Machine Learning

B. Unsupervised Learning
B.1 Cluster Analysis

Lars Schmidt-Thieme, Nicolas Schilling

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

Outline

1. K-Means & K-Medoids

2. Mixture Models & EM Algorithm

Syllabus

Wed. 21.10.	(1)	0. Introduction
		A. Supervised Learning
Wed. 28.10.	(2)	A.1 Linear Regression
Wed. 04.11.	(3)	A.2 Linear Classification
Wed. 11.11.	(4)	A.3 Regularization (Given by Martin)
Wed. 18.11.	(5)	A.4 High-dimensional Data
Wed. 25.11.	(6)	A.5 Nearest-Neighbor Models
Wed. 02.12.	(7)	A.6 Support Vector Machines
Wed. 09.12.	(8)	A.7 Decision Trees
Wed. 06.01.	(9)	A.8 A First Look at Bayesian and Markov Networks
		Extra:
Wed. 16.12.	(E)	Invited Talk: Recommender Systems in work at Volkswagen
		B. Unsupervised Learning
Wed. 20.01.	(10)	B.1 Clustering
Wed. 27.01.	(11)	B.2 Dimensionality Reduction
Wed. 03.02.	(12)	B.3 Frequent Pattern Mining
		C. Reinforcement Learning
Wed. ??.??.	(13)	C.1 State Space Models
Wed. ??.??.	(14)	C.2 Markov Decision Processes

Outline

Unsupervised Learning

For **supervised learning** problems, we were always given some training data

$$\mathcal{D}^{\text{train}} = \{(x_1, y_1), ..., (x_N, y_N)\}$$

- $x_i \in X$ corresponds to a measurement (a data instance)
- $y_i \in Y$ is a label

Then the goal was to find a model $f : X \mapsto Y$ with minimal training error and decent generalization ability.

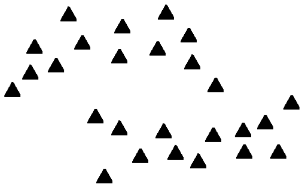
In **unsupervised learning**, there are **no labels given!!**

Cluster Analysis

Assume we have a dataset

$$\mathcal{D}^{\text{train}} = \{x_1, ..., x_N\}$$

with no further information given.



- **cluster analysis** tries to find commonalities among all data instances to group them into K many groups.
- we have to find a partition of X.

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**: $X_k \neq \emptyset, \quad k \in \{1, \ldots, K\}$.

The sets X_k are also called **clusters**, a partition P a **clustering**.
$K \in \mathbb{N}$ is called **number of clusters**.

Partitions

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

$$p : \{1, \ldots, |X|\} \rightarrow \{1, \ldots, K\}$$

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

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

x_i	$p(x_i)$
x_1	1
x_2	2
x_3	2
x_4	1

$$p^{-1}(1) = \{x_1, x_4\} \qquad p^{-1}(2) = \{x_2, x_3\}$$

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

The sets $X_k := \{x_i \mid P_{i,k} = 1\}$ form a partition $P := \{X_1, \ldots, X_K\}$.

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

Partitions

For the example given through:

x_i	$p(x_i)$
x_1	1
x_2	2
x_3	2
x_4	1

the partition matrix would look like:

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

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

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

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,

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

The Cluster Analysis Problem (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

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

  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 $D(P)$.

Distortion Measures: Intuition

Assume we have the following data and two cluster centers μ_1 and μ_2:

- we would assign the left points to the red cluster, the right points to the blue cluster
- we want a distortion measure that encourages this behaviour

k-means: Distortion Sum of Distances to Cluster Centers

Find a partition P such that the sum of squared distances to cluster centers in minimal:

$$D(P) := \sum_{k=1}^{K} \sum_{\substack{i=1: \\ P_{i.k}=1}}^{n} ||x_i - \mu_k||^2$$

with

$$\mu_k := \text{mean } \{x_i \mid P_{i,k} = 1, i = 1, \ldots, n\}$$

k-means: Distortion Sum of Distances to Cluster Centers

Find a partition P such that the sum of squared distances to cluster centers in minimal:

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

with

$$\mu_k := \frac{\sum_{i=1}^{n} P_{i,k} x_i}{\sum_{i=1}^{n} P_{i,k}} = \text{mean } \{x_i \mid P_{i,k} = 1, i = 1, \ldots, n\}$$

On the role of K

Minimizing D over partitions with varying number of clusters (varying K) does not make sense

- a singleton clustering, where each point is its own cluster center and $K = N$ has minimal D
- only minimizing with a 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:

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

k-means: Minimizing Distances to Cluster Centers

Add cluster centers μ as auxiliary optimization variables:

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

Block coordinate descent:

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

$$P_{i,k} := \delta(k = \ell_i), \quad \ell_i := \operatorname*{arg\,min}_{k \in \{1,\ldots,K\}} ||x_i - \mu_k||^2$$

k-means: Minimizing Distances to Cluster Centers

Add cluster centers μ as auxiliary optimization variables:

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

Block coordinate descent:

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

$$P_{i,k} := \delta(k = \ell_i), \quad \ell_i := \underset{k \in \{1,\dots,K\}}{\arg\min} ||x_i - \mu_k||^2$$

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

$$\mu_k := \frac{\sum_{i=1}^{n} P_{i,k} x_i}{\sum_{i=1}^{n} P_{i,k}}$$

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 chosen cluster centers as next cluster center

$$\mu_k := x_i, \quad i := \arg\max_{i \in \{1,\ldots,n\}} \sum_{\ell=1}^{k-1} ||x_i - \mu_\ell||^2, \quad k = 2, \ldots, K$$

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 chosen cluster centers as next cluster center

$$\mu_k := x_i, \quad i := \arg\max_{i \in \{1,\ldots,n\}} \sum_{\ell=1}^{k-1} ||x_i - \mu_\ell||^2, \quad k = 2, \ldots, K$$

Different initializations may lead to different local minima.

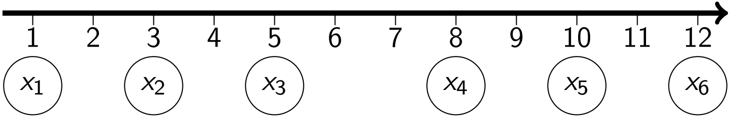
- run k-means with different random initializations and
- keep only the one with the smallest distortion (**random restarts**).

k-means Algorithm

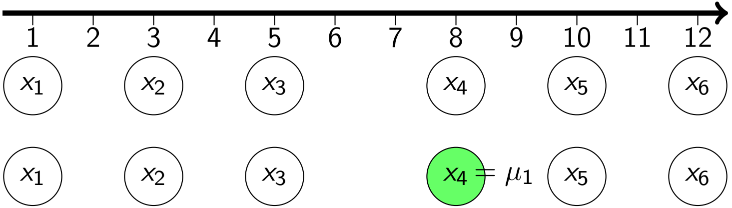
1: **procedure** CLUSTER-KMEANS$(\mathcal{D} := \{x_1, \ldots, x_N\} \subseteq \mathbb{R}^M, K \in \mathbb{N}, \epsilon \in \mathbb{R}^+)$
2: $\quad i_1 \sim \text{unif}(\{1, \ldots, N\}), \mu_1 := x_{i_1}$
3: $\quad$ **for** $k := 2, \ldots, K$ **do**
4: $\quad\quad i_k := \arg\max_{n \in \{1,\ldots,N\}} \sum_{\ell=1}^{k-1} ||x_n - \mu_\ell||, \mu_i := x_{i_k},.$
5: $\quad$ **repeat**
6: $\quad\quad \mu^{\text{old}} := \mu$
7: $\quad\quad$ **for** $n := 1, \ldots, N$ **do**
8: $\quad\quad\quad P_n := \arg\min_{k \in \{1,\ldots,K\}} ||x_n - \mu_k||$
9: $\quad\quad$ **for** $k := 1, \ldots, K$ **do**
10: $\quad\quad\quad \mu_k := \text{mean } \{x_n \mid P_n = k\}$
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 6 and 9) can be combined in one loop.

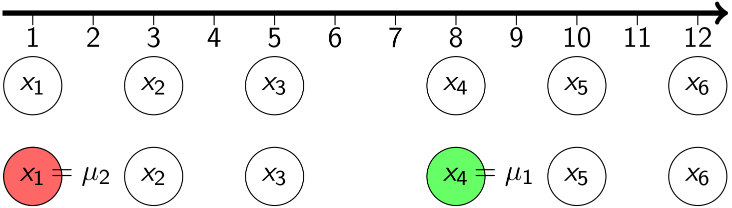
Example



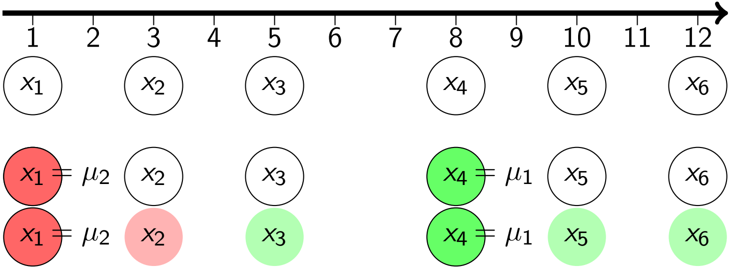
Example



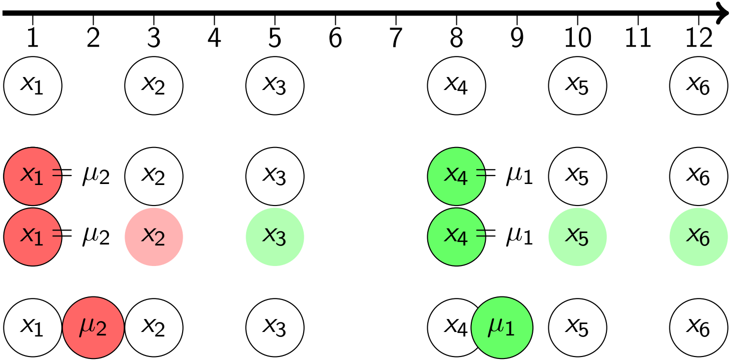
Example



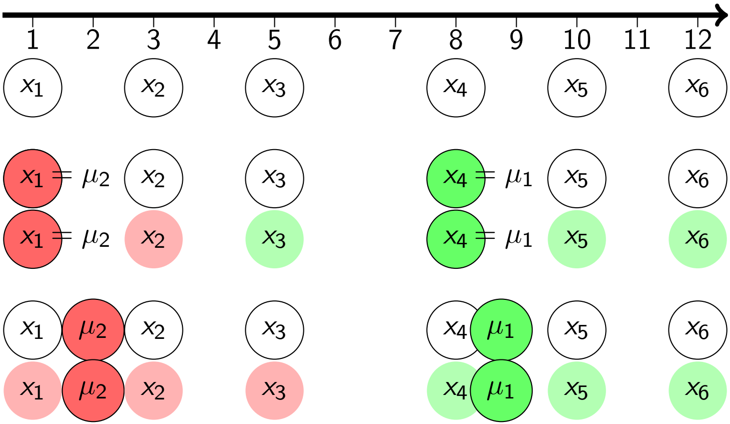
Example


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

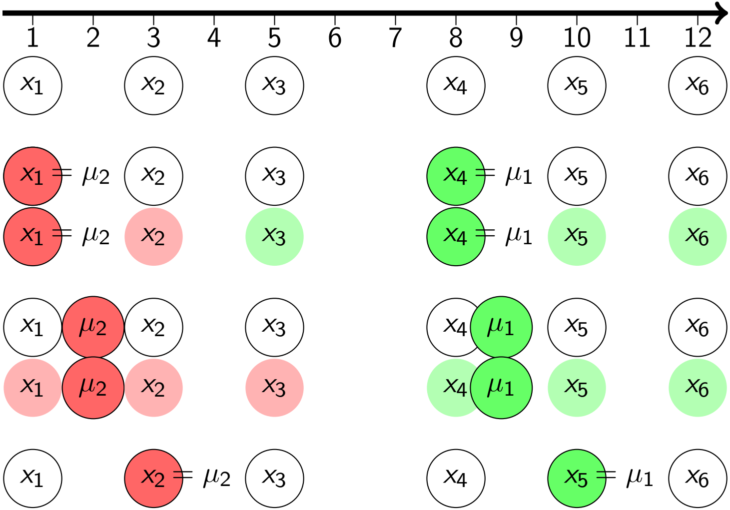
Example


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

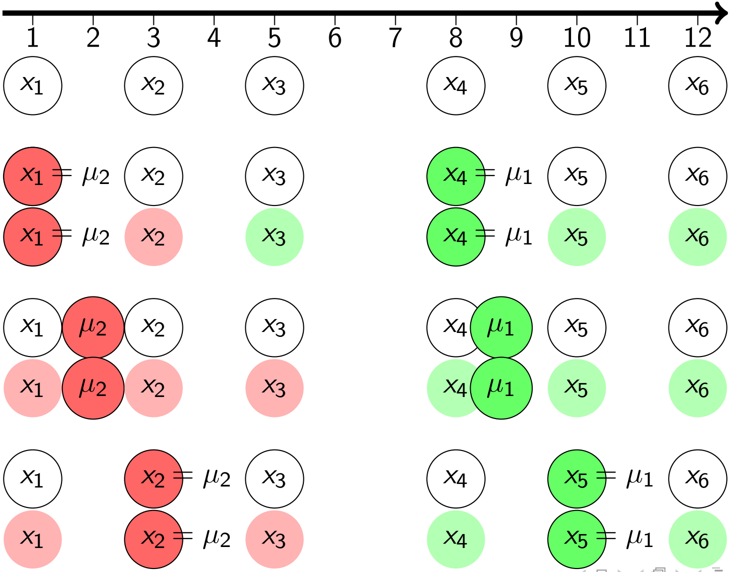
Example


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

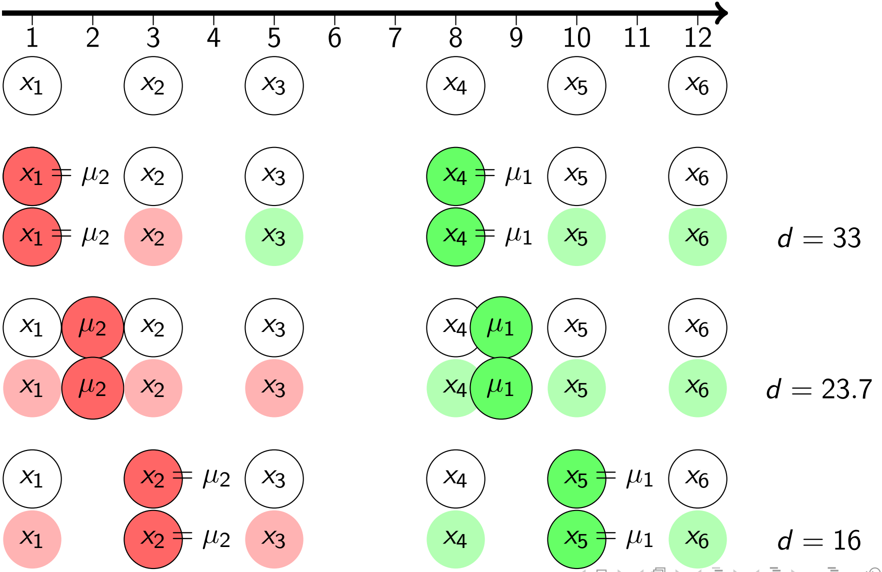
Example


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_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
x_1 $x_2 = \mu_2$ x_3 x_4 $x_5 = \mu_1$ 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
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

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
$d = 16$

K-medoids: K-means for General Distances

One can generalize k-means to general distances d:

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

K-medoids: K-means for General Distances

One can generalize k-means to general distances d:

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

- step 1 assigning data points to clusters remains the same

$$P_{i,k} := \underset{k \in \{1,\ldots,K\}}{\arg\min}\; d(x_i, \mu_k)$$

- 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:

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

- step 1 assigning data points to clusters remains the same

$$P_{i,k} := \arg\min_{k \in \{1,\ldots,K\}} d(x_i, \mu_k)$$

- 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:

$$\mu_k := x_j, \quad j := \arg\min_{j \in \{1,\ldots,n\}: P_{j,k}=1} \sum_{i=1}^{n} P_{i,k} d(x_i, x_j)$$

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

$P_{i,k}$ is called the **membership degree of instance i in class k** or the **cluster weight of instance i 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

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

  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,

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

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

  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, 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

For our data, no clusters are given, but this does not mean that they do not exist, there is just no way for us to measure them.

Mixture models assume that there exists an **unobserved nominal variable** Z with K levels, which is distributed according to:

$$Z \sim \text{Cat}(\pi)$$

or

$$p(Z = k) = \pi_k$$

for some probabilities π_k with

$$\sum_{k=1}^{K} \pi_k = 1$$

Mixture Models

Mixture models then model the joint probability of X and Z:

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

And the marginal probability of a given X is:

$$p(X) = \sum_{k=1}^{K} p(Z = k)p(X \mid Z = k) = \sum_{k=1}^{K} \pi_k p(X \mid Z = k)$$

All we need to specify is $p(X|Z)$!

Gaussian Mixture Models

Gaussian mixture models are mixture models where the probability of seeing an instance, given its cluster membership is a Gaussian:

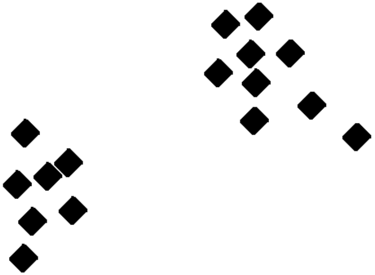
$$p(x_i|z_i = k) = \mathcal{N}(x_i; \mu_k, \Sigma_k)$$

Or equivalently:

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

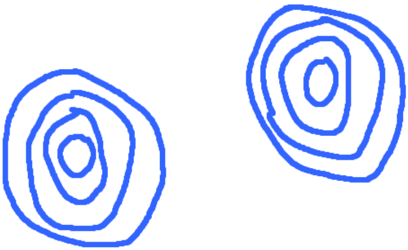
for a mean μ_k and Covariance Matrix Σ_k

Mixture Models: Intuition



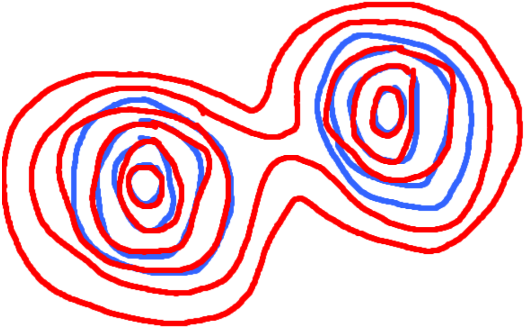
Clearly, we see that the hidden variable Z has two outcomes!

Mixture Models: Intuition



Data may come from a mixture of two Gaussians!

Mixture Models: Intuition



Heightlines of a mixture of two Gaussians!

Maximum Likelihood Estimate?

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

$$\ell(\Theta; X, Z) := \sum_{i=1}^{n} \sum_{k=1}^{K} \delta(Z_i = k)(\ln \pi_k + \ln p(X = x_i \mid Z = k; \theta_k))$$
$$\text{with } \Theta := (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K) \quad \theta_k = (\mu_k, \Sigma_k)$$

ℓ cannot be computed because z_i's are unobserved.

We cannot learn this model by computing a maximum likelihood estimate!

Expected Complete Likelihood & EM Algorithm

We calculate the **expected value of the loglikelihood** with respect to the conditional distribution of Z given X under the currently estimated θ^{t-1}:

$$Q(\Theta|\Theta^{t-1}) = \mathrm{E}_{Z|X,\Theta^{t-1}}[\ell(\Theta; X, Z)]$$

- From old Θ^{t-1}, we know the distribution of the Z, then compute the expectation value of ℓ with respect to Z (**Expectation Step**)
- We derive a quantity Q, that we can then maximize by optimizing Θ (**Maximization Step**)
- From the new Θ, we can then update Z and repeat the process

Expected Complete Likelihood

$$
\begin{aligned}
Q(\Theta|\Theta^{t-1}) &= \mathrm{E}\left[\sum_{i=1}^{N} \log p(x_i, z_i|\Theta)\right] \\
&= \sum_{i=1}^{N} \mathrm{E}\left[\log\left[\prod_{k=1}^{K} \pi_k p(x_i|\theta_k))^{\delta(z_i=k)}\right]\right] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} \mathrm{E}[\delta(z_i = k)] \log[\pi_k p(x_i|\theta_k] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p(z_i = k|x_i, \Theta^{t-1}) \log[\pi_k p(x_i|\theta_k] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log p(x_i|\theta_k)
\end{aligned}
$$

Expected Complete Likelihood (Expectation Step)

$$
\begin{aligned}
Q(\Theta|\Theta^{t-1}) &= \mathrm{E}\left[\sum_{i=1}^{N} \log p(x_i, z_i|\Theta)\right] \\
&= \dots \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log p(x_i|\theta_k)
\end{aligned}
$$

with

$$
r_{ik} = p(Z = k|x_i, \Theta) = \frac{\pi_k p(x_i|\theta_k)}{\sum_{k'} \pi_{k'} p(x_i|\theta_{k'})}
$$

which is called the **responsibilities** of a cluster k to an instance i

- Computing the r_{ik} yields the probabilites for Z (Expectation Step)

Maximization Step (I)

$Q(\Theta|\Theta^{t-1})$ needs to be maximized for all π_k and for the parameters of the individual Gaussians $\theta_k = (\mu_k, \Sigma_k)$.

For π, Q is maximized by setting

$$\pi_k = \frac{1}{N} \sum_i r_{ik} \quad \forall k$$

For μ_k and Σ_k, we only have to look at the second part of Q

$$\begin{aligned}\ell(\mu_k, \Sigma_k) &= \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log p(x_i|\Theta_k) \\ &= -\frac{1}{2} \sum_i r_{ik} [\log |\Sigma_k| + (x_i - \mu_k)^\top \Sigma_k (x_i - \mu_k)]\end{aligned}$$

Maximization Step (II)

$\ell(\mu_k, \Sigma_k)$ is maximized for:

$$\mu_k = \frac{\sum_{i=1}^{n} r_{i,k} x_i}{\sum_{i=1}^{k} r_{i,k}}$$

And

$$\Sigma_k = \frac{\sum_{i=1}^{n} r_{i,k} (x_i - \mu_k)^\top (x_i - \mu_k)}{\sum_{i=1}^{n} r_{i,k}}$$
$$= \frac{\sum_{i=1}^{n} r_{i,k} x_i^T x_i - \mu_k^\top \mu_k}{\sum_{i=1}^{n} r_{i,k}}$$

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:

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

- 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:

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

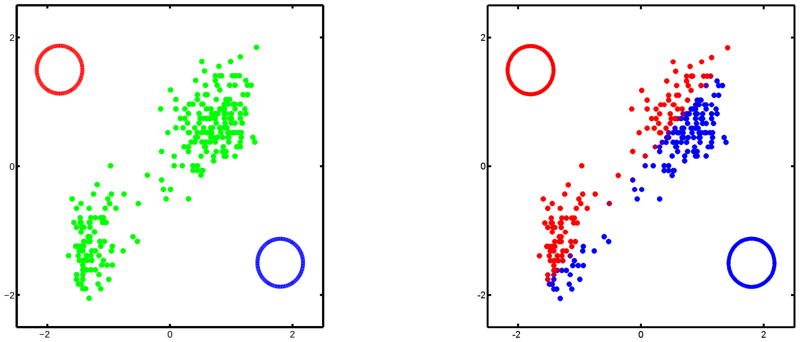
- To optimize D, we simply can run EM.

For hard clustering:

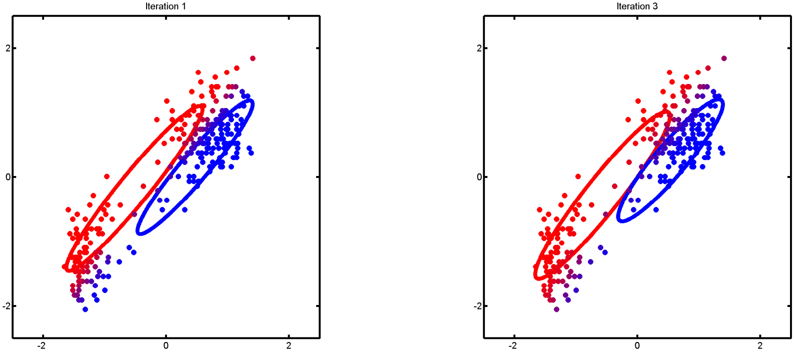
- assign points to the cluster with highest responsibility (**hard EM**):

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

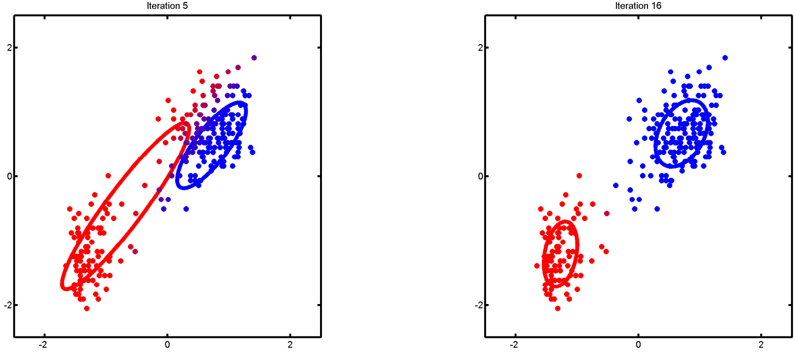
Gaussian Mixtures for Soft Clustering / Example



Gaussian Mixtures for Soft Clustering / Example



Gaussian Mixtures for Soft Clustering / Example



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.