Machine Learning

C. Unsupervised Learning
C.1 Cluster Analysis

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Syllabus

Fri. 26.10.	(1)	0. Introduction
		A. Supervised Learning: Linear Models & Fundamentals
Fri. 2.11.	(2)	A.1 Linear Regression
Fri. 9.11.	(3)	A.2 Linear Classification
Fri. 16.11.	(4)	A.3 Regularization
Fri. 23.11.	(5)	A.4 High-dimensional Data
		B. Supervised Learning: Nonlinear Models
Fri. 30.11.	(6)	B.1 Nearest-Neighbor Models
Fri. 7.12.	(7)	B.2 Neural Networks
Fri. 14.12.	(8)	B.3 Decision Trees
Fri. 21.12.	(9)	B.4 Support Vector Machines
		— *Christmas Break* —
Fri. 11.1.	(10)	B.5 A First Look at Bayesian and Markov Networks
		C. Unsupervised Learning
Fri. 18.1.	(11)	C.1 Clustering
Fri. 25.1.	(12)	C.2 Dimensionality Reduction
Fri. 1.2.	(13)	C.3 Frequent Pattern Mining
Fri. 8.2.	(14)	Q&A

Outline

1. k-means & k-medoids

2. Gaussian Mixture Models

3. Hierarchical Cluster Analysis

Outline

Partitions

Let X be a set. A set $P \subseteq \mathcal{P}(X)$ of subsets of X is called a **partition of X** if the subsets

1. are pairwise disjoint: $A \cap B = \emptyset, \quad A, B \in P, A \neq B$
2. cover X: $\bigcup_{A \in P} A = X,$ and
3. do not contain the empty set: $\emptyset \notin P.$

Partitions

Let $X := \{x_1, \ldots, x_N\}$ be a finite set. A set $P := \{X_1, \ldots, X_K\}$ of subsets $X_k \subseteq X$ is called a **partition of X** if the subsets

1. are **pairwise disjoint**: $X_k \cap X_j = \emptyset, \quad k, j \in \{1, \ldots, K\}, k \neq j$
2. **cover X**: $\bigcup_{k=1}^{K} X_k = X$, and
3. do **not contain the empty set**: $X_k \neq \emptyset, \quad k \in \{1, \ldots, K\}$.

A set X_k is also called a **cluster**, a partition P a **clustering**.
$K \in \mathbb{N}$ is called **number of clusters**.

$\text{Part}(X)$ denotes the set of all partitions of X.

Partitions

Let X be a finite set. A **surjective** function

$$p : X \to \{1, \ldots, K\}$$

is called a **K partition function of X**.

The sets $X_k := p^{-1}(k)$ form a partition $P := \{X_1, \ldots, X_K\}$.

Partitions

Let $X := \{x_1, \ldots, x_N\}$ be a finite set. A binary $N \times K$ matrix

$$P \in \{0,1\}^{N \times K}$$

is called a **partition matrix of X** if it

1. is **row-stochastic**: $$\sum_{k=1}^{K} P_{n,k} = 1, \quad n \in \{1, \ldots, N\}$$
2. does **not contain a zero column**: $$X_{.,k} \neq (0, \ldots, 0)^T, \quad k \in \{1, \ldots, K\}$$

The sets $X_k := \{x_n \mid n \in \{1, \ldots, N\}, P_{n,k} = 1\}$ form a partition $P := \{X_1, \ldots, X_K\}$.

$P_{.,k}$ is called **membership vector of class k**.

The Cluster Analysis Problem

Given

- a set $\mathcal{X}$ called **data space**, e.g., $\mathcal{X} := \mathbb{R}^M$,
- a set $X \subseteq \mathcal{X}$ called **data**, and
- a function

$$D : \bigcup_{X \subseteq \mathcal{X}} \text{Part}(X) \to \mathbb{R}_0^+$$

called **distortion measure** where $D(P)$ measures how bad a partition $P \in \text{Part}(X)$ for a data set $X \subseteq \mathcal{X}$ is,

find a partition $P = \{X_1, X_2, \ldots X_K\} \in \text{Part}(X)$ with minimal distortion $D(P)$.

The Cluster Analysis Problem (with K clusters)

Given

- a set $\mathcal{X}$ called **data space**, e.g., $\mathcal{X} := \mathbb{R}^M$,
- a set $X \subseteq \mathcal{X}$ called **data**,
- a function

$$D : \bigcup_{X \subseteq \mathcal{X}} \text{Part}(X) \rightarrow \mathbb{R}_0^+$$

  called **distortion measure** where $D(P)$ measures how bad a partition $P \in \text{Part}(X)$ for a data set $X \subseteq \mathcal{X}$ is, and

- a number $K \in \mathbb{N}$ of clusters,

find a partition $P = \{X_1, X_2, \ldots X_K\} \in \text{Part}_K(X)$ with K clusters with minimal distortion $D(P)$.

k-means: Distortion Sum of Distances to Cluster Centers

Sum of squared distances to cluster centers:

$$D(P) := \sum_{k=1}^{K} \sum_{\substack{n=1: \\ P_{n,k}=1}}^{N} ||x_n - \mu_k||^2$$

with

$$\mu_k := \text{mean} \, \{x_n \mid P_{n,k} = 1, n \in \{1, \ldots, N\}\}$$

k-means: Distortion Sum of Distances to Cluster Centers

Sum of squared distances to cluster centers:

$$D(P) := \sum_{n=1}^{N} \sum_{k=1}^{K} P_{n,k} ||x_n - \mu_k||^2 = \sum_{k=1}^{K} \sum_{\substack{n=1: \\ P_{n,k}=1}}^{N} ||x_n - \mu_k||^2$$

with

$$\mu_k := \frac{\sum_{n=1}^{N} P_{n,k} x_n}{\sum_{n=1}^{N} P_{n,k}} = \text{mean } \{x_n \mid P_{n,k} = 1, n \in \{1, \ldots, N\}\}$$

k-means: Distortion Sum of Distances to Cluster Centers

Sum of squared distances to cluster centers:

$$D(P) := \sum_{n=1}^{N} \sum_{k=1}^{K} P_{n,k} ||x_n - \mu_k||^2 = \sum_{k=1}^{K} \sum_{\substack{n=1: \\ P_{n,k}=1}}^{N} ||x_n - \mu_k||^2$$

with

$$\mu_k := \text{ mean } \{x_n \mid P_{n,k} = 1, n \in \{1, \ldots, N\}\}$$

Minimizing D over partitions with varying number of clusters leads to singleton clustering with distortion 0; only the cluster analysis problem with given K makes sense.

Minimizing D is not easy as reassigning a point to a different cluster also shifts the cluster centers.

k-means: Minimizing Distances to Cluster Centers

Add cluster centers μ as auxiliary optimization variables:

$$D(P, \mu) := \sum_{n=1}^{N} \sum_{k=1}^{K} P_{n,k} ||x_n - \mu_k||^2$$

k-means: Minimizing Distances to Cluster Centers

Add cluster centers μ as auxiliary optimization variables:

$$D(P, \mu) := \sum_{n=1}^{N} \sum_{k=1}^{K} P_{n,k} ||x_n - \mu_k||^2$$

Block coordinate descent:

1. fix μ, optimize P $\rightsquigarrow$ reassign data points to clusters:

$$P_{n,k} := \mathbb{I}(k = \ell_n), \quad \ell_n := \underset{k \in \{1,\ldots,K\}}{\arg\min} ||x_n - \mu_k||^2$$

k-means: Minimizing Distances to Cluster Centers

Add cluster centers μ as auxiliary optimization variables:

$$D(P, \mu) := \sum_{n=1}^{N} \sum_{k=1}^{K} P_{n,k} ||x_n - \mu_k||^2$$

Block coordinate descent:

1. fix μ, optimize $P \rightsquigarrow$ reassign data points to clusters:

$$P_{n,k} := \mathbb{I}(k = \ell_n), \quad \ell_n := \arg\min_{k \in \{1,\ldots,K\}} ||x_n - \mu_k||^2$$

2. fix P, optimize $\mu \rightsquigarrow$ recompute cluster centers:

$$\mu_k := \frac{\sum_{n=1}^{N} P_{n,k} x_n}{\sum_{n=1}^{N} P_{n,k}}$$

Iterate until partition is stable.

k-means: Initialization

k-means is usually initialized by picking K data points as cluster centers at random:

1. pick the first cluster center μ_1 out of the data points at random and then
2. sequentially select the data point with the largest sum of distances to already choosen cluster centers as next cluster center

$$\mu_k := x_n, \quad n := \underset{n \in \{1,\ldots,N\}}{\arg\max} \sum_{\ell=1}^{k-1} ||x_n - \mu_\ell||^2, \quad k = 2, \ldots, K$$

k-means: Initialization

k-means is usually initialized by picking K data points as cluster centers at random:

1. pick the first cluster center μ_1 out of the data points at random and then
2. sequentially select the data point with the largest sum of distances to already choosen cluster centers as next cluster center

$$\mu_k := x_n, \quad n := \underset{n \in \{1,\ldots,N\}}{\arg\max} \sum_{\ell=1}^{k-1} ||x_n - \mu_\ell||^2, \quad k = 2, \ldots, K$$

Different initializations may lead to different local minima.

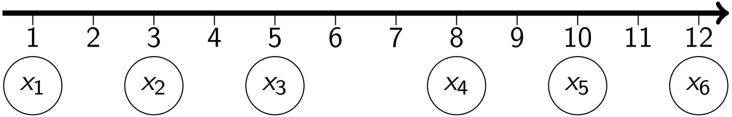
- run k-means with different random initializations and
- keep only the one with the smallest distortion (**random restarts**).

k-means Algorithm

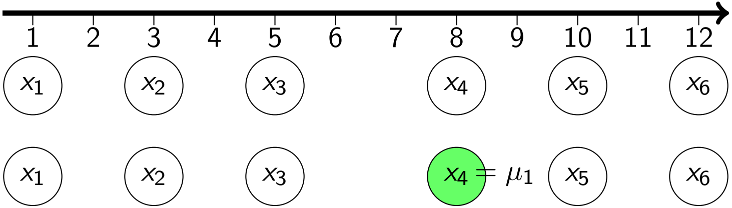
1 **cluster-kmeans**$(\mathcal{D} := \{x_1, \ldots, x_N\} \subseteq \mathbb{R}^M, K \in \mathbb{N}, \epsilon \in \mathbb{R}^+)$:
2 $\quad n_1 \sim \text{unif}(\{1, \ldots, N\}), \quad \mu_1 := x_{n_1}$
3 $\quad$ for $k := 2, \ldots, K$:
4 $\quad\quad n_k := \arg\max_{n \in \{1,\ldots,N\}} \sum_{j=1}^{k-1} ||x_n - \mu_j||, \quad \mu_k := x_{n_k}$
5 $\quad$ repeat
6 $\quad\quad \mu^{\text{old}} := \mu$
7 $\quad\quad$ for $n := 1, \ldots, N$:
8 $\quad\quad\quad P_n := \arg\min_{k \in \{1,\ldots,K\}} ||x_n - \mu_k||$
9 $\quad\quad$ for $k := 1, \ldots, K$:
10 $\quad\quad\quad \mu_k := \text{mean } \{x_n \mid P_n = k, n \in \{1, \ldots, N\}\}$
11 $\quad$ until $\frac{1}{K} \sum_{k=1}^{K} ||\mu_k - \mu_k^{\text{old}}|| < \epsilon$
12 $\quad$ return P

Note: In implementations, the two loops over the data (lines 7 and 10) can be combined in one loop.

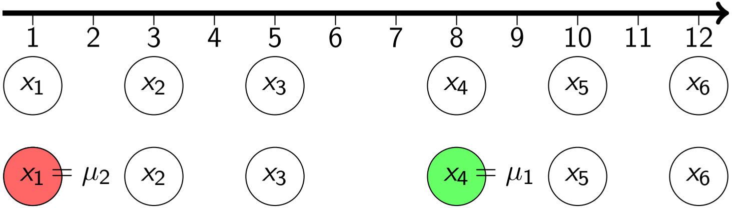
Example



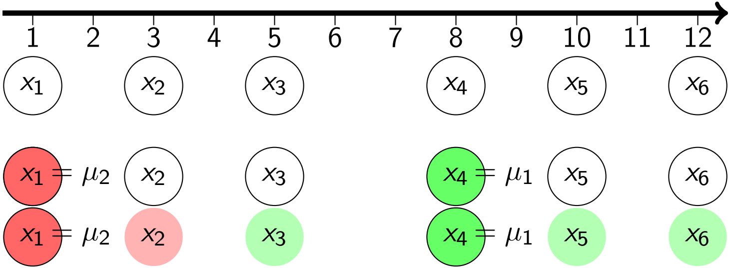
Example



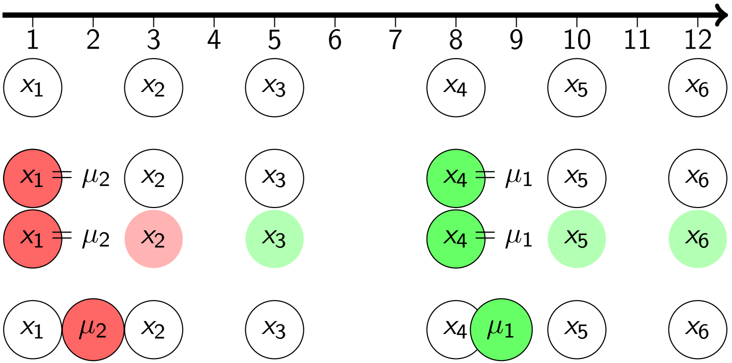
Example



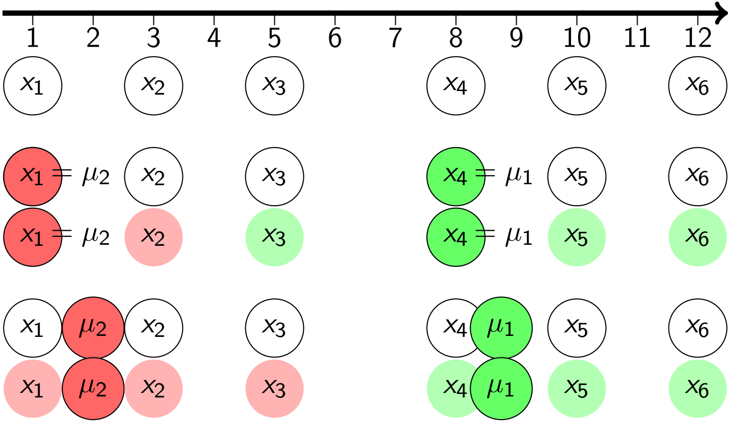
Example


1
2
3
4
5
6
7
8
9
10
11
12
x_1
x_2
x_3
x_4
x_5
x_6
$x_1 = \mu_2$
x_2
x_3
$x_4 = \mu_1$
x_5
x_6
$x_1 = \mu_2$
x_2
x_3
$x_4 = \mu_1$
x_5
x_6

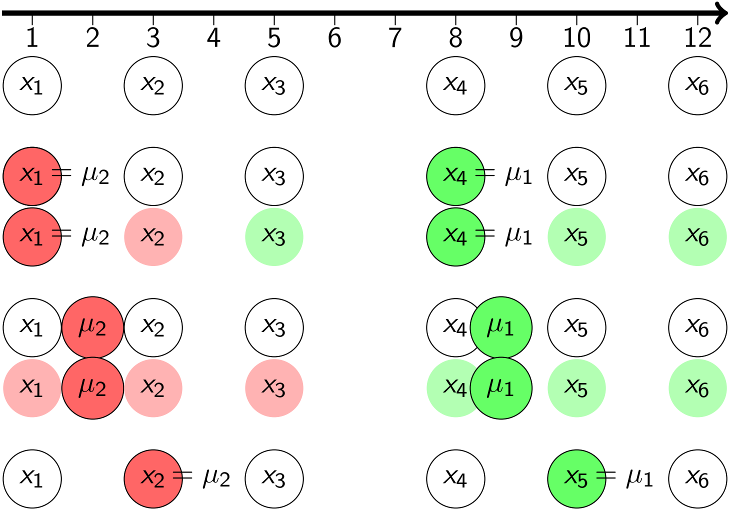
Example


1 2 3 4 5 6 7 8 9 10 11 12
x_1 x_2 x_3 x_4 x_5 x_6
$x_1 = \mu_2$ x_2 x_3 $x_4 = \mu_1$ x_5 x_6
$x_1 = \mu_2$ x_2 x_3 $x_4 = \mu_1$ x_5 x_6
x_1 μ_2 x_2 x_3 x_4 μ_1 x_5 x_6

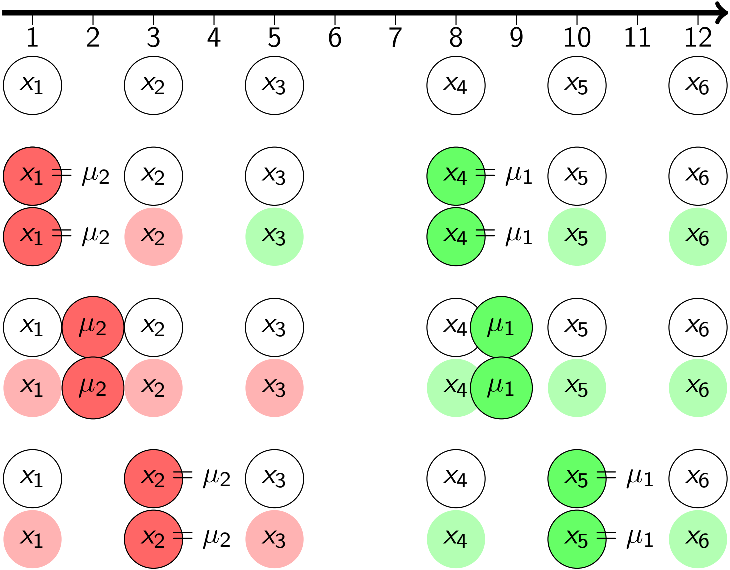
Example


1 2 3 4 5 6 7 8 9 10 11 12
x_1 x_2 x_3 x_4 x_5 x_6
$x_1 = \mu_2$ x_2 x_3 $x_4 = \mu_1$ x_5 x_6
$x_1 = \mu_2$ x_2 x_3 $x_4 = \mu_1$ x_5 x_6
x_1 μ_2 x_2 x_3 x_4 μ_1 x_5 x_6
x_1 μ_2 x_2 x_3 x_4 μ_1 x_5 x_6

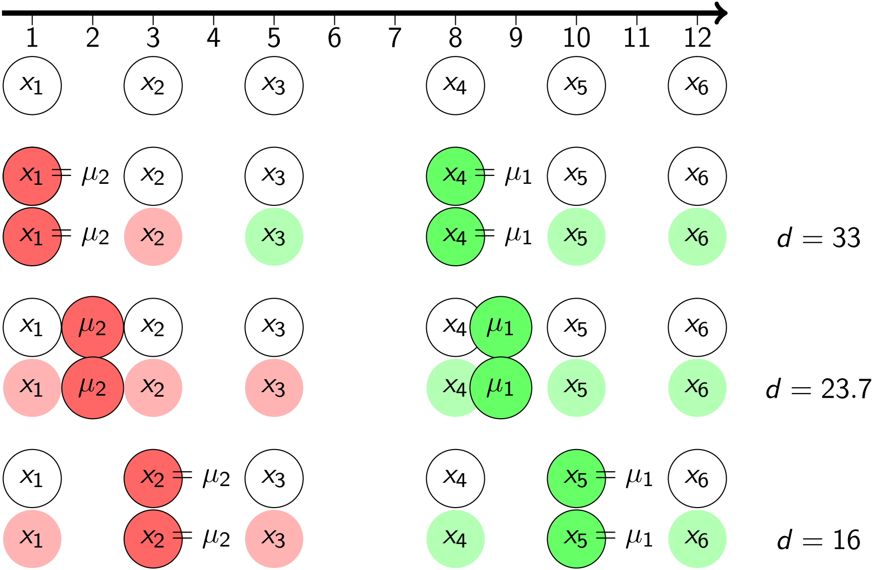
Example


1
2
3
4
5
6
7
8
9
10
11
12
x_1
x_2
x_3
x_4
x_5
x_6
$x_1 = \mu_2$
$x_4 = \mu_1$
μ_2
μ_1
$x_2 = \mu_2$
$x_5 = \mu_1$

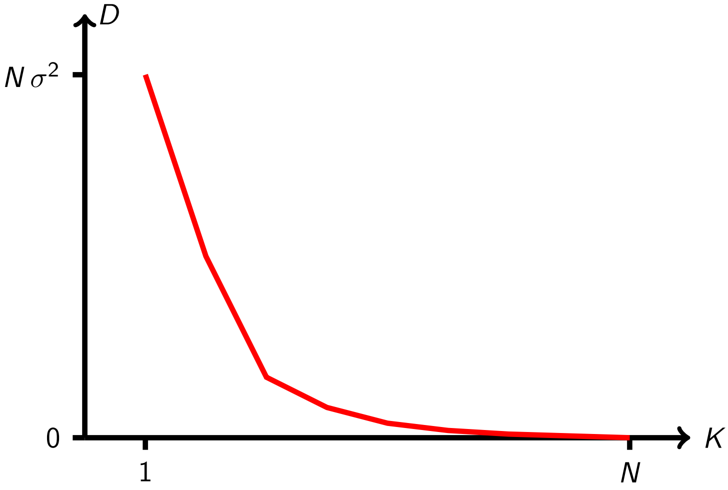
Example


1
2
3
4
5
6
7
8
9
10
11
12
x_1
x_2
x_3
x_4
x_5
x_6
$x_1 = \mu_2$
$x_4 = \mu_1$
μ_2
μ_1
$x_2 = \mu_2$
$x_5 = \mu_1$

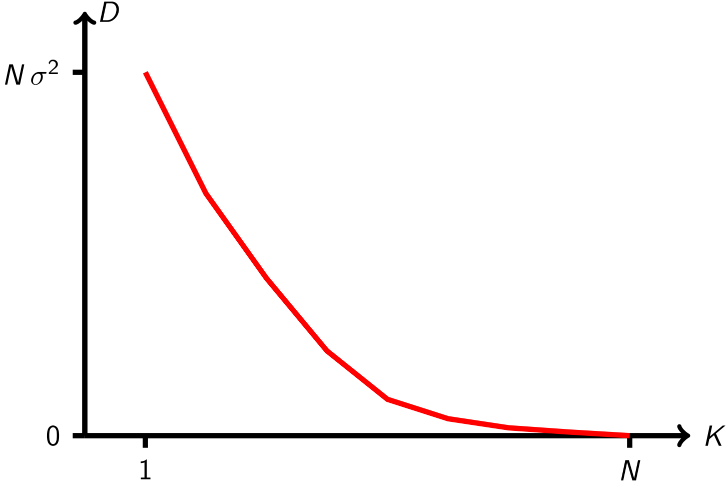
Example


1 2 3 4 5 6 7 8 9 10 11 12
x_1 x_2 x_3 x_4 x_5 x_6
$x_1 = \mu_2$ x_2 x_3 $x_4 = \mu_1$ x_5 x_6
$x_1 = \mu_2$ x_2 x_3 $x_4 = \mu_1$ x_5 x_6
$d = 33$
x_1 μ_2 x_2 x_3 x_4 μ_1 x_5 x_6
x_1 μ_2 x_2 x_3 x_4 μ_1 x_5 x_6
$d = 23.7$
x_1 $x_2 = \mu_2$ x_3 x_4 $x_5 = \mu_1$ x_6
x_1 $x_2 = \mu_2$ x_3 x_4 $x_5 = \mu_1$ x_6
$d = 16$

How Many Clusters K?



How Many Clusters K?



k-medoids: k-means for General Distances

One can generalize k-means to general distances d:

$$D(P, \mu) := \sum_{n=1}^{N} \sum_{k=1}^{K} P_{n,k} d(x_n, \mu_k)$$

k-medoids: k-means for General Distances

One can generalize k-means to general distances d:

$$D(P,\mu) := \sum_{n=1}^{N}\sum_{k=1}^{K} P_{n,k}\, d(x_n,\mu_k)$$

- step 1 assigning data points to clusters remains the same

$$P_{n,k} := \mathbb{I}(k=\ell_n), \quad \ell_n := \arg\min_{k\in\{1,\ldots,K\}} d(x_n,\mu_k)$$

- but step 2 finding the best **cluster representatives** μ_k is not solved by the mean and may be difficult in general.

k-medoids: k-means for General Distances

One can generalize k-means to general distances d:

$$D(P, \mu) := \sum_{n=1}^{N} \sum_{k=1}^{K} P_{n,k} d(x_n, \mu_k)$$

- step 1 assigning data points to clusters remains the same

$$P_{n,k} := \mathbb{I}(k = \ell_n), \quad \ell_n := \arg\min_{k \in \{1,\ldots,K\}} d(x_n, \mu_k)$$

- but step 2 finding the best **cluster representatives** μ_k is not solved by the mean and may be difficult in general.

idea **k-medoids**: choose cluster representatives out of cluster data points:

$$\mu_k := x_n, \quad n := \arg\min_{n \in \{1,\ldots,N\}: P_{n,k}=1} \sum_{\ell=1}^{N} P_{\ell,k} d(x_\ell, x_n)$$

k-medoids: k-means for General Distances

k-medoids is a “kernel method”: it requires no access to the variables, just to the distance measure.

For the **Manhattan distance/L_1 distance**, step 2 finding the best cluster representatives μ_k can be solved without restriction to cluster data points:

$$(\mu_k)_m := \text{median}\{(x_n)_m \mid P_{n,k} = 1, n \in \{1, \ldots, N\}\}, \quad m = 1, \ldots, M$$

Outline

Soft Partitions: Row Stochastic Matrices

Let $X := \{x_1, \ldots, x_N\}$ be a finite set. A $N \times K$ matrix

$$P \in [0,1]^{N \times K}$$

is called a **soft partition matrix of X** if it

1. is row-stochastic: $$\sum_{k=1}^{K} P_{n,k} = 1, \quad n \in \{1, \ldots, N\}$$
2. does not contain a zero column: $$X_{.,k} \neq (0, \ldots, 0)^T, \quad k \in \{1, \ldots, K\}$$

$P_{n,k}$ is called the

- **membership degree of instance n in class k** or the
- **cluster weight of instance n in cluster k**.

$P_{.,k}$ is called **membership vector of class k**.

SoftPart(X) denotes the set of all soft partitions of X.

Note: Soft partitions are also called **soft clusterings** and **fuzzy clusterings**.

The Soft Clustering Problem

Given

- a set $\mathcal{X}$ called **data space**, e.g., $\mathcal{X} := \mathbb{R}^M$,
- a set $X \subseteq \mathcal{X}$ called **data**, and
- a function

$$D : \bigcup_{X \subseteq \mathcal{X}} \text{SoftPart}(X) \rightarrow \mathbb{R}_0^+$$

called **distortion measure** where $D(P)$ measures how bad a soft partition $P \in \text{SoftPart}(X)$ for a data set $X \subseteq \mathcal{X}$ is,

find a soft partition $P \in \text{SoftPart}\,(X)$ with minimal distortion $D(P)$.

The Soft Clustering Problem (with given K)

Given

- a set $\mathcal{X}$ called **data space**, e.g., $\mathcal{X} := \mathbb{R}^M$,
- a set $X \subseteq \mathcal{X}$ called **data**,
- a function

$$D : \bigcup_{X \subseteq \mathcal{X}} \text{SoftPart}(X) \to \mathbb{R}_0^+$$

  called **distortion measure** where $D(P)$ measures how bad a soft partition $P \in \text{SoftPart}(X)$ for a data set $X \subseteq \mathcal{X}$ is, and

- a number $K \in \mathbb{N}$ of clusters,

find a soft partition $P \in \text{SoftPart}_K(X) \subseteq [0,1]^{|X| \times K}$ with K clusters with minimal distortion $D(P)$.

Mixture Models

Mixture models assume that there exists an **unobserved nominal variable** Z with K levels:

$$p(X, Z) = p(Z)p(X \mid Z) = \prod_{k=1}^{K} (\pi_k p(X \mid Z = k))^{\mathbb{I}(Z=k)}$$

The **complete data loglikelihood** of the **completed data** (X, Z) then is

$$\ell(\Theta; X, Z) := \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}(Z_n = k)(\ln \pi_k + \ln p(X = x_n \mid Z = k; \theta_k)$$

$$\text{with } \Theta := (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$$

ℓ cannot be computed because z_n's are unobserved.

Mixture Models: Expected Loglikelihood

Given an estimate $\Theta^{(t-1)}$ of the parameters, mixtures aim to optimize the **expected complete data loglikelihood**:

$$Q(\Theta;\Theta^{(t-1)}) := \mathbb{E}[\ell(\Theta; X, Z) \mid \Theta^{(t-1)}]$$
$$= \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}[\mathbb{I}(Z_n = k) \mid x_n, \Theta^{(t-1)}](\ln \pi_k + \ln p(X = x_n \mid Z = k; \theta_k))$$

which is relaxed to

$$Q(\Theta, r; \Theta^{(t-1)}) = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{n,k}(\ln \pi_k + \ln p(X = x_n \mid Z = k; \theta_k))$$
$$+ (r_{n,k} - \mathbb{E}[\mathbb{I}(Z_n = k) \mid x_n, \Theta^{(t-1)}])^2$$

Mixture Models: Expected Loglikelihood

Block coordinate descent (**EM algorithm**): alternate until convergence

1. **expectation step**:

$$
\begin{aligned}
r_{n,k}^{(t-1)} &:= \mathbb{E}[\mathbb{I}(Z_n = k) \mid x_n, \Theta^{(t-1)}] = p(Z = k \mid X = x_n; \Theta^{(t-1)}) \\
&= \frac{p(X = x_n \mid Z = k; \Theta^{(t-1)}) p(Z = k; \Theta^{(t-1)})}{\sum_{k'=1}^{K} p(X = x_n \mid Z = k'; \Theta^{(t-1)}) p(Z = k'; \Theta^{(t-1)})} \\
&= \frac{p(X = x_n \mid Z = k; \theta_k^{(t-1)}) \pi_k^{(t-1)}}{\sum_{k'=1}^{K} p(X = x_n \mid Z = k'; \theta_k^{(t-1)}) \pi_k^{(t-1)}}
\end{aligned} \quad (0)
$$

2. **maximization step**:

$$
\begin{aligned}
\Theta^{(t)} &:= \arg\max_{\Theta} Q(\Theta, r^{(t-1)}; \Theta^{(t-1)}) \\
&= \arg\max_{\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{n,k} (\ln \pi_k + \ln p(X = x_n \mid Z = k; \theta_k))
\end{aligned}
$$

Mixture Models: Expected Loglikelihood

2. **maximization step**:

$$\Theta^{(t)} = \underset{\pi_1,\ldots,\pi_K,\theta_1,\ldots,\theta_K}{\arg\max} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{n,k} (\ln \pi_k + \ln p(X = x_n \mid Z = k; \theta_k))$$

$$\rightsquigarrow \quad \pi_k^{(t)} = \frac{\sum_{n=1}^{N} r_{n,k}}{N} \tag{1}$$

$$\sum_{n=1}^{N} \frac{r_{n,k}}{p(X = x_n \mid Z = k; \theta_k)} \frac{\partial p(X = x_n \mid Z = k; \theta_k)}{\partial \theta_k} = 0, \quad \forall k \tag{$*$}$$

(*) needs to be solved for specific cluster specific distributions $p(X|Z)$.

Gaussian Mixtures

Gaussian mixtures:

- use Gaussians for $p(X|Z)$:

$$p(X = x \mid Z = k) = \frac{1}{\sqrt{(2\pi)^M |\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}, \quad \theta_k := (\mu_k, \Sigma_k$$

$$\rightsquigarrow \quad \mu_k^{(t)} = \frac{\sum_{n=1}^N r_{n,k}^{(t-1)} x_n}{\sum_{n=1}^N r_{n,k}^{(t-1)}} \tag{2}$$

$$\Sigma_k^{(t)} = \frac{\sum_{n=1}^N r_{n,k}^{(t-1)} (x_n - \mu_k^{(t)})(x_n - \mu_k^{(t)})^T}{\sum_{n=1}^N r_{n,k}^{(t-1)}}$$

$$= \frac{\sum_{n=1}^N r_{n,k}^{(t-1)} x_n x_n^T - \mu_k^{(t)} \mu_k^{(t)\,T}}{\sum_{n=1}^N r_{n,k}^{(t-1)}} \tag{3}$$

Gaussian Mixtures: EM Algorithm, Summary

1. **expectation step**: $\forall n, k$

$$\tilde{r}_{n,k}^{(t-1)} = \pi_k^{(t-1)} \frac{1}{\sqrt{(2\pi)^M |\Sigma_k^{(t-1)}|}} e^{-\frac{1}{2}(x_n - \mu_k^{(t-1)})^T \Sigma_k^{(t-1)-1} (x_n - \mu_k^{(t-1)})} \quad (0a)$$

$$r_{n,k}^{(t-1)} = \frac{\tilde{r}_{n,k}^{(t-1)}}{\sum_{k'=1}^K \tilde{r}_{n,k'}^{(t-1)}} \quad (0b)$$

2. **maximization step**: $\forall k$

$$\pi_k^{(t)} = \frac{\sum_{n=1}^N r_{n,k}^{(t-1)}}{N} \quad (1)$$

$$\mu_k^{(t)} = \frac{\sum_{n=1}^N r_{n,k}^{(t-1)} x_n}{\sum_{n=1}^N r_{n,k}^{(t-1)}} \quad (2)$$

$$\Sigma_k^{(t)} = \frac{\sum_{n=1}^N r_{n,k}^{(t-1)} x_n x_n^T - \mu_k^{(t)} \mu_k^{(t)T}}{\sum_{n=1}^N r_{n,k}^{(t-1)}} \quad (3)$$

Gaussian Mixtures for Soft Clustering

- The **responsibilities** $r \in [0,1]^{N\times K}$ are a soft partition.

$$P := r$$

- The negative expected loglikelihood can be used as cluster distortion:

$$D(P) := -\max_{\Theta} Q(\Theta, P)$$

- To optimize D, we simply can run EM.

Gaussian Mixtures for Soft Clustering

- The **responsibilities** $r \in [0,1]^{N \times K}$ are a soft partition.

$$P := r$$

- The negative expected loglikelihood can be used as cluster distortion:

$$D(P) := -\max_{\Theta} Q(\Theta, P)$$

- To optimize D, we simply can run EM.

For hard clustering:

- assign points to the cluster with highest responsibility (**hard EM**):

$$r_{n,k}^{(t-1)} = \mathbb{I}(k = \arg\max_{k'=1,\ldots,K} \tilde{r}_{n,k'}^{(t-1)}) \qquad (0b')$$

Gaussian Mixtures: EM Algorithm

cluster-soft-em$(\mathcal{D} := \{x_1, \ldots, x_N\} \subseteq \mathbb{R}^M, K \in \mathbb{N}, \epsilon \in \mathbb{R}^+)$:

$\quad \tilde{r}^{(0)}_{n,k} \sim \text{unif}([0,1]), \quad n := 1, \ldots, N, k := 1, \ldots, K$

$\quad r^{(0)}_{n,k} := \tilde{r}^{(0)}_{n,k} / \sum_{k'=1}^{K} \tilde{r}^{(0)}_{n,k'}, \quad n := 1, \ldots, N, k := 1, \ldots, K$

$\quad$ repeat

$\quad\quad t := t + 1$

$\quad\quad$ for $k := 1 : K$:

$\quad\quad\quad \pi^{(t)}_k := \sum_{n=1}^{N} r^{(t-1)}_{n,k} / N$

$\quad\quad\quad \mu^{(t)}_k := \sum_{n=1}^{N} r^{(t-1)}_{n,k} x_n / \sum_{n=1}^{N} r^{(t-1)}_{n,k}$

$\quad\quad\quad \Sigma^{(t)}_k := (\sum_{n=1}^{N} r^{(t-1)}_{n,k} x_n x_n^T - \mu^{(t)}_k \mu^{(t)\,T}_k) / \sum_{n=1}^{N} r^{(t-1)}_{n,k}$

$\quad\quad$ for $n := 1 : N$:

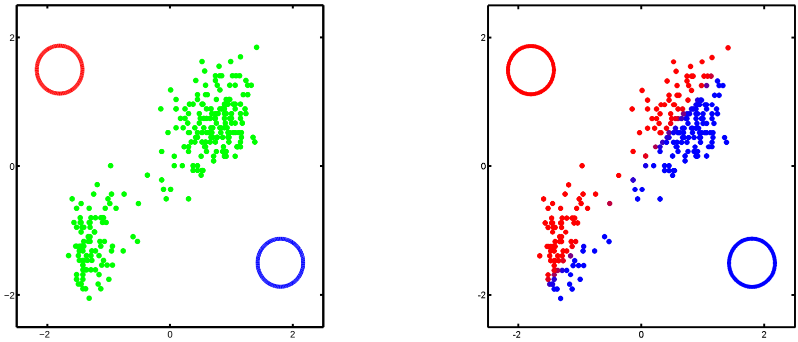
$\quad\quad\quad \tilde{r}^{(t)}_{n,k} := \pi^{(t)}_k \frac{1}{\sqrt{(2\pi)^M |\Sigma^{(t)}_k|}} e^{-\frac{1}{2}(x_n - \mu^{(t)}_k)^T \Sigma^{(t)\,-1}_k (x_n - \mu^{(t)}_k)}, \quad k := 1 : K$

$\quad\quad\quad r^{(t)}_{n,k} := \tilde{r}^{(t)}_{n,k} / \sum_{k'=1}^{K} \tilde{r}^{(t)}_{n,k'}, \quad k := 1 : K$

$\quad$ until $||r^{(t)} - r^{(t-1)}|| < \epsilon$

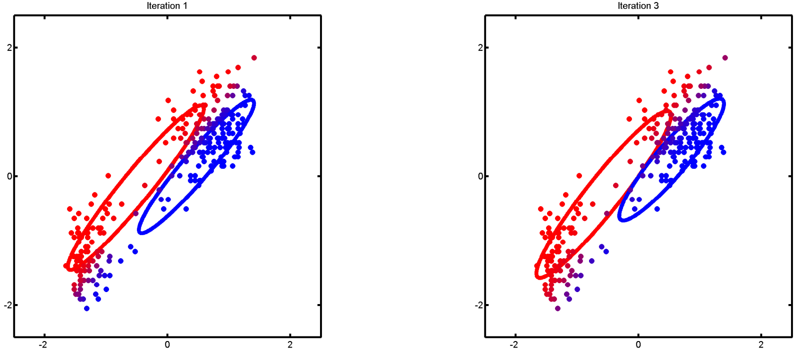
$\quad$ return $\pi^{(t)}, \mu^{(t)}, \Sigma^{(t)}_k, r^{(t)}$

Gaussian Mixtures for Soft Clustering / Example



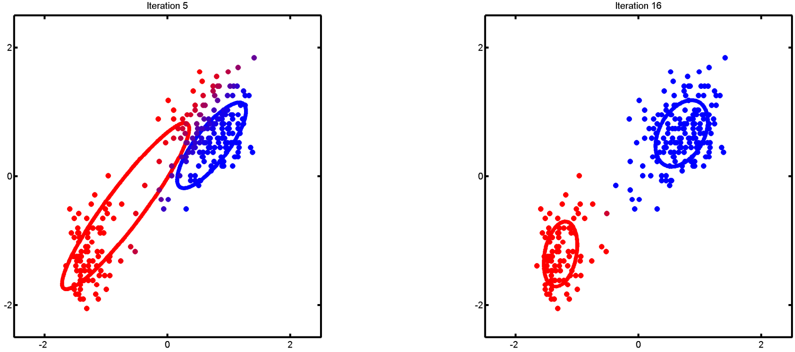
[Murphy, 2012, fig. 11.11

Gaussian Mixtures for Soft Clustering / Example



[Murphy, 2012, fig. 11.11

Gaussian Mixtures for Soft Clustering / Example



[Murphy, 2012, fig. 11.11

Model-based Cluster Analysis

Different parametrizations of the covariance matrices Σ_k restrict possible **cluster shapes**:

- full Σ:
  all sorts of ellipsoid clusters.
- diagonal Σ:
  ellipsoid clusters with axis-parallel axes
- unit Σ:
  spherical clusters.

One also distinguishes

- cluster-specific Σ_k:
  each cluster can have its own shape.
- shared $\Sigma_k = \Sigma$:
  all clusters have the same shape.

k-means: Hard EM with spherical clusters

1. **expectation step**: $\forall n, k$

$$\tilde{r}_{n,k}^{(t-1)} = \frac{1}{\sqrt{(2\pi)^M |\Sigma_k^{(t-1)}|}} e^{-\frac{1}{2}(x_n - \mu_k^{(t-1)})^T \Sigma_k^{(t-1)\,-1} (x_n - \mu_k^{(t-1)})} \quad (0a)$$

$$= \frac{1}{\sqrt{(2\pi)^M}} e^{-\frac{1}{2}(x_n - \mu_k^{(t-1)})^T (x_n - \mu_k^{(t-1)})}$$

$$r_{n,k}^{(t-1)} = \mathbb{I}(k = \arg\max_{k'=1,\ldots,K} \tilde{r}_{n,k'}^{(t-1)}) \quad (0b')$$

$$\arg\max_{k'=1,\ldots,K} \tilde{r}_{n,k'}^{(t-1)} = \arg\max_{k'=1,\ldots,K} \frac{1}{\sqrt{(2\pi)^M}} e^{-\frac{1}{2}(x_n - \mu_k^{(t-1)})^T (x_n - \mu_k^{(t-1)})}$$

$$= \arg\max_{k'=1,\ldots,K} -(x_n - \mu_k^{(t-1)})^T (x_n - \mu_k^{(t-1)})$$

$$= \arg\min_{k'=1,\ldots,K} ||x_n - \mu_k^{(t-1)}||^2$$

Outline

Hierarchies

Let X be a set.

A tree (H, E), $E \subseteq H \times H$ edges pointing towards root

- with leaf nodes h corresponding bijectively to elements $x_h \in X$
- plus a surjective map $\mathrm{L} : H \rightarrow \{0, \ldots, d\}, d \in \mathbb{N}$ with
  - $L(\text{root}) = 0$ and
  - $L(h) = d$ for all leaves $h \in H$ and
  - $L(h) \leq L(g)$ for all $(g, h) \in E$

  called **level map**

is called an **hierarchy over X**.

Hierarchies

Let X be a set.
A tree (H, E), $E \subseteq H \times H$ edges pointing towards root

- with leaf nodes h corresponding bijectively to elements $x_h \in X$
- plus a surjective map $\mathrm{L} : H \rightarrow \{0, \ldots, d\}, d \in \mathbb{N}$ with
  - $L(\text{root}) = 0$ and
  - $L(h) = d$ for all leaves $h \in H$ and
  - $L(h) \leq L(g)$ for all $(g, h) \in E$

  called **level map**

is called an **hierarchy over X**.

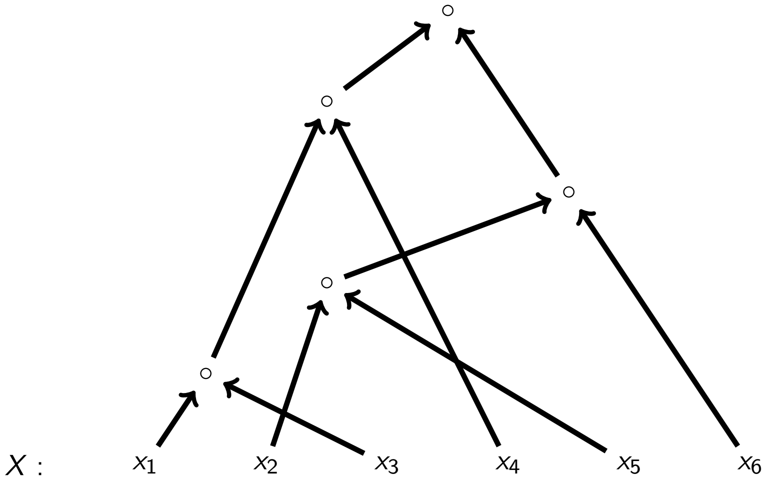
d is called the **depth** of the hierarchy.

$\text{Hier}(X)$ denotes the set of all hierarchies over X.

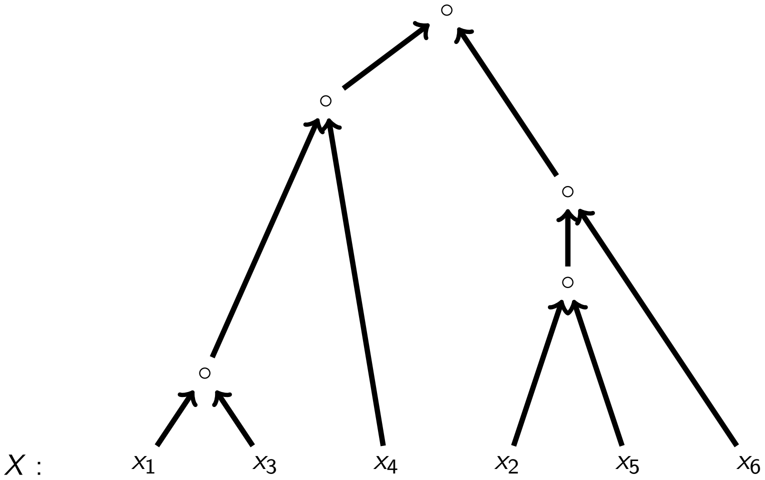
Hierarchies / Example

X : x_1 x_2 x_3 x_4 x_5 x_6

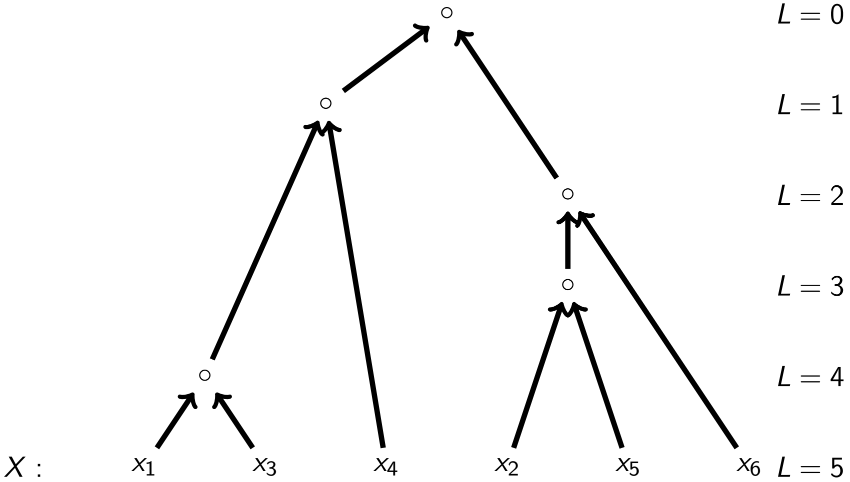
Hierarchies / Example



Hierarchies / Example



Hierarchies / Example



Hierarchies: Nodes Correspond to Subsets

Let (H, E) be such an hierarchy:

- nodes of an hierarchy correspond to subsets of X:
  - leaf nodes h correspond to a singleton subset by definition.

$$\text{subset}(h) := \{x_h\}, \quad x_h \in X \text{ corresponding to leaf } h$$

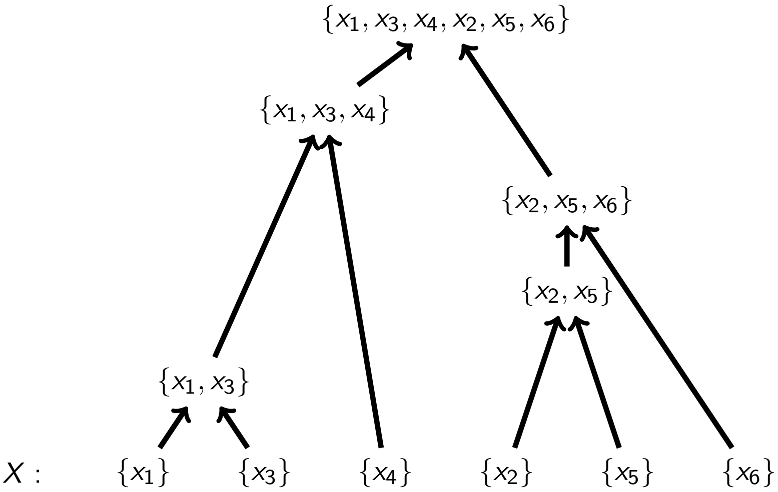
  - interior nodes h correspond to the union of the subsets of their children:

$$\text{subset}(h) := \bigcup_{\substack{g \in H \\ (g,h) \in E}} \text{subset}(g)$$

- thus the root node h corresponds to the full set X:

$$\text{subset}(h) = X$$

Hierarchies: Nodes Correspond to Subsets



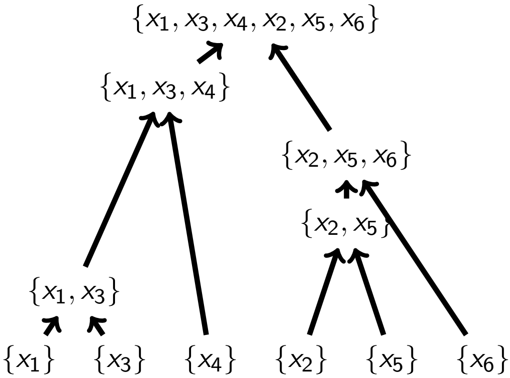
Hierarchies: Levels Correspond to Partitions

Let (H, E) be such an hierarchy:

- levels $\ell \in \{0, \ldots, d\}$ correspond to partitions

$$P_\ell(H, L) := \{\text{subset}(h) \mid h \in H, L(h) \geq \ell, \nexists g \in H : L(g) \geq \ell, \\ \text{subset}(h) \subsetneq \text{subset}(g)\}$$

Hierarchies: Levels Correspond to Partitions



$\{\{x_1, x_3, x_4, x_2, x_5, x_6\}\}$

$\{\{x_1, x_3, x_4\}, \{x_2, x_5, x_6\}\}$

$\{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5, x_6\}\}$

$\{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}, \{x_6\}\}$

$\{\{x_1, x_3\}, \{x_4\}, \{x_2\}, \{x_5\}, \{x_6\}\}$

$\{\{x_1\}, \{x_3\}, \{x_4\}, \{x_2\}, \{x_5\}, \{x_6\}\}$

The Hierarchical Cluster Analysis Problem

Given

- a set $\mathcal{X}$ called **data space**, e.g., $\mathcal{X} := \mathbb{R}^M$,
- a set $X \subseteq \mathcal{X}$ called **data** and
- a function

$$D : \bigcup_{X \subseteq \mathcal{X}} \text{Hier}(X) \rightarrow \mathbb{R}_0^+$$

called **distortion measure** where $D(P)$ measures how bad a hierarchy $H \in \text{Hier}(X)$ for a data set $X \subseteq \mathcal{X}$ is,

find a hierarchy $H \in \text{Hier}(X)$ with minimal distortion $D(H)$.

Distortions for Hierarchies

Examples for distortions for hierarchies:

$$D(H) := \sum_{K=1}^{N} \tilde{D}(P_K(H))$$

where

- $P_K(H)$ denotes the partition at level $K-1$ (with K classes) and
- $\tilde{D}$ denotes a distortion for partitions.

Agglomerative and Divisive Hierarchical Clustering

Hierarchies are usually learned by greedy search level by level:

- **agglomerative clustering:**
  1. start with the singleton partition P_N:

     $$P_N := \{X_k \mid k = 1, \ldots, N\}, \quad X_k := \{x_k\}, \quad k = 1, \ldots, N$$

  2. in each step $K = N, \ldots, 2$ build P_{K-1} by joining the two clusters $k, \ell \in \{1, \ldots, K\}$ that lead to the minimal distortion

     $$D(\{X_1, \ldots, \cancel{X_k}, \ldots, \cancel{X_\ell}, \ldots, X_K, X_k \cup X_\ell\})$$

Agglomerative and Divisive Hierarchical Clustering

Hierarchies are usually learned by greedy search level by level:

- **divisive clustering:**
  1. start with the all partition P_1:

$$P_1 := \{X\}$$

  2. in each step $K = 1, N-1$ build P_{K+1} by splitting one cluster X_k in two clusters X'_k, X'_ℓ that lead to the minimal distortion

$$D(\{X_1, \ldots, \cancel{X_k}, \ldots, X_K, X'_k, X'_\ell), \quad X_k = X'_k \cup X'_\ell$$

Class-wise Defined Partition Distortions

If the partition distortion can be written as a sum of distortions of its classes,

$$D(\{X_1, \ldots, X_K\}) = \sum_{k=1}^{K} \tilde{D}(X_k)$$

then the optimal pair does only depend on X_k, X_ℓ:

$$\begin{aligned} & D(\{X_1, \ldots, \cancel{X_k}, \ldots, \cancel{X_\ell}, \ldots, X_K, X_k \cup X_\ell) \\ & - D(\{X_1, \ldots, X_k, \ldots, X_\ell, \ldots, X_K) \\ & = \tilde{D}(X_k \cup X_\ell) - (\tilde{D}(X_k) + \tilde{D}(X_\ell)) \end{aligned}$$

Closest Cluster Pair Partition Distortions

For a **cluster distance**

$$\tilde{d} : \mathcal{P}(X) \times \mathcal{P}(X) \to \mathbb{R}_0^+$$
$$\text{with} \quad \tilde{d}(A \cup B, C) \geq \min\{\tilde{d}(A, C), \tilde{d}(B, C)\}, \quad A, B, C \subseteq X$$

a partition can be judged by the closest cluster pair it contains:

$$D(\{X_1, \ldots, X_K\}) := \min_{\substack{k,\ell=1,K \\ k \neq \ell}} \tilde{d}(X_k, X_\ell)$$

Such a distortion has to be maximized.

To increase it, the closest cluster pair has to be joined.

Single Link Clustering

$$d_{\mathrm{sl}}(A, B) := \min_{x \in A, y \in B} d(x, y), \quad A, B \subseteq X$$

Complete Link Clustering

$$d_{\text{cl}}(A, B) := \max_{x \in A, y \in B} d(x, y), \quad A, B \subseteq X$$

Average Link Clustering

$$d_{\mathrm{al}}(A, B) := \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y), \quad A, B \subseteq X$$

Recursion Formulas for Cluster Distances

$$
\begin{aligned}
d_{\text{sl}}(X_i \cup X_j, X_k) &:= \min_{x \in X_i \cup X_j, y \in X_k} d(x, y) \\
&= \min\{\min_{x \in X_i, y \in X_k} d(x, y), \min_{x \in X_j, y \in X_k} d(x, y)\} \\
&= \min\{d_{\text{sl}}(X_i, X_k), d_{\text{sl}}(X_j, X_k)\}
\end{aligned}
$$

Recursion Formulas for Cluster Distances

$$
\begin{aligned}
d_{\text{sl}}(X_i \cup X_j, X_k) &= \min\{d_{\text{sl}}(X_i, X_k), d_{\text{sl}}(X_j, X_k)\} \\
d_{\text{cl}}(X_i \cup X_j, X_k) &:= \max_{x \in X_i \cup X_j, y \in X_k} d(x, y) \\
&= \max\{\max_{x \in X_i, y \in X_k} d(x, y), \max_{x \in X_j, y \in X_k} d(x, y)\} \\
&= \max\{d_{\text{cl}}(X_i, X_k), d_{\text{cl}}(X_j, X_k)\}
\end{aligned}
$$

Recursion Formulas for Cluster Distances

$$d_{\text{sl}}(X_i \cup X_j, X_k) = \min\{d_{\text{sl}}(X_i, X_k), d_{\text{sl}}(X_j, X_k)\}$$

$$d_{\text{cl}}(X_i \cup X_j, X_k) = \max\{d_{\text{cl}}(X_i, X_k), d_{\text{cl}}(X_j, X_k)\}$$

$$\begin{aligned}
d_{\text{al}}(X_i \cup X_j, X_k) &:= \frac{1}{|X_i \cup X_j||X_k|} \sum_{x \in X_i \cup X_j, y \in X_k} d(x, y) \\
&= \frac{|X_i|}{|X_i \cup X_j|} \frac{1}{|X_i||X_k|} \sum_{x \in X_i, y \in X_k} d(x, y) \\
&\quad + \frac{|X_j|}{|X_i \cup X_j|} \frac{1}{|X_j||X_k|} \sum_{x \in X_j, y \in X_k} d(x, y) \\
&= \frac{|X_i|}{|X_i| + |X_j|} d_{\text{al}}(X_i, X_k) + \frac{|X_j|}{|X_i| + |X_j|} d_{\text{al}}(X_j, X_k)
\end{aligned}$$

Recursion Formulas for Cluster Distances

$$d_{\text{sl}}(X_i \cup X_j, X_k) = \min\{d_{\text{sl}}(X_i, X_k), d_{\text{sl}}(X_j, X_k)\}$$
$$d_{\text{cl}}(X_i \cup X_j, X_k) = \max\{d_{\text{cl}}(X_i, X_k), d_{\text{cl}}(X_j, X_k)\}$$
$$d_{\text{al}}(X_i \cup X_j, X_k) = \frac{|X_i|}{|X_i| + |X_j|} d_{\text{al}}(X_i, X_k) + \frac{|X_j|}{|X_i| + |X_j|} d_{\text{al}}(X_j, X_k)$$

$\rightsquigarrow$ agglomerative hierarchical clustering requires to compute the **distance matrix** $D \in \mathbb{R}^{N \times N}$ only once:

$$D_{n,\ell} := d(x_n, x_\ell), \quad n, \ell = 1, \ldots, N$$

Thus it is a **kernel method**.

Conclusion (1/2)

- Cluster analysis aims at **detecting latent groups** in data, without labeled examples ($\leftrightarrow$ **record linkage**).
- Latent groups can be described in three different granularities:
  - **partitions** segment data into K subsets (**hard clustering**).
  - **soft clusterings / row-stochastic matrices** build overlapping groups to which data points can belong with some **membership degree** (**soft clustering**).
  - **hierarchies** structure data into an hierarchy, in a sequence of consistent partitions (**hierarchical clustering**).
- **k-means** finds a K-partition by finding K **cluster centers** with smallest **Euclidean distance** to all their cluster points.
- **k-medoids** generalizes k-means to **general distances**; it finds a K-partition by selecting K data points as **cluster representatives** with smallest distance to all their cluster points.

Conclusion (2/2)

- **Gaussian Mixture Models** find soft clusterings by modeling data by a class-specific multivariate Gaussian distribution $p(X \mid Z)$ and estimating expected class memberships (**expected likelihood**).
- The **Expectation Maximiation Algorithm (EM)** can be used to learn Gaussian Mixture Models via block coordinate descent.
- k-means is a special case of a Gaussian Mixture Model
  - with hard/binary cluster memberships (**hard EM**) and
  - **spherical cluster shapes**.
- **hierarchical single link, complete link and average link methods**
  - find a hierarchy by greedy search over consistent partitions,
  - starting from the singleton parition (**agglomerative**)
  - being efficient due to **recursion formulas**,
  - requiring only a distance matrix.

Readings

- k-means:
  - Hastie et al. [2005], ch. 14.3.6, 13.2.3, 8.5 Bishop [2006], ch. 9.1, Murphy [2012], ch. 11.4.2
- hierarchical cluster analysis:
  - Hastie et al. [2005], ch. 14.3.12, Murphy [2012], ch. 25.5. Press et al. [2007], ch. 16.4.
- Gaussian mixtures:
  - Hastie et al. [2005], ch. 14.3.7, Bishop [2006], ch. 9.2, Murphy [2012], ch. 11.2.3, Press et al. [2007], ch. 16.1.

References

Christopher M. Bishop. *Pattern Recognition and Machine Learning*, volume 1. springer New York, 2006.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, volume 27. Springer, 2005.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes*. Cambridge University Press, 3rd edition, 2007.