_{c-1} \in S_{c-1} \wedge x_{c+1} \in S_{c+1}} \wedge \ldots x_p \in S_p)}$$

where $P(A|B)$ is the probability of A, given B, x_1, x_2 ... x_p are the attributes of the data and S_1, S_2 ... S_p are the sets of allowed values (one for each attribute) that define the combination of attribute values present in the rule. The attribute x_c represents the attribute that features in the *consequent* of the rule and S_c is the set of values that the consequent allows for this attribute (the consequent could feature more than one attribute but we will only look at the case where one attribute is involved). It is absent from the combination of attribute values that define the *antecedent* (the part of the attribute list from which x_c is missing is shown in boldface). The confidence is calculated by dividing the support for the entire rule by the support for the antecedent (the part of the equation after the equal sign).

The confidence for the example is:

$$P(shower\_curtain = 1 \,\big|\, (18 \leq customer\_age \leq 20) \wedge (towel = 1) \wedge (bath\_mat = 1))$$

✓ **Lift**: the ratio between the actual probability of the items in the combination occurring together and the the probability of the co-occurrence happening by chance :

$$\frac{P(x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots x_p \in S_p)}{P(x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots \boldsymbol{x_{c-1} \in S_{c-1} \wedge x_{c+1} \in S_{c+1}} \wedge \ldots x_p \in S_p)P(x_c \in S_c)}$$

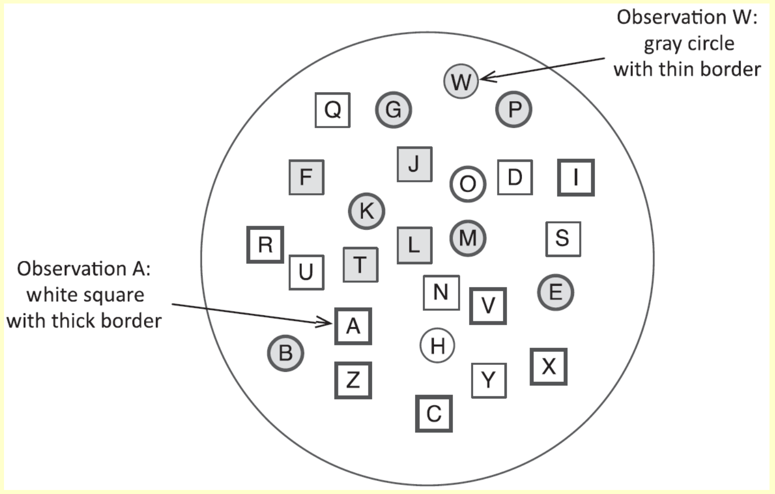
Three probabilities feature in the formula, each the *support* for different part of the rule:

- the entire rule: $P(x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots x_p \in S_p)$
- the antecedent: $P(x_1 \in S_1 \wedge x_2 \in S_2 \wedge \ldots \boldsymbol{x_{c-1} \in S_{c-1} \wedge x_{c+1} \in S_{c+1}} \wedge \ldots x_p \in S_p)$
- the consequent: $P(x_c \in S_c)$

The lift for the example is:

$$\frac{P((18 \leq customer\_age \leq 20) \wedge (towel = 1) \wedge (bath\_mat = 1) \wedge (shower\_curtain = 1))}{P((18 \leq customer\_age \leq 20) \wedge (towel = 1) \wedge (bath\_mat = 1))P(shower\_curtain = 1)}$$

- Association rules are mined from the data by setting required levels for the different measures, for example 50% rule support, 50% confidence and a lift of 3, then searching for rules that meet these minimal requirements.
- The **apriori** method is based around the fact that support for a value combination is always the same or lower than that of any of its subset combinations. It starts with a search for single-value combinations that meet the support level, progressing onto 2-value combinations that are supersets of the retained 1-value combinations and so on.


Observation W: gray circle with thin border
Observation A: white square with thick border
Q
G
W
P
F
J
O
D
I
K
R
U
T
L
M
S
N
V
E
A
H
B
Z
Y
X
C

Source: [MSD]

The picture shows a set of elements with three attributes: ***`shape`***, ***`color`*** *and* ***`border`*** *with respective value sets* `{circle, square}`, `{white, gray}` *and* `{thin, thick}`*. The elements represent data instances.*

A rule for this data set and associated measures are as follows:

```
IF colour = gray AND
shape = circle
THEN border = thick
```

- ***support***: *6/26 (there are 6 out of 26 elements have the three values included in the definition of the rule:* `B, E, G, K, M, P`*)*
- ***confidence***: *6/7 (we divide the support for the rule 6/26 with the support for the antecedent, which is 7/26, as the antecedent is true for element* `W` *in addition to those that support the entire rule)*
- ***lift***: $\dfrac{6/26}{(7/26)\times(14/26)} = 1.59$

  The lift value means that the antecedent and consequent occur together 1.59 times more often than they would by pure chance.

References

The pictures in this presentation were taken from the following books. The source for each picture is cited beside it.

[MSD] *Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining*, by Glenn J. Myatt and Wayne P. Johnson, John Wiley & Sons, 2014.