Machine Intelligence

Lecture 10: Clustering

Thomas Dyhre Nielsen

Aalborg University

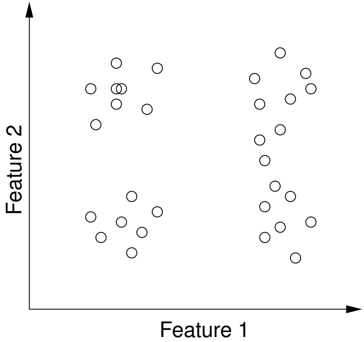
Tentative course overview

Topics:

- Introduction
- Search-based methods
- Constrained satisfaction problems
- Logic-based knowledge representation
- Representing domains endowed with uncertainty.
- Bayesian networks
- Inference in Bayesian networks
- Machine learning: Classification
- **Machine learning: Clustering**
- Planning
- Multi-agent systems

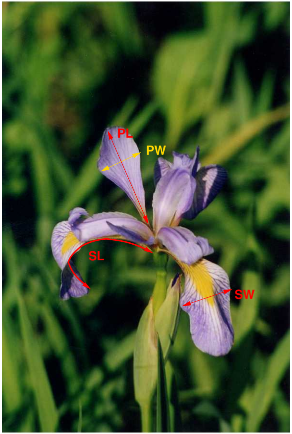
Clustering

The objective of clustering is to find structure in the data.



Examples:

- Based on customer data, find groups of customers with similar profiles.
- Based on image data, find groups of images with similar motif.
- Based on article data, find groups of articles with the same topics.
- ...



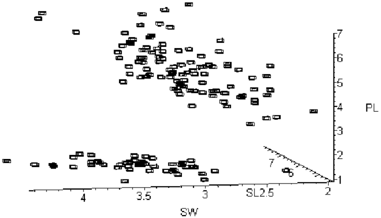
Measurement of petal width/length and sepal width/length for 150 flowers of 3 different species of Iris.

first reported in:
Fisher,R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7 (1936).

	Attributes			**Class variable**
SL	SW	PL	PW	Species
5.1	3.5	1.4	0.2	Setosa
4.9	3.0	1.4	0.2	Setosa
6.3	2.9	6.0	2.1	Virginica
6.3	2.5	4.9	1.5	Versicolor
...	...	...	...	...

The Iris data with class labels removed:

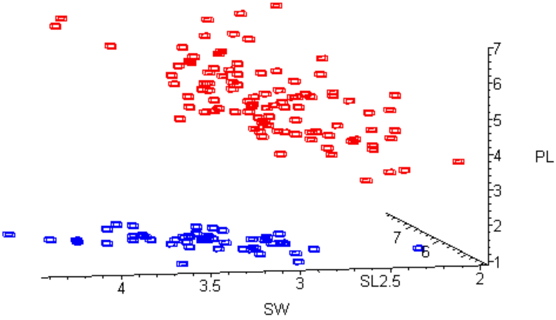
	Attributes		
SL	SW	PL	PW
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
6.3	2.9	6.0	2.1
6.3	2.5	4.9	1.5
...	...	...	...



Clustered Iris

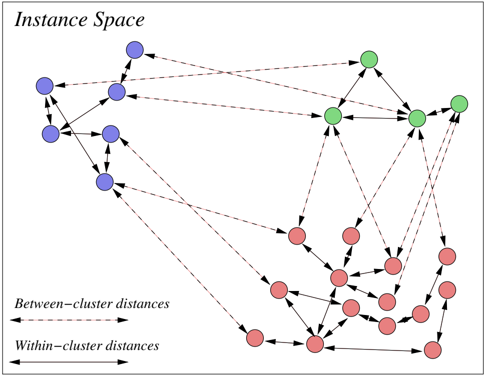
A clustering of the data $S = \mathbf{a}_1, \ldots, \mathbf{a}_N$ consists of a set $C = \{c_1, \ldots, c_k\}$ of *cluster labels*, and a *cluster assignment ca* : $S \to C$.

Clustering Iris with $C = \{$*blue*, *red*$\}$:



The k-means algorithm

Distance Function and Clustering



A candidate clustering (indicated by colors) of data cases in instance space. Arrows indicate between- and within-cluster distances (selected).

General goal: find clustering with

- large between-cluster variation (sum of between-cluster distances)
- small within-cluster variation (sum of within-cluster distances)

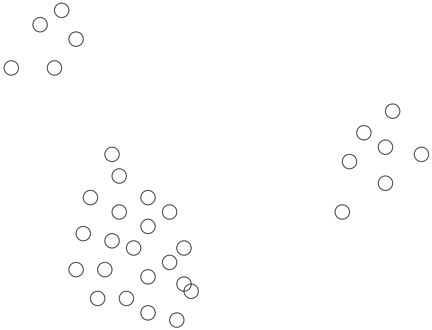
We consider the scenario, where

- the number k of clusters is known.
- we have a distance measure $d(\mathbf{x}_i, \mathbf{x}_j)$ between pairs of data points (feature vectors).
- we can calculate a centroid for a collection of data points $S = \{\mathbf{x}_1, \dots \mathbf{x}_n\}$.

Initialize: *randomly pick* k *data points as initial cluster centers* $\mathbf{c} = c_1, \dots, c_k$ from S
`repeat`
    Form k clusters by assigning each point in S to its closest centroid.
    Recompute the centroid for each cluster.
`until` Centroids do not change

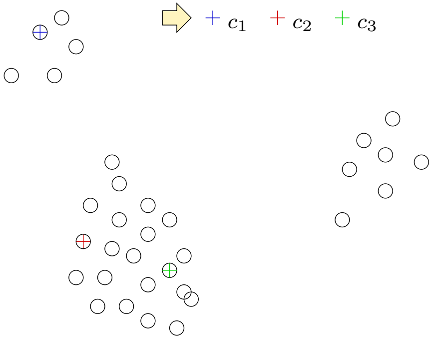
The k-means algorithm: Example

$k = 3$:

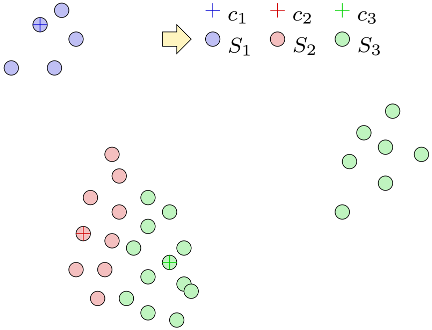


The k-means algorithm: Example

$k = 3$:

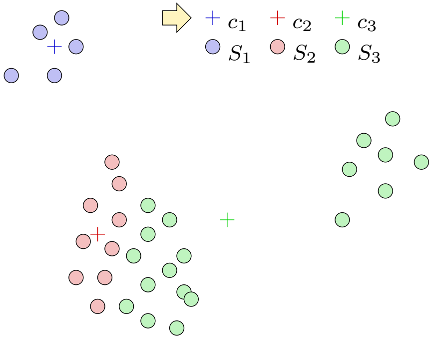


$k = 3$:



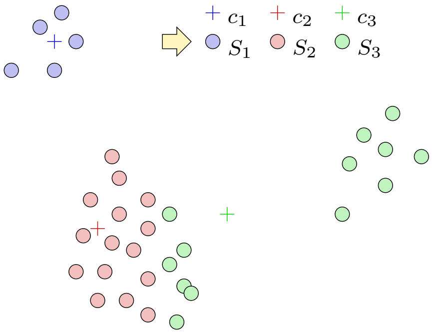
The k-means algorithm: Example

$k = 3$:

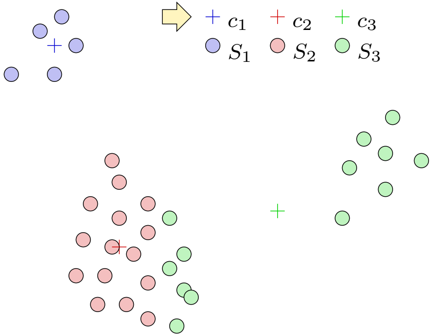


The k-means algorithm: Example

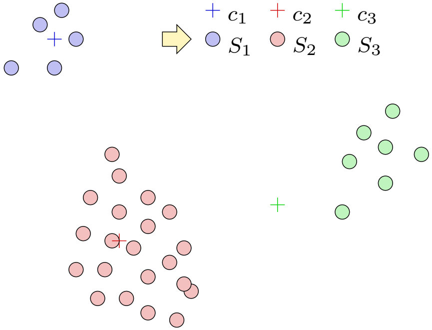
$k = 3$:



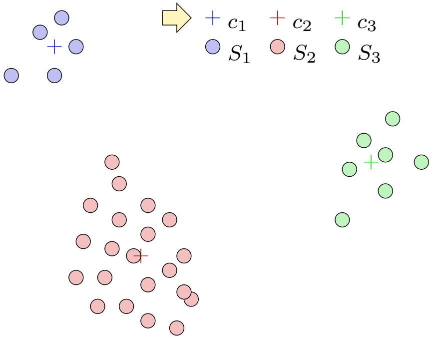
$k = 3$:



$k = 3$:

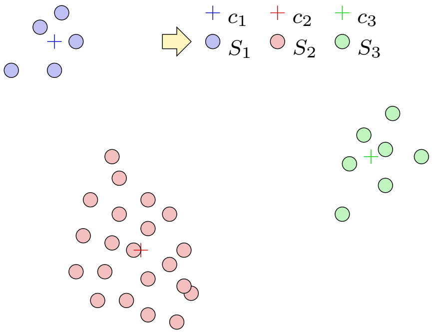


$k = 3$:

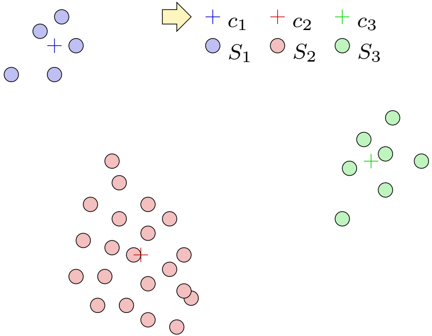


The k-means algorithm: Example

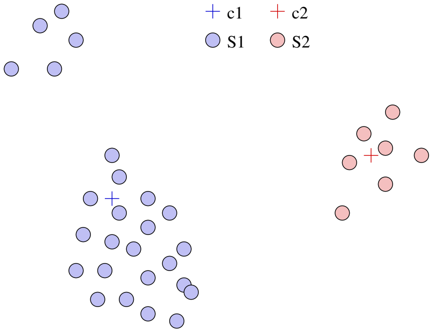
$k = 3$:



$k = 3$:

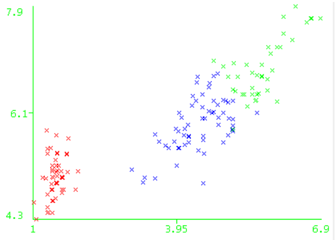


Result for clustering the same data with $k = 2$:

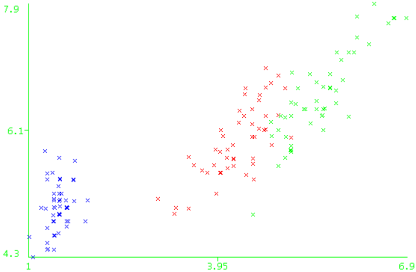


Result can depend on choice of initial cluster centers!

Clustering Iris



Iris 3-means clustered



Iris true classes

The k-means algorithm: Background

k-means as an optimization problem

Assume that we use the Euclidean distance d as proximity measure and that the quality of the clustering is measured by the sum of squared errors:

$$SSE = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{c}_i, \mathbf{x})^2,$$

where:

- $\mathbf{c}_i$ is the i'th centroid
- $C_i \subseteq S$ is the points closets to $\mathbf{c}_i$ according to d.

In principle ...

We can minimize the SSE by looking at all possible partitionings $\rightsquigarrow$ not feasible!

k-means as an optimization problem

Assume that we use the Euclidean distance d as proximity measure and that the quality of the clustering is measured by the sum of squared errors:

$$SSE = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{c}_i, \mathbf{x})^2,$$

where:

- $\mathbf{c}_i$ is the i'th centroid
- $C_i \subseteq S$ is the points closets to $\mathbf{c}_i$ according to d.

In principle . . .

We can minimize the SSE by looking at all possible partitionings $\rightsquigarrow$ not feasible!

Instead, k-means

The centroid that minimizes the SSE is the *mean* of the data-points in that cluster:

$$\mathbf{c}_i = \frac{1}{[C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

Local optimum found by alternating between cluster assignments and centroid estimation.

Convergence

The k-means algorithm is guaranteed to converge

- Each step reduces the sum of squared errors.
- There is only a finite number of cluster assignments.

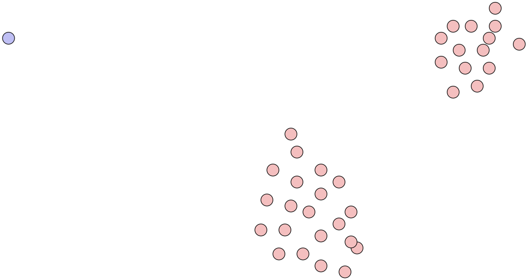
There is no guarantee of reaching the global optimum:

- Improve by running with multiple random restarts.

Some practical issues

Outliers

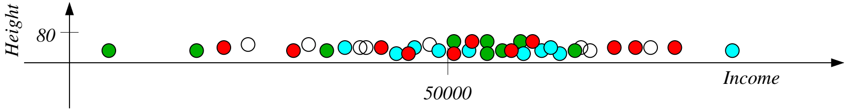
The result of partitional clustering can be skewed by outliers. Example with $k = 2$:



$\rightsquigarrow$ useful preprocessing: outlier detection and elimination.

Instances defined by attributes

$A_1 =$ *height in inches* and $A_2 =$ *annual income in \$* :



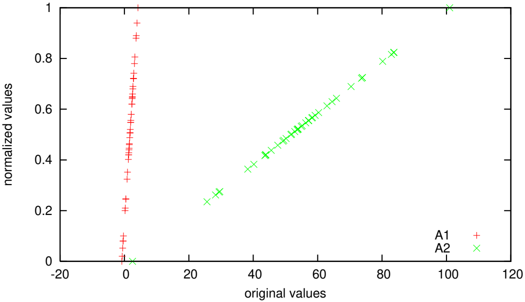
- all distance functions for continuous attributes dominated by *income* values
- $\rightsquigarrow$ may need to *rescale* or *normalize* continuous attributes

Min-Max Normalization

replace A_i with

$$\frac{A_i - \min(A_i)}{\max(A_i) - \min(A_i)}$$

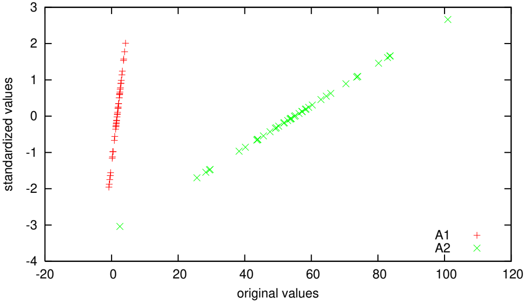
$(\min(A_i), \max(A_i))$ are min/max values of A_i appearing in the data)



Z-score Standardization

replace A_i with

$$\frac{A_i - \mathit{mean}(A_i)}{\mathit{standard\ deviation}(A_i)}$$



where

$$\mathit{mean}(A_i) = \frac{1}{n}\sum_{j=1}^{n} a_{j,i}$$

$$\mathit{standard\ deviation}(A_i) = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(a_{j,i} - \mathit{mean}(A_i))^2}$$

Soft clustering

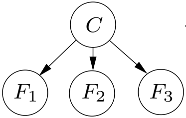
Soft clustering

The k-means algorithm generates a *hard* clustering: each example is assigned to a single cluster.

Alternatively: In *soft* clustering each example is assigned to a cluster with a certain probability.

The naive Bayes model for clustering

Model



Data

F_1	F_2	F_3	C
t	t	t	?
t	f	t	?
t	f	f	?
f	f	t	?
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- C is the hidden cluster variable.
- F_1, F_2, and F_3 are the feature variables.

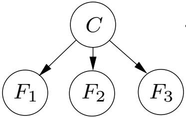
Soft clustering

The k-means algorithm generates a *hard* clustering: each example is assigned to a single cluster.

Alternatively: In *soft* clustering each example is assigned to a cluster with a certain probability.

The naive Bayes model for clustering

Model



Data

F_1	F_2	F_3	C
t	t	t	?
t	f	t	?
t	f	f	?
f	f	t	?
⋮	⋮	⋮	⋮

- C is the hidden cluster variable.
- F_1, F_2, and F_3 are the feature variables.

Procedure

- Set the number of clusters, i.e., the states of C
- Learn the probabilities of the model:
  - $P(C)$, $P(F_1|C)$, $P(F_2|C)$, and $P(F_3|C)$
- Use the learned probabilities to cluster the (future) instances.

When learning the probability distributions of the model, the variable C is hidden

- $\rightsquigarrow$ we *cannot* directly estimate the probabilities using frequency counts

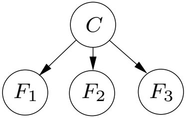
Instead we employ the *Expectation-maximization algorithm*

The EM-algorithm

The main idea:

- Use hypothetical completions of the data using the current probability estimates.
- Infer the maximum likelihood probabilities for the model based on completed data set.

Expectation



Probability tables:

$P_0(C) = (0.6, 0.4)$

$P_0(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Data:

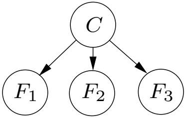
F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C|F_1, F_2, F_3)$
t	t	t	
t	f	t	
t	f	f	
f	f	t	

Maximization

EM for soft clustering: an example



Expectation

Probability tables:

$P_0(C) = (0.6, 0.4)$

$P_0(F_1 \mid C)$	$C = 1$	$C = 2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2 \mid C)$ and $P_0(F_3 \mid C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C \mid F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	
t	f	f	
f	f	t	

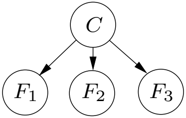
Maximization

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Expectation

- Fractional counts are being calculated by probability updating.



Expectation

Probability tables:

$P_0(C) = (0.6, 0.4)$

$P_0(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	
f	f	t	

Data:

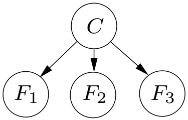
F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Maximization

Expectation

- Fractional counts are being calculated by probability updating.

Expectation



Probability tables:

$P_0(C) = (0.6, 0.4)$

$P_0(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	

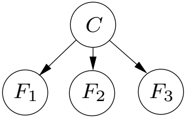
Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Maximization

Expectation

- Fractional counts are being calculated by probability updating.



Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Expectation

Probability tables:

$P_0(C) = (0.6, 0.4)$

$P_0(F_1|C)$

	$C = 1$	$C = 2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

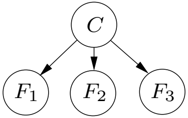
F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	$(0.5, 0.5)$

Maximization

Expectation

- Fractional counts are being calculated by probability updating.

EM for soft clustering: an example



Expectation

Probability tables:

$P_0(C) = (0.6, 0.4)$

$P_0(F_1 \mid C)$	$C=1$	$C=2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2 \mid C)$ and $P_0(F_3 \mid C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C \mid F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	$(0.5, 0.5)$

Maximization

Data:

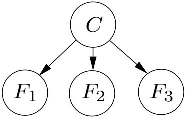
F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Maximization

$$P_1(C) = \frac{1}{4} \sum_{F_1, F_2, F_3} A(F_1, F_2, F_3, C) = \frac{1}{4}(0.84 + 0.69 + 0.5 + 0.5, 0.16 + 0.31 + 0.5 + 0.5)$$
$$= (0.63, 0.37)$$

EM for soft clustering: an example

Expectation



Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_0(F_1|C)$

	$C = 1$	$C = 2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	$(0.5, 0.5)$

Data:

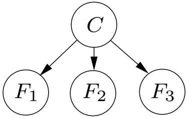
F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Maximization

Maximization

$$P_1(C) = \frac{1}{4} \sum_{F_1, F_2, F_3} A(F_1, F_2, F_3, C) = \frac{1}{4}(0.84 + 0.69 + 0.5 + 0.5, 0.16 + 0.31 + 0.5 + 0.5)$$
$$= (0.63, 0.37)$$

EM for soft clustering: an example



Expectation

Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_0(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.6	0.4
$F_1 = f$	0.4	0.6

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	$(0.5, 0.5)$

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Maximization

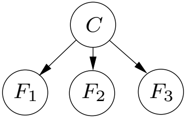
Maximization

$$P_1(F_1|C) = \frac{\sum_{F_2,F_3} A(F_1, F_2, F_3, C)}{\sum_{F_1,F_2,F_3} A(F_1, F_2, F_3, C)} = \frac{\begin{pmatrix} 0.84 + 0.69 + 0.5 + 0 & 0.16 + 0.31 + 0.5 + 0 \\ 0 + 0 + 0 + 0.5 & 0 + 0 + 0 + 0.5 \end{pmatrix}}{(2.53, 1.47)}$$

$$= \begin{pmatrix} 0.8 & 0.65 \\ 0.2 & 0.35 \end{pmatrix}$$

EM for soft clustering: an example

Expectation



Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$:

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	$(0.5, 0.5)$

Data:

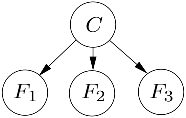
F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Maximization

Maximization

$$P_1(F_1|C) = \frac{\sum_{F_2,F_3} A(F_1, F_2, F_3, C)}{\sum_{F_1,F_2,F_3} A(F_1, F_2, F_3, C)} = \frac{\begin{pmatrix} 0.84+0.69+0.5+0 & 0.16+0.31+0.5+0 \\ 0+0+0+0.5 & 0+0+0+0.5) \end{pmatrix}}{(2.53, 1.47)}$$

$$= \begin{pmatrix} 0.8 & 0.65 \\ 0.2 & 0.35 \end{pmatrix}$$



Expectation

Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C = 1$	$C = 2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_0(F_2|C)$ and $P_0(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$:

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	$(0.5, 0.5)$

Maximization

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

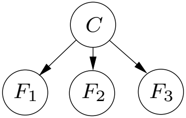
Maximization

$$P_1(F_2|C) = \ldots\ldots = \begin{pmatrix} 0.33 & 0.11 \\ 0.67 & 0.89 \end{pmatrix}$$

$$P_1(F_3|C) = \ldots\ldots = \begin{pmatrix} 0.80 & 066 \\ 0.20 & 0.34 \end{pmatrix}$$

EM for soft clustering: an example

Expectation



Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.84, 0.16)$
t	f	t	$(0.69, 0.31)$
t	f	f	$(0.5, 0.5)$
f	f	t	$(0.5, 0.5)$

Maximization

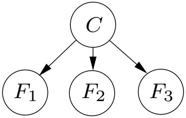
Maximization

$$P_1(F_2|C) = \ldots\ldots = \begin{pmatrix} 0.33 & 0.11 \\ 0.67 & 0.89 \end{pmatrix}$$

$$P_1(F_3|C) = \ldots\ldots = \begin{pmatrix} 0.80 & 066 \\ 0.20 & 0.34 \end{pmatrix}$$

EM for soft clustering: an example

Expectation



Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

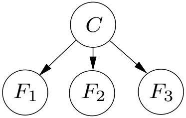
Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	
t	f	f	
f	f	t	

Maximization

Expectation

- Fractional counts are being calculated by probability updating.



Expectation

Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C = 1$	$C = 2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	
f	f	t	

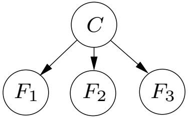
Maximization

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Expectation

- Fractional counts are being calculated by probability updating.



Expectation

Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	$(0.48, 0.52)$
f	f	t	

Maximization

Data:

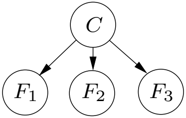
F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Expectation

- Fractional counts are being calculated by probability updating.

EM for soft clustering: an example

Expectation



Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$:

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	$(0.48, 0.52)$
f	f	t	$(0.47, 0.53)$

Maximization

Data:

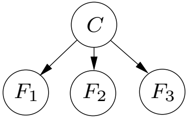
F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Expectation

- Fractional counts are being calculated by probability updating.

EM for soft clustering: an example

Expectation



Probability tables:

$P_1(C) = (0.63, 0.37)$

$P_1(F_1|C)$

	$C=1$	$C=2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

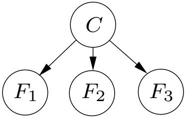
Count table $A(F_1, F_2, F_3, C)$:

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	$(0.48, 0.52)$
f	f	t	$(0.47, 0.53)$

Maximization

Maximization

$$P_2(C) = \frac{1}{4} \sum_{F_1, F_2, F_3} A(F_1, F_2, F_3, C) = \frac{1}{4}(0.88 + 0.66 + 0.48 + 0.47, 0.12 + 0.34 + 0.52 + 0.53)$$
$$= (0.62, 0.38)$$



Expectation

Probability tables:

$P_2(C) = (0.62, 0.38)$

$P_1(F_1|C)$

	$C = 1$	$C = 2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$:

F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	$(0.48, 0.52)$
f	f	t	$(0.47, 0.53)$

Maximization

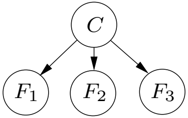
Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Maximization

$$P_2(C) = \frac{1}{4} \sum_{F_1, F_2, F_3} A(F_1, F_2, F_3, C) = \frac{1}{4}(0.88 + 0.66 + 0.48 + 0.47, 0.12 + 0.34 + 0.52 + 0.53)$$
$$= (0.62, 0.38)$$

EM for soft clustering: an example



Expectation

Probability tables:

$P_2(C) = (0.62, 0.38)$

$P_1(F_1 \mid C)$	$C=1$	$C=2$
$F_1 = t$	0.8	0.65
$F_1 = f$	0.2	0.35

Also $P_1(F_2 \mid C)$ and $P_1(F_3 \mid C)$

Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Count table $A(F_1, F_2, F_3, C)$**:**

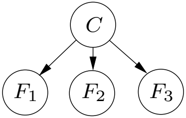
F_1	F_2	F_3	$P(C \mid F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	$(0.48, 0.52)$
f	f	t	$(0.47, 0.53)$

Maximization

Maximization

$$P_2(F_1 \mid C) = \frac{\sum_{F_2,F_3} A(F_1,F_2,F_3,C)}{\sum_{F_1,F_2,F_3} A(F_1,F_2,F_3,C)} = \frac{\begin{pmatrix} 0.88+0.66+0.48+0 & 0.12+0.34+0.52+0 \\ 0+0+0+0.47 & 0+0+0+0.53 \end{pmatrix}}{(2.49, 1.51)}$$

$$= \begin{pmatrix} 0.81 & 0.65 \\ 0.19 & 0.35 \end{pmatrix}$$



Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Expectation

Probability tables:

$P_2(C) = (0.62, 0.38)$

$P_2(F_1|C)$

	$C = 1$	$C = 2$
$F_1 = t$	0.81	0.65
$F_1 = f$	0.19	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$:

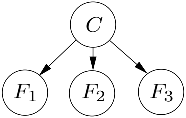
F_1	F_2	F_3	$P(C\|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	$(0.48, 0.52)$
f	f	t	$(0.47, 0.53)$

Maximization

Maximization

$$P_2(F_1|C) = \frac{\sum_{F_2,F_3} A(F_1, F_2, F_3, C)}{\sum_{F_1,F_2,F_3} A(F_1, F_2, F_3, C)} = \frac{\begin{pmatrix} 0.88 + 0.66 + 0.48 + 0 & 0.12 + 0.34 + 0.52 + 0 \\ 0 + 0 + 0 + 0.47 & 0 + 0 + 0 + 0.53) \end{pmatrix}}{(2.49, 1.51)}$$

$$= \begin{pmatrix} 0.81 & 0.65 \\ 0.19 & 0.35 \end{pmatrix}$$



Data:

F_1	F_2	F_3
t	t	t
t	f	t
t	f	f
f	f	t

Expectation

Probability tables:

$P_2(C) = (0.62, 0.38)$

$P_2(F_1|C)$

	$C = 1$	$C = 2$
$F_1 = t$	0.81	0.65
$F_1 = f$	0.19	0.35

Also $P_1(F_2|C)$ and $P_1(F_3|C)$

Count table $A(F_1, F_2, F_3, C)$**:**

F_1	F_2	F_3	$P(C|F_1, F_2, F_3)$
t	t	t	$(0.88, 0.12)$
t	f	t	$(0.66, 0.34)$
t	f	f	$(0.48, 0.52)$
f	f	t	$(0.47, 0.53)$

Maximization

Maximization

...and so we continue until a termination criterion is reached.

Correctness

- The sequence of probability estimates generated by the EM algorithm converges to a local maximum (in rare cases: a saddle point) of the marginal likelihood given the data.
- To avoid sub-optimal local maxima: run EM several times with different starting points.

Correctness

- The sequence of probability estimates generated by the EM algorithm converges to a local maximum (in rare cases: a saddle point) of the marginal likelihood given the data.
- To avoid sub-optimal local maxima: run EM several times with different starting points.

Notes

- Any permutation of the cluster labels of a local maximum will also be a local maximum.
- Rather than keeping track of a full count table, it suffices to store counts for the variable families, *fa*$(X) = \{X\} \cup$ *pa*(X). Only one pass through the data is necessary.
- Clustering an existing or new instance $\mathbf{x}$ amounts to calculating $P(C|\mathbf{x})$.

Cluster evaluation

Cluster evaluation

A clustering algorithm applied to a dataset will return a clustering - even if there is no meaningful structure in the data!

Question: Do the clusters actually correspond to meaningful groups of data instances?
Question: Are all the clusters relevant, or are there some real and some meaningless clusters?

Supervised vs. Unsupervised Evaluation

Unsupervised

- Uses only the data as given to the clustering algorithm, and the resulting clustering
- The realistic scenario
- Can be guided by considering changes in evaluation score.

Supervised

- Uses external information, e.g. a true class label as the “gold standard” for actual groups in the data
- Not representative for actual clustering applications
- Can be useful to evaluate *clustering algorithms*
- Caveat: no guarantee that the class labels actually describe the most natural or relevant groups in the data