AARHUS
UNIVERSITET

Optimization
and
Data Analytics

Alexandros Iosifidis
@
Aarhus University, Department of Engineering

Decision functions of the Normal Density

What are corrupted variables? We only draw some subset of the distribution

Given a set of corrupted variables x, or a function of x f(x), the expected value is calculated as

$$\mathcal{E}[f(x)] = \int_{-\infty}^{\infty} f(x)p(x)dx$$

Or in the case where the variables are discrete and belong to a set D

$$\mathcal{E}[f(x)] = \sum_{x \in D} f(x)P(x)$$

Decision functions of the Normal Density

The Normal (or Gaussian) Density of a continuous variable x is give by

$$p(x) = \frac{1}{\sqrt{2\pi}\,\sigma} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

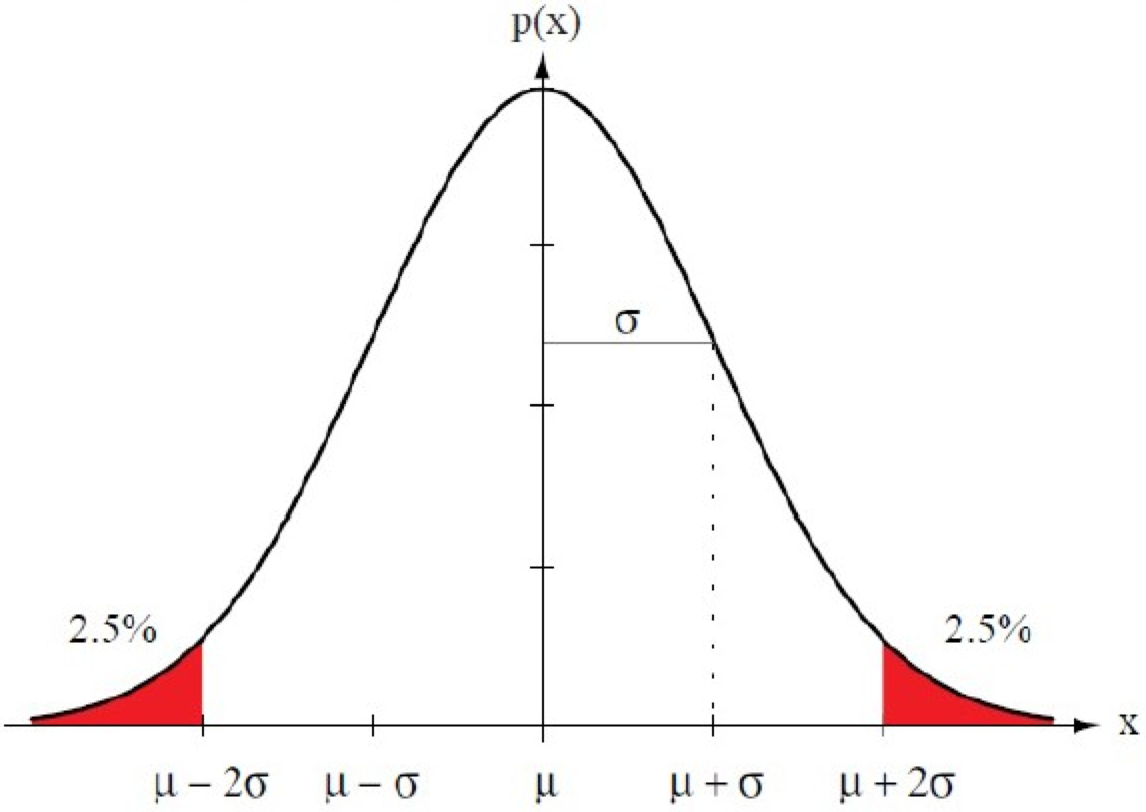
where μ and σ are the mean value and standard deviation

$$\mu = \mathcal{E}[x] = \int_{-\infty}^{\infty} x p(x) dx$$

$$\sigma^2 = \mathcal{E}[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx$$ the variance

Decision functions of the Normal Density

The Normal (or Gaussian) Density of a continuous variable x



Decision functions of the Normal Density

Using the two parameters μ and σ, the Normal Density (or Normal Distribution) is completely specified.

We use the notation $p(x) \sim N(\mu, \sigma^2)$ in order to denote that x follows a Normal Distribution centered as μ and having variance σ^2. ~ means that x is drawn from the distribution

Decision functions of the Normal Density

In the case where **x** is a D-dimensional vector following a multivariate Normal Distribution, we have

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

where

$$\boldsymbol{\mu} = \mathcal{E}[\mathbf{x}] = \int \mathbf{x}p(\mathbf{x})d\mathbf{x}$$

and can be calculated using 1-D operations

$$\mu_d = \mathcal{E}[x_d]$$

Decision functions of the Normal Density

In the case where **x** is a D-dimensional vector following a multivariate Normal Distribution, we have

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

where

$$\boldsymbol{\Sigma} = \mathcal{E}[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T] = \int (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T p(\mathbf{x}) d\mathbf{x}$$

and can be calculated using 1-D operations

$$\Sigma_{ij} = \mathcal{E}[(x_i-\mu_i)(x_j-\mu_j)]$$

Decision functions of the Normal Density

The covariance matrix **Σ** defines some important properties of the distribution

- **Σ**$_{ii}$ is the variance of dimension i
- **Σ**$_{ij}$ is the co-variance of dimensions i and j

1. If **Σ**$_{ij}$ = 0 for i ≠ j then dimensions i and j are statistically independent
2. If **Σ**$_{ij}$ = 0 for i ≠ j then the multivariate Normal Distribution degenerates to the product of D Normal Distributions
3. It can be used to define a distance function taking into account the different scaling of the various dimensions and their co-variances

This is called the Mahanolobis Distance:

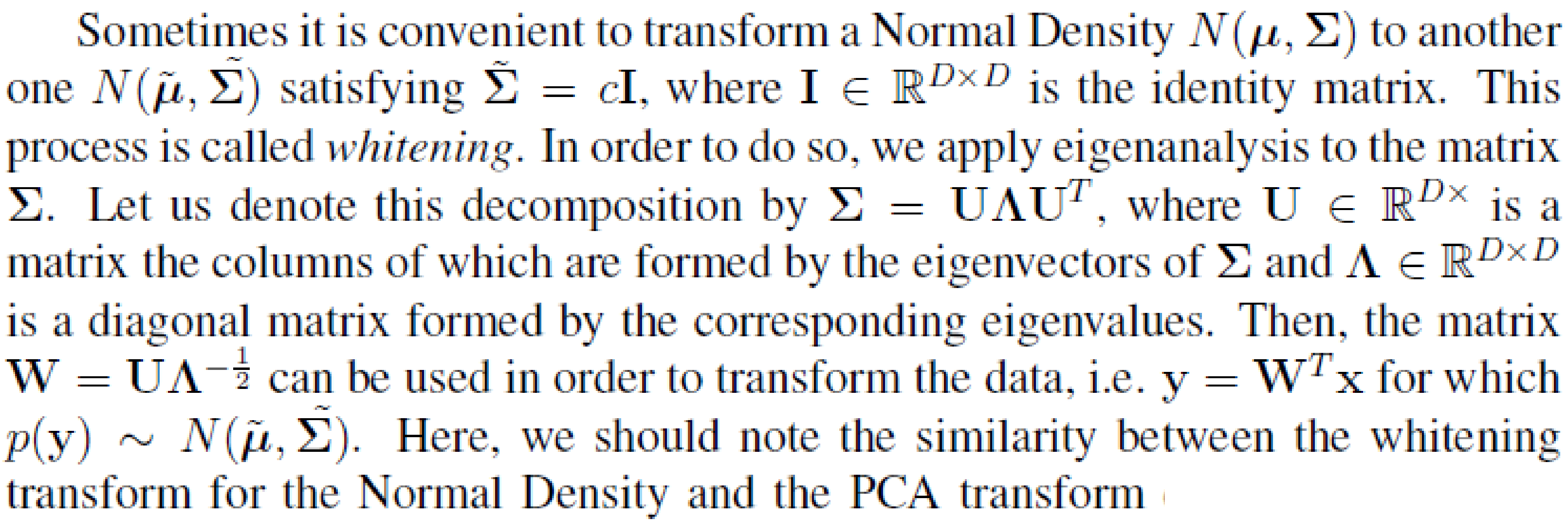
$$d_M(\mathbf{x}-\boldsymbol{\mu}) = (\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

Note that Sigma is Identity Matrix when using Euclidean Distance

In Euclidean Distance, we say that each dimension have equal weight. We assume that each dimensions are independent. With Sigma we can effect how much weight.

Decision functions of the Normal Density

Data whitening: We want a transformation x -> y such that the covariance matrix is the identity



Sometimes it is convenient to transform a Normal Density $N(\boldsymbol{\mu}, \Sigma)$ to another one $N(\tilde{\boldsymbol{\mu}}, \tilde{\Sigma})$ satisfying $\tilde{\Sigma} = c\mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{D \times D}$ is the identity matrix. This process is called *whitening*. In order to do so, we apply eigenanalysis to the matrix Σ. Let us denote this decomposition by $\Sigma = \mathbf{U}\Lambda\mathbf{U}^T$, where $\mathbf{U} \in \mathbb{R}^{D\times}$ is a matrix the columns of which are formed by the eigenvectors of Σ and $\Lambda \in \mathbb{R}^{D \times D}$ is a diagonal matrix formed by the corresponding eigenvalues. Then, the matrix $\mathbf{W} = \mathbf{U}\Lambda^{-\frac{1}{2}}$ can be used in order to transform the data, i.e. $\mathbf{y} = \mathbf{W}^T\mathbf{x}$ for which $p(\mathbf{y}) \sim N(\tilde{\boldsymbol{\mu}}, \tilde{\Sigma})$. Here, we should note the similarity between the whitening transform for the Normal Density and the PCA transform

Why do we do this?

1 . They are no correlaction between the dimensions because the non-diagonal are zero

2. The computation is much easier Sigma is the identity

After we find the Sigma tilde, then we can compute mu tilde by finding the mean of all the y's

Decision functions of the Normal Density

Consider a two-class classification problem, where each class follows a Normal Distribution

$$\begin{aligned} p(\mathbf{x}|c_1) &\sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ p(\mathbf{x}|c_2) &\sim N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \end{aligned}$$

Decision functions of the Normal Density

Consider a two-class classification problem, where each class follows a Normal Distribution

$$\begin{aligned} p(\mathbf{x}|c_1) &\sim N(\boldsymbol{\mu}_1, \Sigma_1) \\ p(\mathbf{x}|c_2) &\sim N(\boldsymbol{\mu}_2, \Sigma_2) \end{aligned}$$

Based on the Bayes' decision rule we will

$$\text{decide } c_1 \text{ if } \; P(c_1|\mathbf{x}) > P(c_2|\mathbf{x}), \; \text{ otherwise decide } c_2$$

Decision functions of the Normal Density

Consider a two-class classification problem, where each class follows a Normal Distribution

$$\begin{aligned} p(\mathbf{x}|c_1) &\sim N(\boldsymbol{\mu}_1, \Sigma_1) \\ p(\mathbf{x}|c_2) &\sim N(\boldsymbol{\mu}_2, \Sigma_2) \end{aligned}$$

Based on the Bayes' decision rule we will

$$\text{decide } c_1 \text{ if } P(c_1|\mathbf{x}) > P(c_2|\mathbf{x}), \text{ otherwise decide } c_2$$

Why divide with p(x)? because p(x) cancel each other

Replacing $P(c_k|\mathbf{x})$ from Bayes' formula we have

$$\text{decide } c_1 \text{ if } p(\mathbf{x}|c_1)P(c_1) > p(\mathbf{x}|c_2)P(c_2), \text{ otherwise decide } c_2$$

Decision functions of the Normal Density

What about monotonically decreasing function?

Note: if g(•) is a monotonic function then

The g() function cancel the exponential function. Therefore, it makes easier to make our computations.

$$P(c_1|\mathbf{x}) \; > \; P(c_2|\mathbf{x}) \quad \longrightarrow \quad g(P(c_1|\mathbf{x})) \; > \; g(P(c_2|\mathbf{x}))$$

We usually use the above in order to make our calculations easier.

Decision functions of the Normal Density

Note: if g(·) is a monotonic function then

$$P(c_1|\mathbf{x}) > P(c_2|\mathbf{x}) \quad \longrightarrow \quad g(P(c_1|\mathbf{x})) > g(P(c_2|\mathbf{x}))$$

We usually use the above in order to make our calculations easier.

Now our decision rule can be expressed as

$$\text{decide } c_1 \text{ if } \; g\left(p(\mathbf{x}|c_1)P(c_1)\right) > g\left(p(\mathbf{x}|c_2)P(c_2)\right), \;\; \text{otherwise decide } c_2$$

or

$$\text{decide } c_1 \text{ if } \; f(\mathbf{x}|c_1) > f(\mathbf{x}|c_2), \;\; \text{otherwise decide } c_2$$

Decision functions of the Normal Density

Remember that

$$\begin{aligned} p(\mathbf{x}|c_1) &\sim N(\boldsymbol{\mu}_1, \Sigma_1) \\ p(\mathbf{x}|c_2) &\sim N(\boldsymbol{\mu}_2, \Sigma_2) \end{aligned}$$

which is an exponential function. In order to simplify our computations, we use the function g(x) = ln(x)

determinant

Then we have f

$$f(\mathbf{x}|c_k) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) - \frac{D}{2}ln(2\pi) - \frac{1}{2}ln(|\Sigma_{\mathrm{k}}|) + ln(P(c_k))$$

Decision functions of the Normal Density

In the case where $\boldsymbol{\Sigma}_k = \sigma^2\mathbf{I}$, we have

- $|\boldsymbol{\Sigma}_k| = \sigma^{2D}$
- $\boldsymbol{\Sigma}_k^{-1} = (1/\sigma^2)\,\mathrm{I}$

which leads to

$$f(\mathbf{x}|c_k) = -\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_k)^T(\mathbf{x}-\boldsymbol{\mu}_k) - \frac{D}{2}ln(2\pi) - \frac{1}{2}ln(\sigma^{2D}) + ln(P(c_k))$$

and the decision rule becomes

$$-\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_1)^T(\mathbf{x}-\boldsymbol{\mu}_1) + ln(P(c_1)) > -\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_2)^T(\mathbf{x}-\boldsymbol{\mu}_2) + ln(P(c_2))$$

or

$$-\frac{1}{2\sigma^2}\|\mathbf{x}-\boldsymbol{\mu}_1\|_2^2 + ln(P(c_1)) > -\frac{1}{2\sigma^2}\|\mathbf{x}-\boldsymbol{\mu}_2\|_2^2 + ln(P(c_2)).$$

When P(c1) = P(c2) then we classify the x based on how close it is to the mean

Decision functions of the Normal Density

In the case where x is formed by discrete values

$$\int p(\mathbf{x}|c_k)d\mathbf{x} \rightarrow \sum_{\mathbf{x}} P(\mathbf{x}|c_k)$$

and Bayes' formula becomes

$$P(c_k|\mathbf{x}) = \frac{P(\mathbf{x}|c_k)P(c_k)}{P(\mathbf{x})}$$

where

$$P(\mathbf{x}) = \sum_{k=1}^{K} P(\mathbf{x}|c_k)P(c_k)$$

Maximum Likelihood Estimation

In practical Pattern Recognition problems, we rarely know the probabilities involved in the above described analysis. In such cases, the only information we have is a set of samples (each represented by a vector) and the corresponding labels (either class or group labels).

Maximum Likelihood Estimation

In practical Pattern Recognition problems, we rarely know the probabilities involved in the above described analysis. In such cases, the only information we have is a set of samples (each represented by a vector) and the corresponding labels (either class or group labels).

Given this information we need to estimate the values of the probabilities involved in our decision functions.

Maximum Likelihood Estimation

In practical Pattern Recognition problems, we rarely know the probabilities involved in the above described analysis. In such cases, the only information we have is a set of samples (each represented by a vector) and the corresponding labels (either class or group labels).

Given this information we need to estimate the values of the probabilities involved in our decision functions.

While in most cases the a priori probabilities $P(c_k)$ are relatively easy to be estimated, the conditional probabilities $p(\mathbf{x}|c_k)$ are generally difficult to be estimated.

because we only have a finite set of X's.

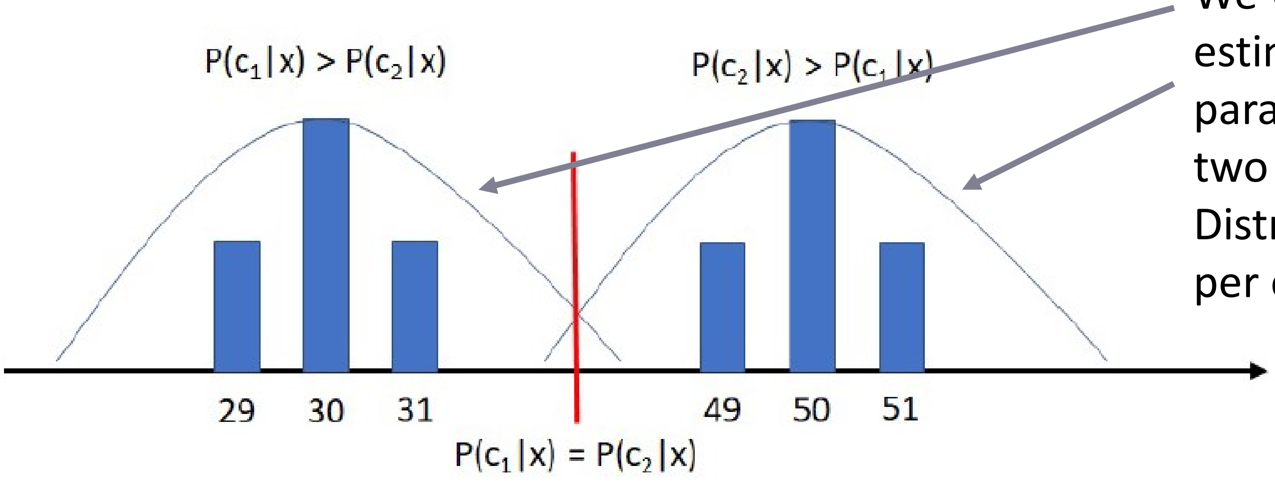
Maximum Likelihood Estimation

In order to make easier the estimation of the conditional probabilities $p(\mathbf{x}|c_k)$, we restrict our solution by setting assumptions on the distributions of the various parameters.

Maximum Likelihood Estimation

In order to make easier the estimation of the conditional probabilities p(**x**|c$_k$), we restrict our solution by setting assumptions on the distributions of the various parameters.

For example, we can assume that p(**x**|c$_k$) follows a Normal Distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$.



We want to estimate the parameters of the two Normal Distributions (one per each class)

Maximum Likelihood Estimation

In order to make easier the estimation of the conditional probabilities $p(\mathbf{x}|c_k)$, we restrict our solution by setting assumptions on the distributions of the various parameters.

For example, we can assume that $p(\mathbf{x}|c_k)$ follows a Normal Distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$.

We also assume that the distribution of each class is independent of those of the other classes.

Maximum Likelihood Estimation

In order to make easier the estimation of the conditional probabilities $p(\mathbf{x}|c_k)$, we restrict our solution by setting assumptions on the distributions of the various parameters.

For example, we can assume that $p(\mathbf{x}|c_k)$ follows a Normal Distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$.

We also assume that the distribution of each class is independent of those of the other classes.

Actually, once we have parameters of the normal distributions, we do not care about the actual dataset.

Then, we can include the parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ form the parameter set $\boldsymbol{\theta}_k$, which needs to be estimated using the data (which are assumed to be independent identically distributed random variables – iid).

Maximum Likelihood Estimation

Given i.i.d. data forming the set D, we have

If we assume that IID data then our computation becomes simpler since p(x|Theta) is

$$p(\mathcal{D}_k|\boldsymbol{\theta}_k) = \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta}_k)$$

p(D_k |$\boldsymbol{\theta}_k$) is the likelihood of $\boldsymbol{\theta}_k$ with respect to the samples $\mathbf{x}_i$ ϵ D_k. The Maximum Likelihood Estimation of $\boldsymbol{\theta}_k$ is the value $\boldsymbol{\theta}_k^*$ maximizing p(D_k |$\boldsymbol{\theta}_k$).

p(D | θ) can be read as the Probability of the Data given the Model parameters θ.

In the Maximum Likelihood Estimation, we want to find the set of model parameter values which gives the highest possible likelihood given the data.

Maximum Likelihood Estimation

Given i.i.d. data forming the set D, we have

$$p(\mathcal{D}_k|\boldsymbol{\theta}_k) = \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta}_k)$$

p(D$_k$ |**θ**$_k$) is the likelihood of **θ**$_k$ with respect to the samples **x**$_i$ ϵ D$_k$. The Maximum Likelihood Estimation of **θ**$_k$ is the value **θ**$_k^*$ maximizing p(D$_k$ |**θ**$_k$).

We use the following function for determining **θ**$_k^*$

$$L(\boldsymbol{\theta}_k) = ln(p(\mathcal{D}_k|\boldsymbol{\theta}_k)) = \sum_{i=1}^{N} ln(p(\mathbf{x}_i|\boldsymbol{\theta}_k))$$

we the property of ln()

Maximum Likelihood Estimation

Then, **θ**$_k$* can be obtained by optimizing for

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\arg\max}\, L(\boldsymbol{\theta}_k)$$

or

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\arg\max} \sum_{i=1}^{N} ln(p(\mathbf{x}_i|\boldsymbol{\theta}_k))$$

Maximum Likelihood Estimation

Then, $\boldsymbol{\theta}_k^*$ can be obtained by optimizing for

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\arg\max}\, L(\boldsymbol{\theta}_k)$$

or

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\arg\max} \sum_{i=1}^{N} ln(p(\mathbf{x}_i|\boldsymbol{\theta}_k))$$

The above problem can be optimized using

Closed-form solution, the parameters
on one side and the variables on the other side

$$\nabla_{\boldsymbol{\theta}_k} L(\boldsymbol{\theta}_k) = 0$$

Maximum Likelihood Estimation

Example:

As an example, let us assume that the samples of the sub-problem $\mathcal{D}_k$ follow a Normal Density with mean value μ_k and variance σ_k^2, which are unknown. The set of parameters to be estimated is now $\boldsymbol{\theta}_k = [\mu_k \; \sigma_k^2]^T$. For one sample x_i we have:

$$L(\boldsymbol{\theta}_k) = ln(p(x_i|\theta_k)) = -\frac{1}{2}ln(2\pi\theta_{k,2}) - \frac{1}{2\theta_{k,2}}(x_i - \theta_{k,1})^2,$$

where we set $\theta_{k,1} = \mu_k$ and $\theta_{k,2} = \sigma_k^2$. The partial derivatives with respect to $\theta_{k,1}$ and $\theta_{k,2}$ are:

$$\frac{\vartheta L(\boldsymbol{\theta}_k)}{\vartheta \theta_{k,1}} = \frac{1}{\theta_{k,2}}(x_i - \theta_{k,1})$$

and

$$\frac{\vartheta L(\boldsymbol{\theta}_k)}{\vartheta \theta_{k,2}} = -\frac{1}{2\theta_{k,2}} + \frac{(x_i - \theta_{k,1})^2}{2\theta_{k,2}^2}$$

Maximum Likelihood Estimation

Example:

By aggregating for all samples and setting equal to zero, we obtain:

$$\sum_{i=1}^{N} \frac{1}{\theta_{k,2}^{*}}(x_i - \theta_{k,1}^{*}) = 0$$

and

$$\sum_{i=1}^{N} \left(\frac{(x_i - \theta_{k,1}^{*})^2}{\theta_{k,2}^{*2}} - \frac{1}{\theta_{k,2}^{*}} \right) = 0,$$

leading to:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

and

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2.$$

Expectation Maximization

Sometimes, in order to solve a problem in which there is more than one parameters we need to apply an alternating iterative optimization process.

Expectation Maximization

Sometimes, in order to solve a problem in which there is more than one parameters we need to apply an alternating iterative optimization process.

At each step of this process, we fix all parameters except one, and optimize a reduced criterion in order to find a better value for the overall optimization problem.

Expectation Maximization

Sometimes, in order to solve a problem in which there is more than one parameters we need to apply an alternating iterative optimization process.

At each step of this process, we fix all parameters except one, and optimize a reduced criterion in order to find a better value for the overall optimization problem.

This process is repeated multiple times, on each of which a different parameter is optimized and ends when no change is observed on the overall optimization criterion.

Expectation Maximization

Sometimes, in order to solve a problem in which there is more than one parameters we need to apply an alternating iterative optimization process.

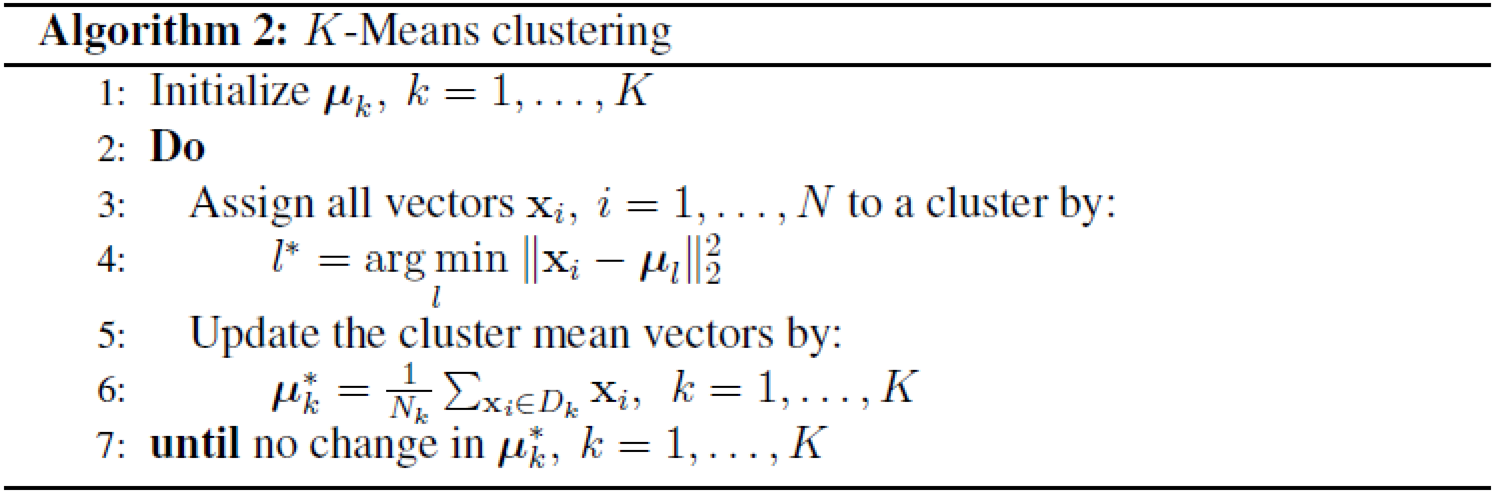
At each step of this process, we fix all parameters except one, and optimize a reduced criterion in order to find a better value for the overall optimization problem.

This process is repeated multiple times, on each of which a different parameter is optimized and ends when no change is observed on the overall optimization criterion.

The above described process, can be sometimes described by using probabilistic terms and is called Expectation-Maximization (EM).

Expectation Maximization

Example: K-Means algorithm



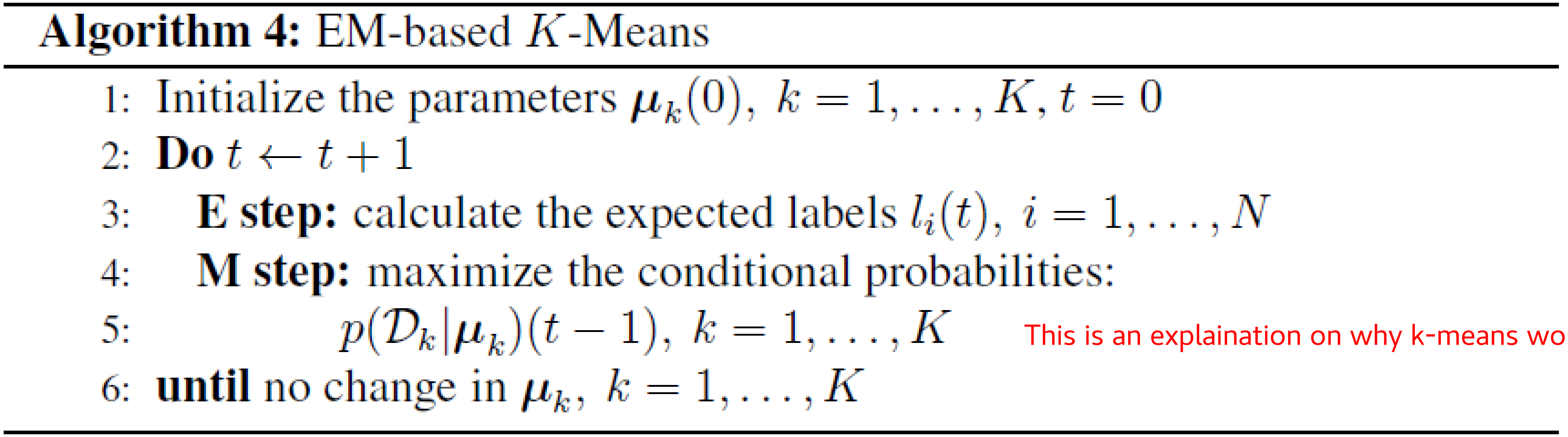
```
Algorithm 2: $K$-Means clustering
Initialize $\boldsymbol{\mu}_k,\ k = 1, \dots, K$
Do
    Assign all vectors $\mathbf{x}_i,\ i = 1, \dots, N$ to a cluster by:
        $l^* = \arg\min_l \|\mathbf{x}_i - \boldsymbol{\mu}_l\|_2^2$
    Update the cluster mean vectors by:
        $\boldsymbol{\mu}_k^* = \frac{1}{N_k} \sum_{\mathbf{x}_i \in D_k} \mathbf{x}_i,\ \ k = 1, \dots, K$
until no change in $\boldsymbol{\mu}_k^*,\ k = 1, \dots, K$
```

How can we formulate the above algorithm as an EM process?

Expectation Maximization

Example: K-Means algorithm

p(Dk | uk): notice that when we maximise the conditional probabilities, we get the mean vector just as we computed in slide 30



Algorithm 4: EM-based K-Means

```
Initialize the parameters $\boldsymbol{\mu}_k(0),\ k = 1, \ldots, K, t = 0$
Do $t \leftarrow t + 1$
    E step: calculate the expected labels $l_i(t),\ i = 1, \ldots, N$
    M step: maximize the conditional probabilities:
            $p(\mathcal{D}_k | \boldsymbol{\mu}_k)(t-1),\ k = 1, \ldots, K$
until no change in $\boldsymbol{\mu}_k,\ k = 1, \ldots, K$
```

This is an explaination on why k-means works!