CLUSTERING

Based on

"Foundations of Statistical NLP", C. Manning & H. Schütze, MIT Press, 2002, ch. 14

and "Machine Learning", T. Mitchell, McGRAW Hill, 1997, ch. 6.12

Plan

1. Introduction to clustering
   - ◦ Clustering vs Classification
   - ◦ Hierarchical vs non-hierarchical clustering
   - ◦ Soft vs hard assignments in clustering
2. Hierarchical clustering
   - • Bottom-up (agglomerative) clustering
   - • Top-down (divisive) clustering
   - ◦ Similarity functions in clustering:
     single link, complete link, group average
3. Non-hierarchical clustering
   - • the k-means clustering algorithm
   - • the EM algorithm for Gaussian Mixture Modelling
     (estimating the means of k Gaussians)

1 Introduction to clustering

Clustering vs Classification

Classification = supervised learning,

i.e. we need a set of labeled training instances for each group/class.

Clustering = unsupervised learning,

because there is no teacher who provides the examples in the training set with class labels.

It assumes no pre-existing categorization scheme;

the clusters are induced from data.

- **Clustering:**
  partition a set of objects into groups/clusters.
- **The goal: place objects which are similar (according to a certain similarity measure) in a same group, and assign dissimilar objects to different groups.**
- **Objects are usually described and clustered using a set of features and values (often known as *the data representation model*).**

Hierarchical vs Non-hierarchical Clustering

Hierarchical Clustering

produces a tree of groups/clusters, each node being a subgroup of its mother.

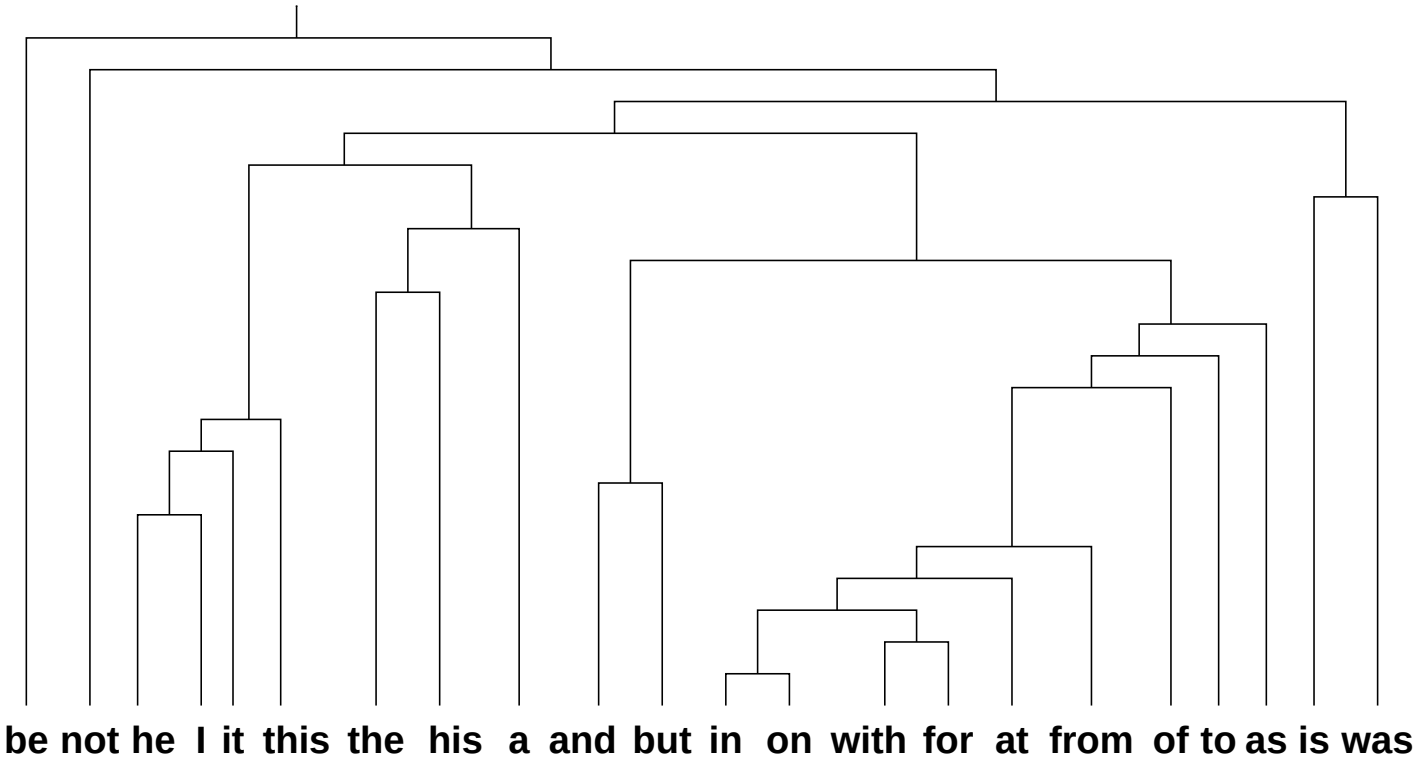
Non-hierarchical Clustering (or, flat clustering):

the relation between clusters is often left undetermined.

Most non-hierarchical clustering algorithms are iterative. They start with a set of initial clusters and then iteratively improve them using a reallocation scheme.

An Example of Hierarchical Clustering:

A Dendrogram showing a clustering of 22 high frequency words from the Brown corpus



The Dendrogram Commented

- Similarity in this case is based on the left and right context of words. (Firth: "one can characterize a word by the words that occur around it".)
  - For instance:
    he, I, it, this have more in common with each other than they have with *and, but*;
    in, on have a greater similarity than *he, I*.
- Each node in the tree represents a cluster that was created by merging two child nodes.
- The height of a connection corresponds to the apparent (di)similarity between the nodes at the bottom of the diagram.

Exemplifying the Main Uses of Clustering (I)

Generalisation

We want to figure out the correct preposition to use with the noun *Friday* when translating a text from French into English.

The days of the week get put in the same cluster by a clustering algorithm which measures similarity of words based on their contexts.

Under the assumption that an environment that is correct for one member of the cluster is also correct for the other members,
we can infer the correctness of *on Friday* from the presence (in the given corpus) of *on Sunday, on Monday.*

Main Uses of Clustering (II)
Exploratory Data Analysis (EDA)

Any technique that lets one to better visualise the data is likely to

- **bring to the fore new generalisations, and**
- **stop one from making wrong assumptions about data.**

This is a 'must' for domains like Statistical Natural Language Processing and Biological Sequence Analysis.

2 Hierarchical Clustering

Botom-up (Agglomerative) Clustering:

Form all possible singleton clusters (each containing a single object).

Greedily combine clusters with "maximum similarity" (or "minimum distance") together into a new cluster.

Continue until all objects are contained in a single cluster.

Top-down (Divisive) Clustering:

Start with a cluster containing all objects.

Greedily split the cluster into two, assigning objects to clusters so as to maximize the within-group similarity.

Continue splitting clusters which are the least coherent until either having only singleton clusters or reaching the number of desired clusters.

The Bottom-up Hierarchical Clustering Algorithm

Given: **a set** $X = \{x_1, \ldots, x_n\}$ **of objects**
a function sim: $\mathcal{P}(X) \times \mathcal{P}(X) \to R$
for $i = 1, n$ **do**
$c_i = \{x_i\}$ **end**
$C = \{c_1, \ldots, c_n\}$
$j = n + 1$
while $\mid C \mid > 1$
$(c_{n_1}, c_{n_2}) = \text{argmax}_{(c_u, c_v) \in C \times C}$ **sim**(c_u, c_v)
$c_j = c_{n_1} \cup c_{n_2}$
$C = C \backslash \{c_{n_1}, c_{n_2}\} \cup \{c_j\}$
$j = j + 1$

Bottom-up Hierarchical Clustering: Further Comments

- **In general, if d is a distance measure, then one can take**

$$\mathbf{sim}(x, y) = \frac{1}{1 + d(x, y)}$$

- **Monotonicity of the similarity function:**
  The operation of merging must not increase the similarity:

$$\forall c, c', c'' : \mathbf{min}(\mathbf{sim}(c, c'), \mathbf{sim}(c, c'')) \geq \mathbf{sim}(c, c' \cup c'').$$

The Top-down Hierarchical Clustering Algorithm

Given: **a set** $X = \{x_1, \ldots, x_n\}$ **of objects**
a function coh: $\mathcal{P}(X) \to R$
a function split: $\mathcal{P}(X) \to \mathcal{P}(X) \times \mathcal{P}(X)$

$C = \{X\} (= \{c_1\})$
$j = 1$
while $\exists c_i \in C$ **such that** $\mid c_i \mid > 1$
$c_u = \text{argmin}_{c_v \in C}$ **coh**(c_v)
$c_{j+1} \cup c_{j+2}$ **= split** (c_u)
$C = C \backslash \{c_u\} \cup \{c_{j+1}, c_{j+2}\}$
$j = j + 2$

Top-down Hierarchical Clustering: Further Comments

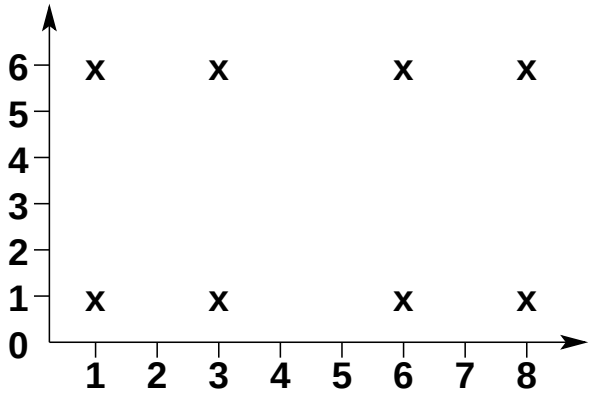
- **Similarity functions (see next slide) can be used here also as coherence.**
- **To split a cluster in two sub-clusters:**
  any bottom-up or non-hierarchical clustering algorithms can be used;
  better use the relative entropy (the Kulback-Leibler (KL) divergence):

$$D(p \,||\, q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

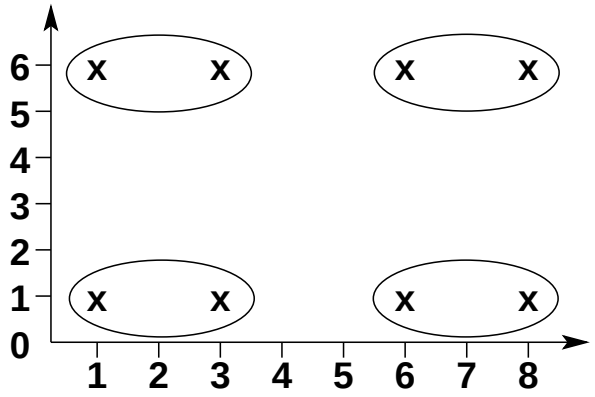
  where it is assumed that $0 \log \frac{0}{q} = 0$**, and** $p \log \frac{p}{0} = \infty$**.**

Classes of Similarity Functions

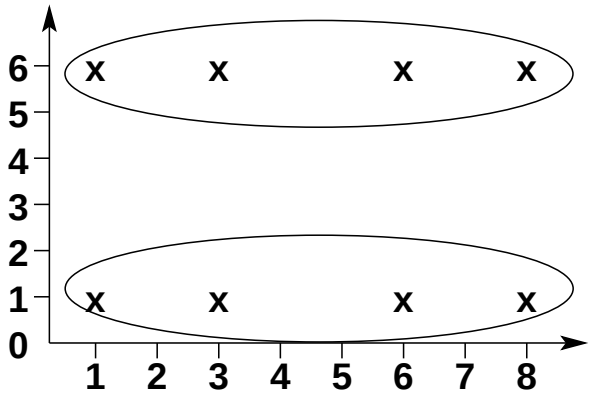
- **single link:** similarity of two clusters considered for merging is determined by the two most similar members of the two clusters
- **complete link:** similarity of two clusters is determined by the two least similar members of the two clusters
- **group average:** similarity is determined by the average similarity between all members of the clusters considered.



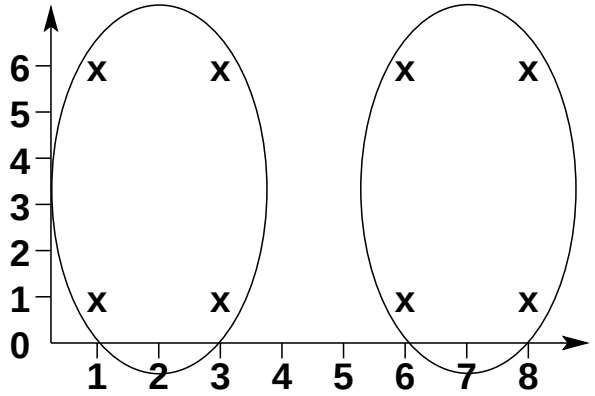
a set of points in a plane



first step in single/complete clustering



single−link clustering



complete−link clustering

Single-link vs Complete-link Clustering: Pros and Cons

Single-link Clustering:

- good local coherence, since the similarity function is locally defined
- can produce elongated clusters ("the chaining effect")
- Closely related to the Minimum Spanning Tree (MST) of a set of points.
  (Of all trees connecting the set of objects, the sum of the edges of the MST is minimal.)
- In graph theory, it corresponds to finding a maximally connected graph. Complexity: $O(n^2)$.

Complete-link Clustering:

- The focuss is on the global cluster quality.
- In graph theory, it corresponds to finding a clique (maximally complete subgraph of) a given graph. Complexity: $O(n^3)$.

Group-average Agglomerative Clustering

The criterion for merges: average similarity, which in some cases can be efficiently computed, implying $O(n^2)$. For example, one can take

$$sim(\overline{x}, \overline{y}) = cos(\overline{x}, \overline{y}) = \frac{\overline{x} \cdot \overline{y}}{\mid \overline{x} \mid\mid \overline{y} \mid} = \sum_{i=1}^{m} x_i y_i$$

with $\overline{x}, \overline{y}$ being length-normalised, i.e., $\mid \overline{x} \mid = \mid \overline{y} \mid = 1$.

Therefore, it is a good compromise between single-link and complete-link clustering.

Group-average Agglomerative Clustering: Computation

Let $\mathcal{X} \subseteq R^m$ be the set of objects to be clustered

The average similarity of a cluster c_j is:

$$S(c_j) = \frac{1}{\mid c_j \mid (\mid c_j \mid -1)} \sum_{\overline{x} \in c_j} \sum_{\overline{y} \neq \overline{x} \in c_j} sim(\overline{x}, \overline{y})$$

Considering $\overline{s}(c_j) = \sum_{\overline{x} \in c_j} \overline{x}$ and assuming $\mid \bar{x} \mid = 1$, then:

$$\overline{s}(c_j) \cdot \overline{s}(c_j) = \sum_{\overline{x} \in c_j} \sum_{\overline{y} \in c_j} \overline{x} \cdot \overline{y} = \mid c_j \mid (\mid c_j \mid -1) S(c_j) + \sum_{\overline{x} \in c_j} \overline{x} \cdot \overline{x} = \mid c_j \mid (\mid c_j \mid -1) S(c_j) + \mid c_j \mid$$

Therefore:

$$S(c_j) = \frac{\overline{s}(c_j) \cdot \overline{s}(c_j) - \mid c_j \mid}{\mid c_j \mid (\mid c_j \mid -1)}$$

and

$$S(c_i \cup c_j) = \frac{(\overline{s}(c_i) + \overline{s}(c_j)) \cdot (\overline{s}(c_i) + \overline{s}(c_j)) - (\mid c_i \mid + \mid c_j \mid)}{(\mid c_i \mid + \mid c_j \mid)(\mid c_i \mid + \mid c_j \mid -1)}$$

and

$$\bar{s}(c_i \cup s_j) = \bar{s}(c_i) + \bar{s}(c_j)$$

which requires constant time for computing.

Application of Hierarchical Clustering:

Improving Language Modeling

[Brown et al., 1992],
[Manning & Schuetze, 1992], pages 509–512

Using cross-entropy ($-\frac{1}{N} log P(w_1, \ldots, w_N)$) and bottom-up clustering, Brown obtained a cluster-based language model which didn't prove better than the word-based model.
But the linear interpolation of the two models was better than both!

Example of 3 clusters obtained by Brown:

- **plan, letter, request, memo, case, question, charge, statement, draft**
- **day, year, week, month, quarter, half**
- **evaluation, assessment, analysis, understanding, opinion, conversation, discussion**

Note that the words in these clusters have similar syntactic and semantic properties.

Soft vs Hard Assignments in Clustering

Hard assignment:
each object is assigned to one and only one cluster.
This is the typical choice for hierarchical clustering.

Soft assignment: allows degrees of membership, and membership in multiple clusters.

In a vector space model,
the **centroid** (or, center of gravity) of each cluster c is

$$\overline{\mu} = \frac{1}{\mid c \mid} \sum_{\overline{x} \in c} \overline{x}$$

and the degree of membership of $\overline{x}$ in multiple clusters can be (for instance) the distance between x and $\overline{\mu}$.

Non-hierarchical clustering works with both hard assignments and soft assignments.

3 Non-hierarchical Clustering

As already mentioned, start with an initial set of seeds (one seed for each cluster), then iteratively refine it.

The initial centers for clusters can be computed by applying a hierarchical clustering algorithm on a subset of the objects to be clustered (especially in the case of ill-behaved sets).

Stopping criteria (examples):

- group-average similarity
- the likelyhood of data, given the clusters
- the Minimum Description Length (MDL) principle
- mutual information between adjiacent clusters
- ...

An Example of Non-hierarchical Clustering:

3.1 The k-Means Algorithm

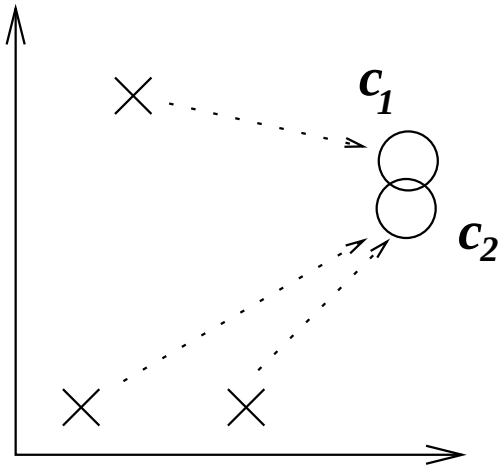
[S. P. Lloyd, 1957]

Given a set $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{R}^m$,
a distance measure d on $\mathcal{R}^m$,
a function for computing the mean $\mu : \mathcal{P}(\mathcal{R}^m) \to \mathcal{R}^m$,
built k clusters so as to satisfy a certain ("stopping") criterion (e.g., maximization of group-average similarity).

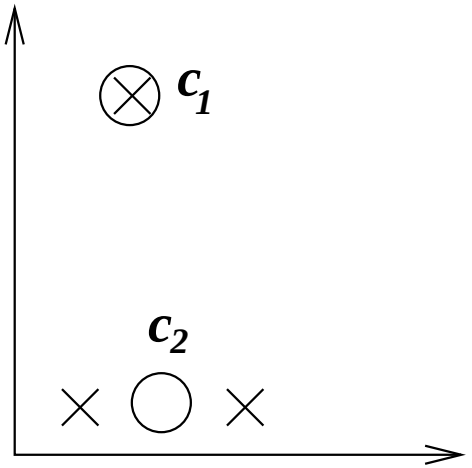
Procedure:

Select (arbitrarily) k initial centers $f_1, \ldots, f_k$ in $\mathcal{R}^m$;
while the stopping criterion is not satisfied
for all clusters c_j do $c_j = \{x_i \mid \forall f_l \; d(x_i, f_j) \leq d(x_i, f_l)\}$ end
for all means f_j do $f_j \leftarrow \mu(c_j)$ end

Illustrating the k-Means Clustering Algorithm



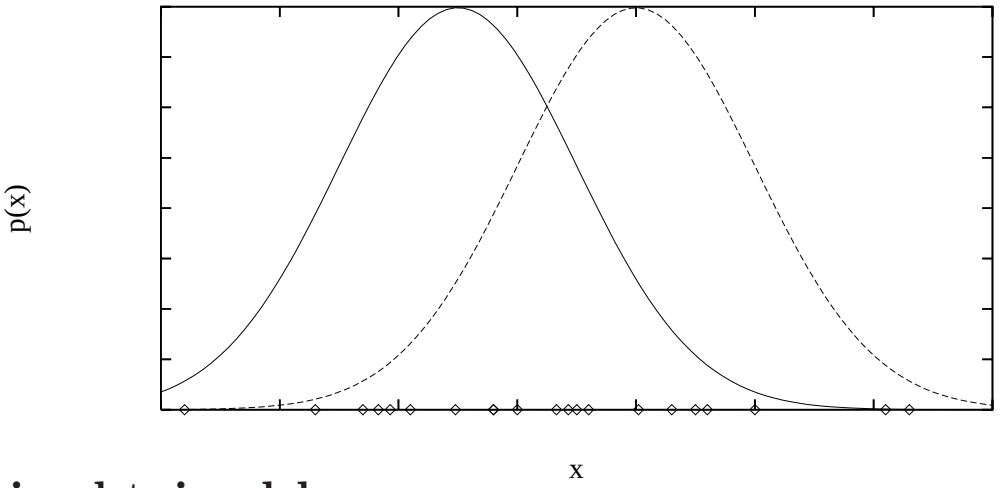
assignment



recomputation of means

3.2 Gaussian Mixture Modeling

3.2.0 Generating Data from a Mixture of k Gaussians



Each instance x is obtained by

1. **Choosing one of the k Gaussians having the same variance σ^2 with – for simplicity – uniform probability;**
2. **Generating randomly an instance according to that Gaussian.**

3.2.1 The Problem

Given

- D, a set of instances from X generated by a mixture of k Gaussian distributions;
- the unknown means $\langle \mu_1, \ldots, \mu_k \rangle$ of the k Gaussians;
  - to simplify the presentation, all Gaussians are assumed to have the same variance σ^2, and they are selected with equal probability;
- we don't know which x_i was generated by which Gaussian;

determine

- h, the ML estimates of $\langle \mu_1, \ldots, \mu_k \rangle$, i.e. $\text{argmax}_h \, P(D \mid h)$.

Notations

For the previously given example $(k = 2)$**,**

we can think of the full description of each instance as

$y_i =< x_i, z_{i1}, z_{i2} >$**, where**

- x_i **is observable,** z_{ij} **is unobservable**
- z_{ij} **is 1 if** x_i **was generated by** j**th Gaussian and 0 otherwise**

Note

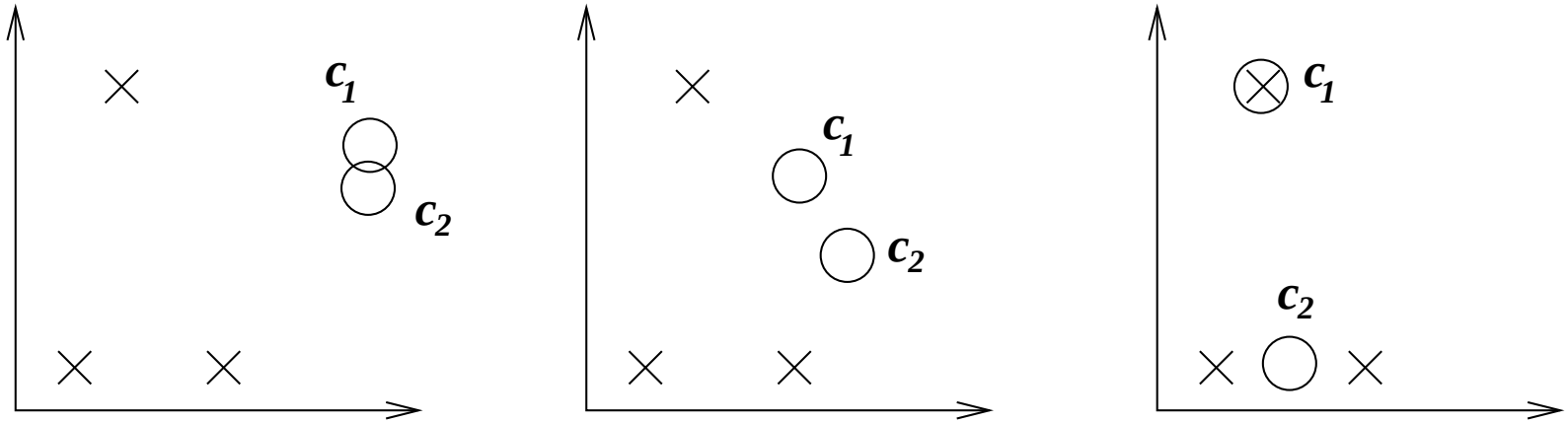
For $k = 1$ there will be no unbservable variables.
We have already shown — see the Bayesian Learning chapter, the ML hypothesis section — that the ML hypothesis is the one that minimizes the sum of squared errors:

$$\mu_{ML} = \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{m} (x_i - \mu)^2 = \frac{1}{m} \sum_{i=1}^{m} x_i$$

Indeed, it is in this way that the k-means algorithm works towards solving the problem of estimating the means of k Gaussians.

REMARK

The k-means algorithm finds a local optimum for a the "sum of squares" criterion. While neither being able to find the global optimum, the following algorithm — which uses soft assignments of instances to clusters, i.e. $z_{ij} \in \{0, 1\}$, and $\sum_{j=1}^{k} P(z_{ij}) = 1$ — may lead to better results, since it uses slower/"softer" changes to the values (and means) of unknown variables.



initial state | after iteration 1 | after iteration 2

3.2.2 The EM Algorithm for Gaussian Mixture Modeling

The Idea

EM finds a local maximum of $E[\ln P(Y|h)]$, where

- **Y is complete set of (observable plus unobservable) variables/data**
- **the expected value of $\ln P(Y|h)$ is taken over possible values of unobserved variables in Y.**

EM for GMM: Algorithm Overview

Initial step: **Pick at random** $h^{(0)} = \langle \mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_k^{(0)} \rangle$**, then – until a certain *condition* is met – iterate:**

Estimation step: **Assuming that the current hypothesis** $h^{(t)} = \langle \mu_1^{(t)}, \mu_2^{(t)}, \ldots, \mu_k^{(t)} \rangle$ **holds, for each hidden variable** Z_{ij} **calculate the expected value** $E[Z_{ij}] \stackrel{not.}{=} E[Z_{ij}|X = x_i; \mu_j^{(t)}]$**:**

$$E[Z_{ij}] = P(Z_{ij} = 1|X = x_i; \mu_j^{(t)}) \stackrel{B.Th.}{=} \frac{p(x = x_i|\mu = \mu_j^{(t)})}{\sum_{l=1}^{k} p(X = x_i|\mu_l^{(t)})} = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j^{(t)})^2}}{\sum_{l=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i - \mu_l^{(t)})^2}}$$

Maximization step: **Assuming that the value of each hidden variable** Z_{ij} **is its own expected value** $E[Z_{ij}]$ **as calculated above, choose a new ML hypothesis** $h^{(t+1)} = \langle \mu_1^{(t+1)}, \mu_2^{(t+1)}, \ldots, \mu_k^{(t+1)} \rangle$ **so as to maximize** $E[\ln P(y_1, \ldots, y_m \mid h)]$ **(see the next slides):**

$$\mu_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^{m} E[Z_{ij}]\ x_i}{\sum_{i=1}^{m} E[Z_{ij}]}$$

Replace $h^{(t)} =< \mu_1^{(t)}, \mu_2^{(t)}, \ldots, \mu_k^{(t)} >$ **by** $< \mu_1^{(t+1)}, \mu_2^{(t+1)}, \ldots, \mu_k^{(t+1)} >$**.**

Calculus for the Expectation Step

$$
\begin{aligned}
E[Z_{ij}] &\overset{def.}{=} 0 \cdot P(Z_{ij} = 0 \mid x_i, h^{(t)}) + 1 \cdot P(Z_{ij} = 1 \mid x_i, h^{(t)}) \\
&= P(Z_{ij} = 1 \mid x_i, h^{(t)}) \\
&\overset{T.Bayes}{=} \frac{P(x_i \mid x_i, Z_{ij} = 1; h^{(t)}) \cdot \overbrace{P(Z_{ij} = 1 \mid x_i, h^{(t)})}^{1/k}}{\sum_{l=1}^{k} (x_i \mid x_i, Z_{il} = 1; h^{(t)}) \cdot \underbrace{P(Z_{il} = 1 \mid x_i, h^{(t)})}_{1/k}} \\
&\overset{p.d.f.}{=} \frac{\mathcal{N}(x = x_i | \mu = \mu_j^{(t)})}{\sum_{l=1}^{k} \mathcal{N}(X = x_i | \mu = \mu_l^{(t)})} = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j^{(t)})^2}}{\sum_{l=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i - \mu_l^{(t)})^2}}
\end{aligned}
$$

***Note*: The a priori probabilities $P(Z_{il} = 1 \mid x_i, h^{(t)})$ have been assumed as being identical, irrespective of l.**

Calculus for the Maximization Step (I)

$$p(y_i|h) \overset{not.}{=} p(x_i, z_{i1}, \ldots, z_{ik}|h) = p(x_i|z_{i1}, \ldots, z_{ik}; h)\underbrace{p(z_{i1}, \ldots, z_{ik}|h)}_{1/k}$$

$$= \frac{1}{k}\frac{1}{\sqrt{2\pi}\sigma}\; e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j)^2}$$

$$\Rightarrow \ln P(Y|h) \overset{i.i.d.}{=} \ln \prod_{i=1}^{m} p(y_i|h) = \sum_{i=1}^{m} \ln p(y_i|h)$$

$$= \sum_{i=1}^{m}(-\ln k + \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i - \mu_j)^2)$$

$$\Rightarrow E[\ln P(Y|h)] \overset{lin.\ med.}{=} \sum_{i=1}^{m}(-\ln k + \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[Z_{ij}](x_i - \mu_j)^2)$$

Calculus for the Maximization Step (II)

$$\begin{aligned}
&\operatorname*{argmax}_h E[\ln P(Y|h)] \\
&= \operatorname*{argmax}_h \sum_{i=1}^{m} (-\ln k + \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} E[Z_{ij}](x_i - \mu_j)^2) \\
&= \operatorname*{argmin}_h \sum_{i=1}^{m} \sum_{j=1}^{k} E[Z_{ij}](x_i - \mu_j)^2 = \operatorname*{argmin}_h \sum_{j=1}^{k} \sum_{i=1}^{m} E[Z_{ij}](x_i - \mu_j)^2 \\
&= \operatorname*{argmin}_h \sum_{j=1}^{k} \{(\sum_{i=1}^{m} E[Z_{ij}])\mu_j^2 - 2(\sum_{i=1}^{m} E[Z_{ij}]x_i)\mu_j + \sum_{i=1}^{m} E[Z_{ij}]x_i^2\}
\end{aligned}$$

$$\Rightarrow \mu_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^{m} E[Z_{ij}]\ x_i}{\sum_{i=1}^{m} E[Z_{ij}]}$$

EM for GMM: Justification

It can be shown (Baum et al. 1970) that after each iteration $P(Y \mid h)$ increases, unless it is a local maximum. Therefore the previously defined EM algorithm

- converges to a (local) maximum likelihood hypothesis h,
- by providing iterative estimates of the hidden variables Z_{ij}.

Hierarchical vs. Non-hierarchical Clustering: Pros and Cons

Hierarchical Clustering:

- ***preferable for detailed data analysis:*** provides more informations than non-hierarchical clustering;
- ***less efficient than non-hierarchical clustering:*** one has to compute at least $n \times n$ similarity coefficients and then update them during the clustering process.

Non-hierarchical Clustering:

- ***preferable if data sets are very large, or efficiency is a key issue***;
- the k-means algo is conceptually the simplest method and should be used first on a new data set (its results are often sufficient);
- k-means (using a simple Euclidian metric), is ***not usable on "nominal" data*** like colours. In such cases, use the EM algorithm.