Christian-Albrechts-Universität zu Kiel

Lecture
“Intelligent Systems“

Chapter 9: Generative Modelling

Prof. Dr.-Ing. habil. Sven Tomforde / Intelligent Systems

Winter term 2020/21

Agenda

Christian-Albrechts-Universität zu Kiel

Content

- Generative Modelling (GM)
- Gaussian Mixture Models (GMM)
- Training Gaussian Mixture Models
- Applications in Intelligent systems
- Conclusion and further readings

Goals

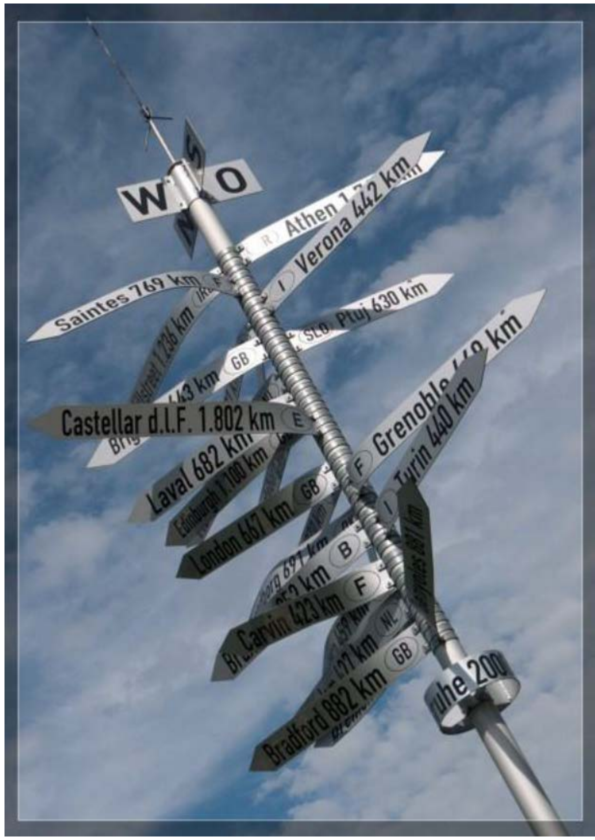
Students should be able to:

- Explain how generative modelling works
- Understand the concept of Gaussian Mixture Models and define all contained parts
- Describe how a GMM is trained using k-means and Expectation Maximisation
- Explain for which tasks GMMs can be used in OC systems

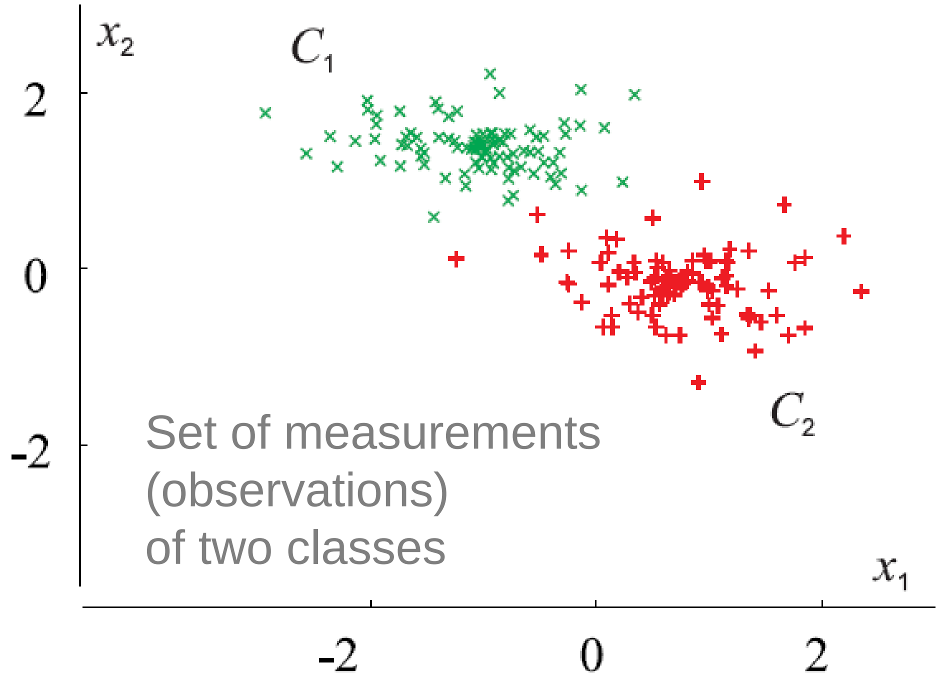
Agenda

Christian-Albrechts-Universität zu Kiel

- Generative Modelling
- Gaussian Mixture Models
- Training Gaussian Mixture Models
- Applications in intelligent systems
- Conclusion and further readings



Example: Classification problem

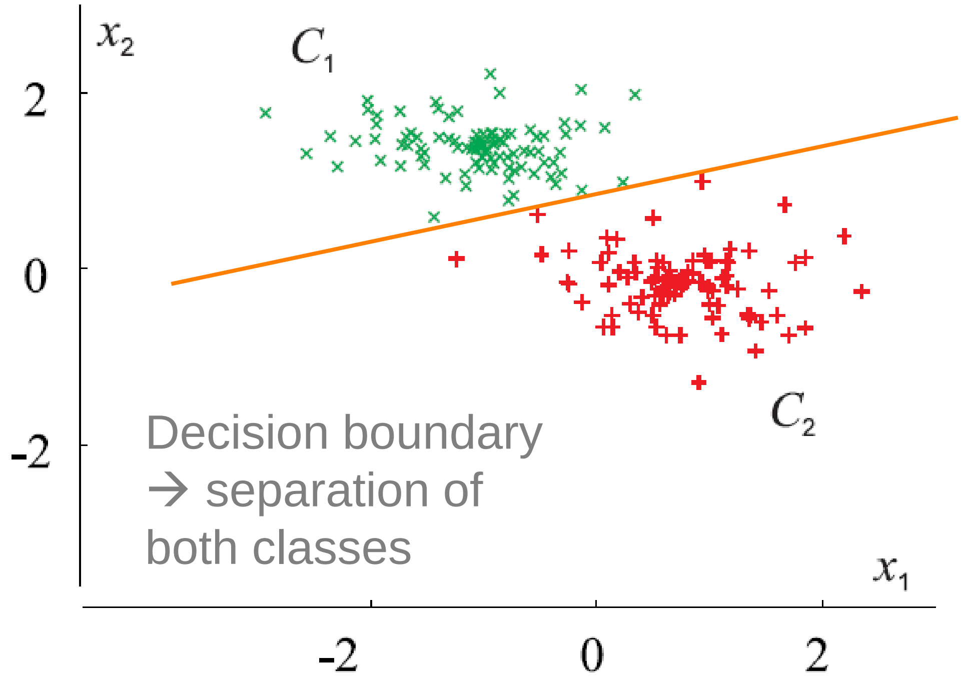


Set of measurements (observations) of two classes

Generative Modelling (2)

Christian-Albrechts-Universität zu Kiel

Example: Classification problem



Decision boundary
→ separation of
both classes

Generative Modelling (3)

- Problem:
  - In most cases, observations are not linearly separable!

- Solution:
  - Use Gaussian functions to transform observation non-linearly.
  - Separate the transformed observations linearly (i.e. with a hypersphere).

Generative Modelling (4)

Christian-Albrechts-Universität zu Kiel

Decision y about class assignment of x (by comparing y_1 and y_2)

Linear combination with parameter vectors:

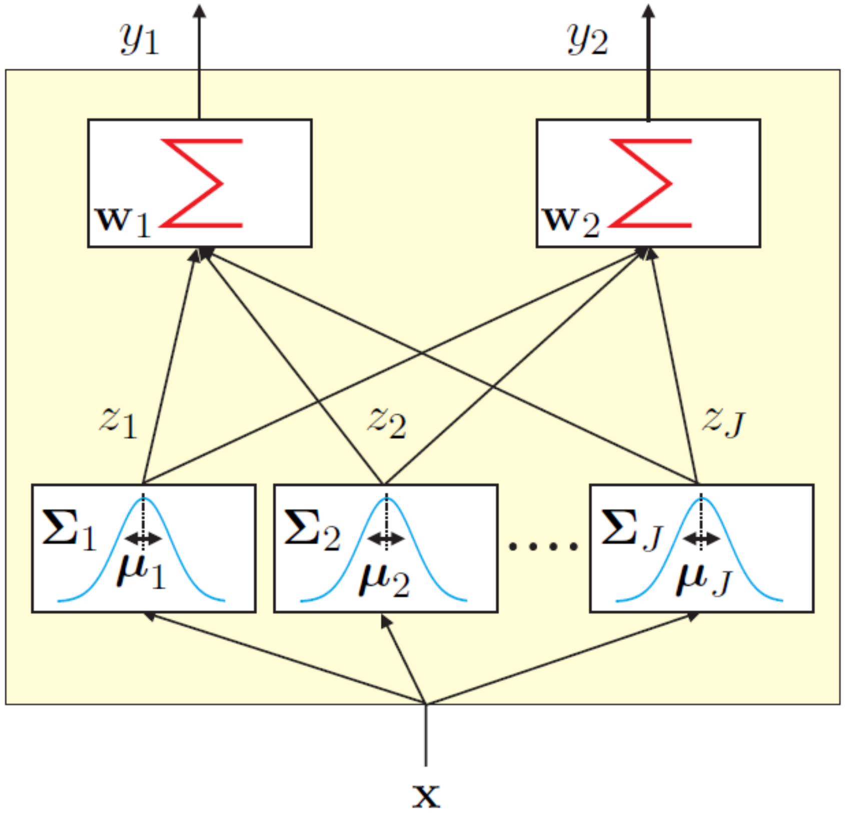
$$w_1 = (w_{01}, w_{11}, \dots, w_{K1})^T$$
$$w_2 = (w_{02}, w_{12}, \dots, w_{K2})^T$$

Scalar output z_k

Several multi-variate Gaussian functions with parameters (centre μ_k and covariance matrix Σ_k)

Observation x

Classifier for 2 classes (y)


y_1
y_2
$\mathbf{w}_1$ $\sum$
$\mathbf{w}_2$ $\sum$
z_1
z_2
z_J
Σ_1 $\boldsymbol{\mu}_1$
Σ_2 $\boldsymbol{\mu}_2$
....
Σ_J $\boldsymbol{\mu}_J$
$\mathbf{x}$

Generative Modelling (5)

Types of classifiers

- Discriminative classifiers:
  Use a discrimination function, which maps an observation to a class (or action).
- Comprehensible classifiers:
  Are constructed and interpreted by human experts.
- Generative classifiers:
  Model those processes that caused the observations (i.e. that generated the observations).

Generative Modelling (6)

Focus: Generative classifiers, i.e. probabilistic ones

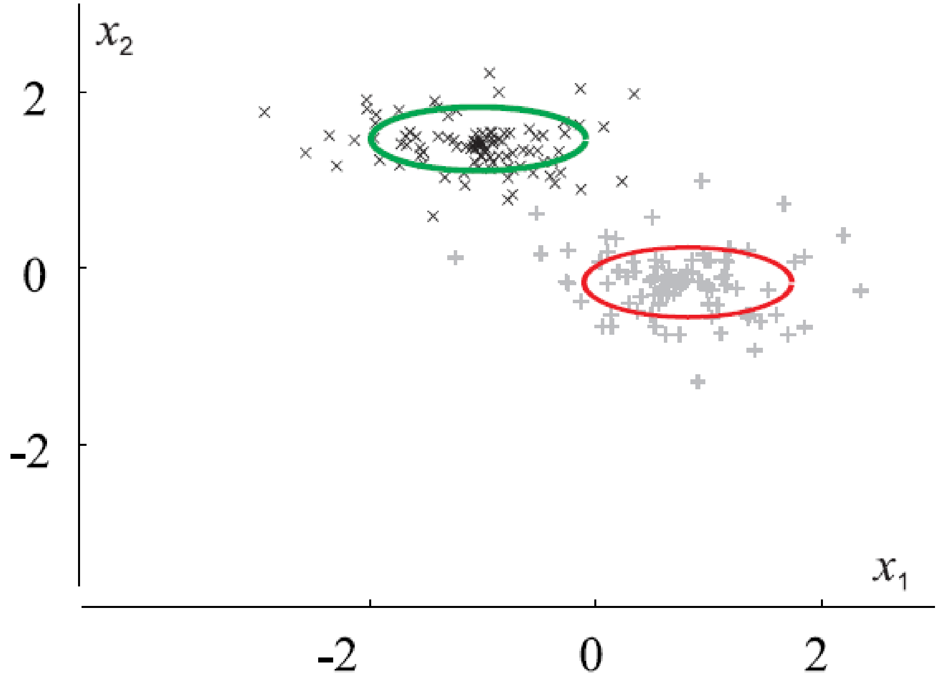
Advantages:

- Easy to combine with cost or utility functions
  $\rightarrow$ Minimise risk of false alarms
- Easy to define a rejection criterion
  $\rightarrow$ Classifier can reject a decision if it is too uncertain
- Detection of outliers in the data or within the observed environment
- Depending on the choice of the distributions, symbolic rules (readable by humans) can be extracted
- …

Generative Modelling (7)

Christian-Albrechts-Universität zu Kiel

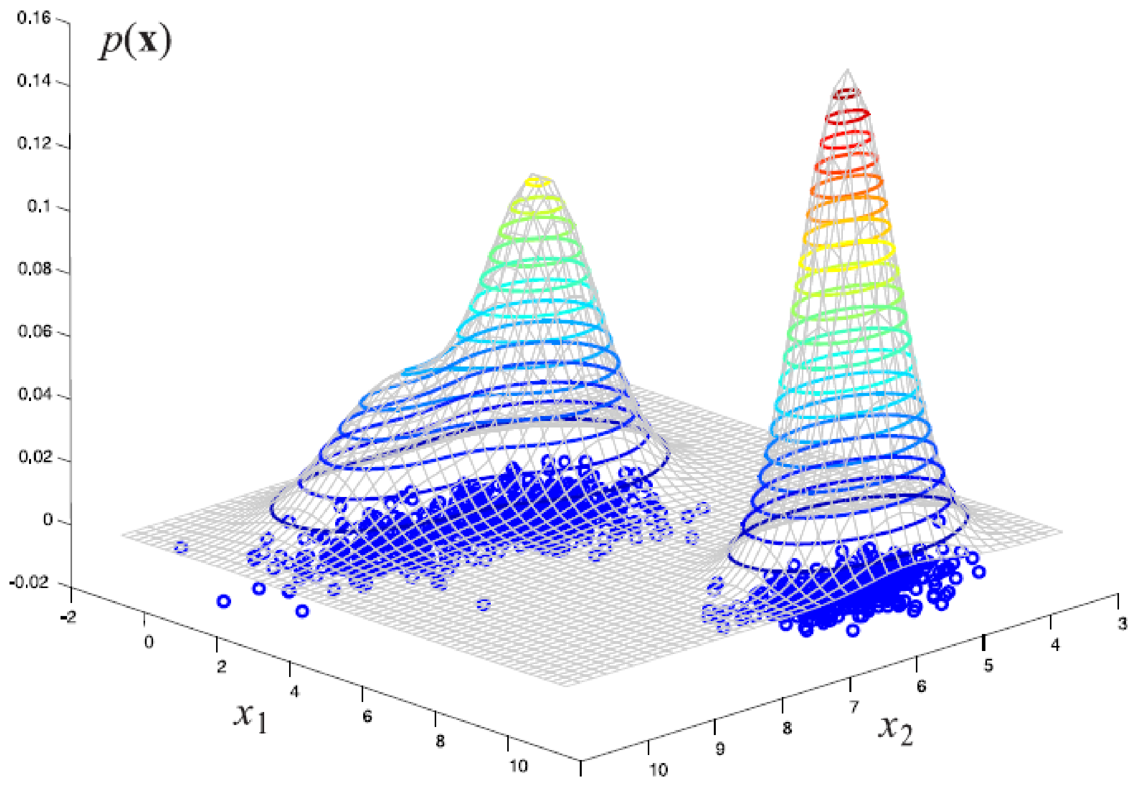
Example



Idea:
Use multi-variate Gaussian functions to model cluster in the data

- Ellipses define contour lines → I.e. similar to lines indicating the same height of mountains to depict descent of areas.

Generative Modelling (8)



- Linear combination of several Gaussian functions (here: 3)
  → called: Gaussian Mixture Model

Generative Modelling (9)

The Gaussian distribution

- The Gaussian distribution (or normal/standard distribution) is a well-known model for the distribution of continuous variables.
- With real-valued random variables x holds:

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

  with mean (average) μ and variance σ^2.
- A multi-variate Gaussian distribution over a D-dimensional vector x containing continuous variables is defined by:

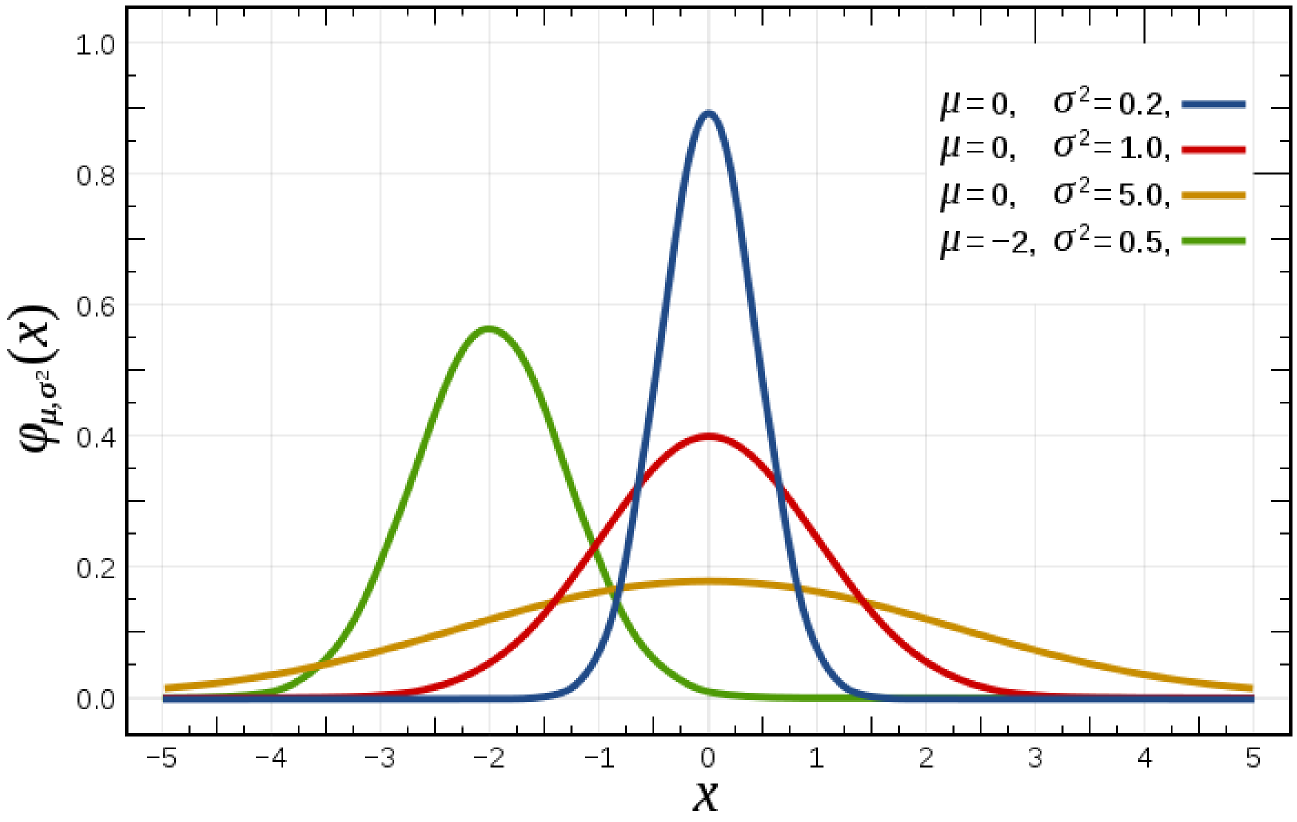
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- The D-dimensional vector μ is called mean, the $D \times D$ matrix Σ covariance matrix, and $|\Sigma|$ defines the determinate of Σ.

Generative Modelling (10)

Christian-Albrechts-Universität zu Kiel

The Gaussian distribution



Generative Modelling (11)

The Gaussian distribution

- The (multi-variate) Gaussian distribution depends on x using the so-called quadratic form:

$$\Delta^2 = (x - \mu)^T \Sigma^{-1} (x - \mu)$$

(see exponent)

- Δ is called Mahalanobis distance of μ and x. If Σ is the identity matrix (or unit matrix), the Mahalanobis distance is equal to the Euclidian distance.

Generative Modelling (12)

Christian-Albrechts-Universität zu Kiel

Properties of the Gaussian distribution

- Although the multivariate normal distribution is commonly used as a model for probability densities, the application is limited.
- A symmetric covariance matrix Σ has $D(D+1)/2$ parameters.
- Together with the D parameters in μ this sums up to $D(D+3)/2$ parameters. Consequently, the number of parameters grows quadratic with D (problematic, e.g. when inverting a matrix).
- Thereby, D describes the number of dimensions of the input space.
- Solution: Restrictions regarding the covariance matrix

Generative Modelling (13)

Christian-Albrechts-Universität zu Kiel

Possible restrictions

- Diagonal covariance matrices
  - I.e. $\Sigma = \mathrm{diag}(\sigma_d^2)$
  - Have $2D$ parameters
  - The corresponding surfaces of constant density (contour lines) are axis-oriented.
- Covariance matrices proportional to the unit/identity matrix:
  - I.e. $\Sigma = \sigma\, I$ (so-called isotropic covariance)
  - Have only $D + 1$ parameters
  - The corresponding surfaces of constant density (contour lines) are hyperspheres.

Drawback: Limitation of the modelling power.

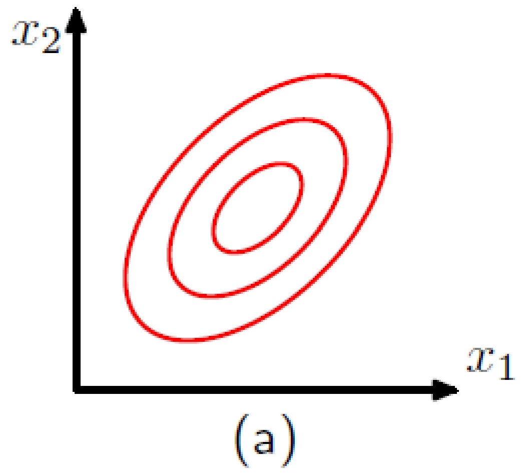
Generative Modelling (14)

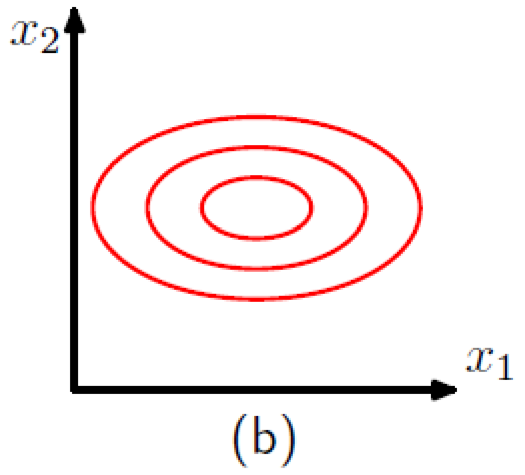
Examples for surfaces with a constant density (“contour lines“)

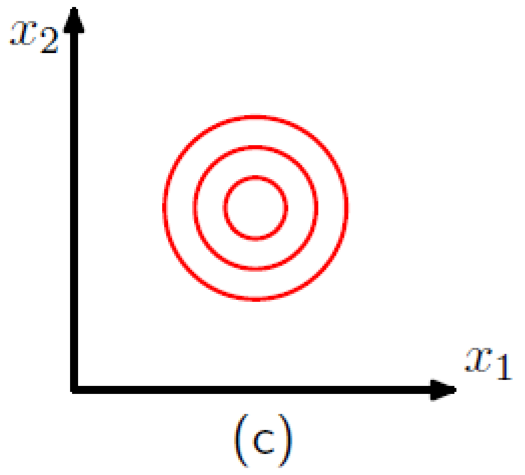
a) Arbitrary covariance matrices

b) Diagonal covariance matrices

c) Covariance matrices proportional to the identity/unit matrix







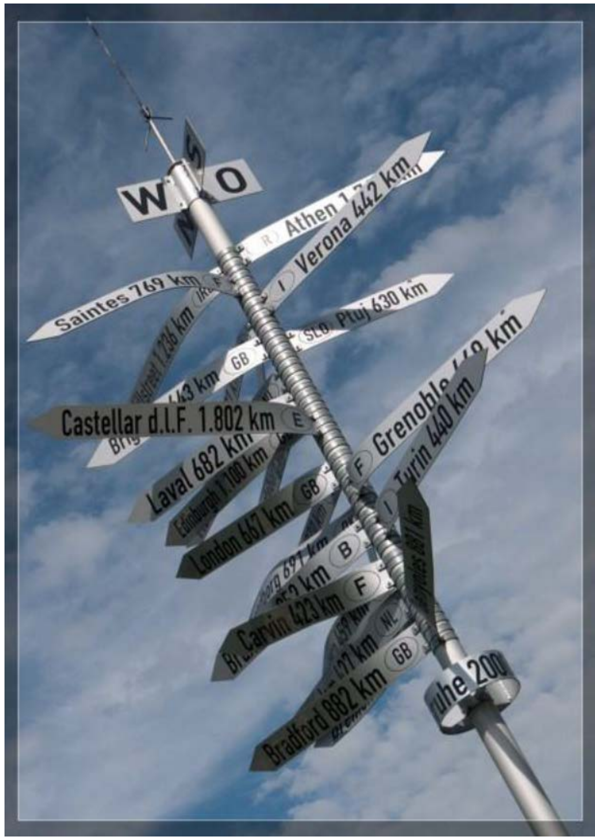
Generative Modelling (15)

Further limitations for using normal/standard distributions

- The Gaussian distribution is uni-modal, i.e. it comes with one isolated maximum.
- It cannot be used to approximate multi-modal distributions.
- One the one hand, the Gaussian distribution is highly flexible in terms of a variety of parameters. On the other hand, it is not flexible enough in terms of the set of distributions that can be represented.

Agenda

- Generative Modelling
- Gaussian Mixture Models
- Training Gaussian Mixture Models
- Applications in intelligent systems
- Conclusion and further readings



Gaussian Mixture Models

Observation

- Modelling of real-world data sets
  reveals drawbacks of Gaussian distributions

Example:

- Old Faithful

Source: M. R. Marty

Gaussian Mixture Models (2)

Old Faithful / Yellowstone National Park

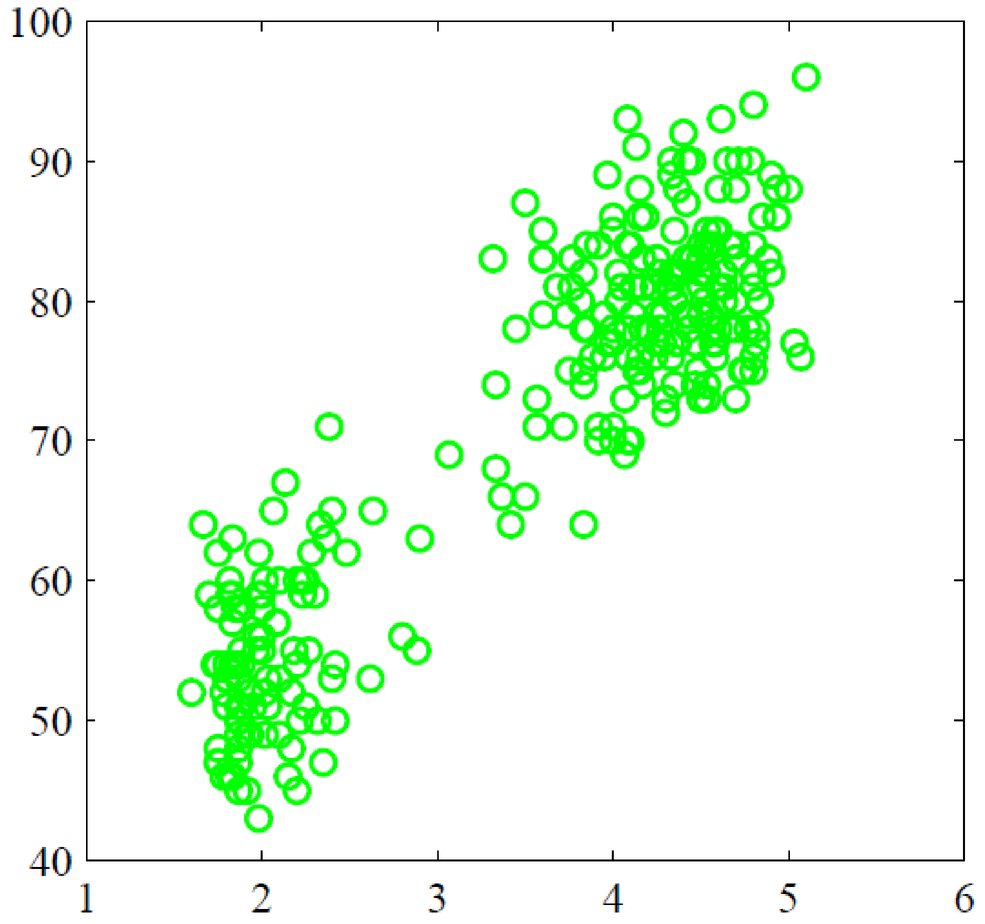
- One of the most prominent geysers
- Name given by members of the Washburn-Langford-Doane expedition since it erupts with high frequency and reliability.
- Belongs to the class of jet-geysers (comes with a water column of small diameter). Diameter is about 11 cm at 7m under surface.
- Water column of about 30 to 55 meters into the air, resolves 14,000 to 32,000 litres of water within one eruption of 1.5 to 5 min duration.
- We consider a data set containing 272 observations of Old Faithful (i.e. 272 individual eruptions) with two variables: duration of eruption and time until next eruption.

Gaussian Mixture Models (3)

Old Faithful

- Duration of an eruption (horizontal axis)
- Time until next eruption (vertical axis)

→ We can identify two clusters of observations!

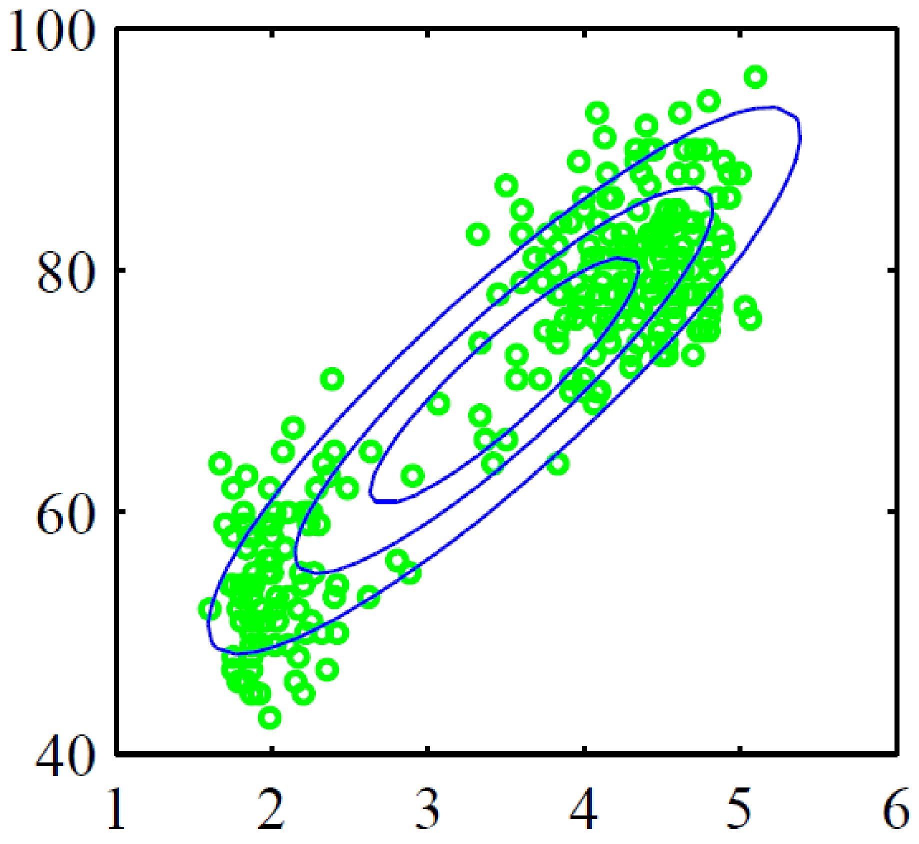


Gaussian Mixture Models (4)

Old Faithful

- A first approach: Modelling the behaviour using a single Gaussian distribution.

→ Bad representation….

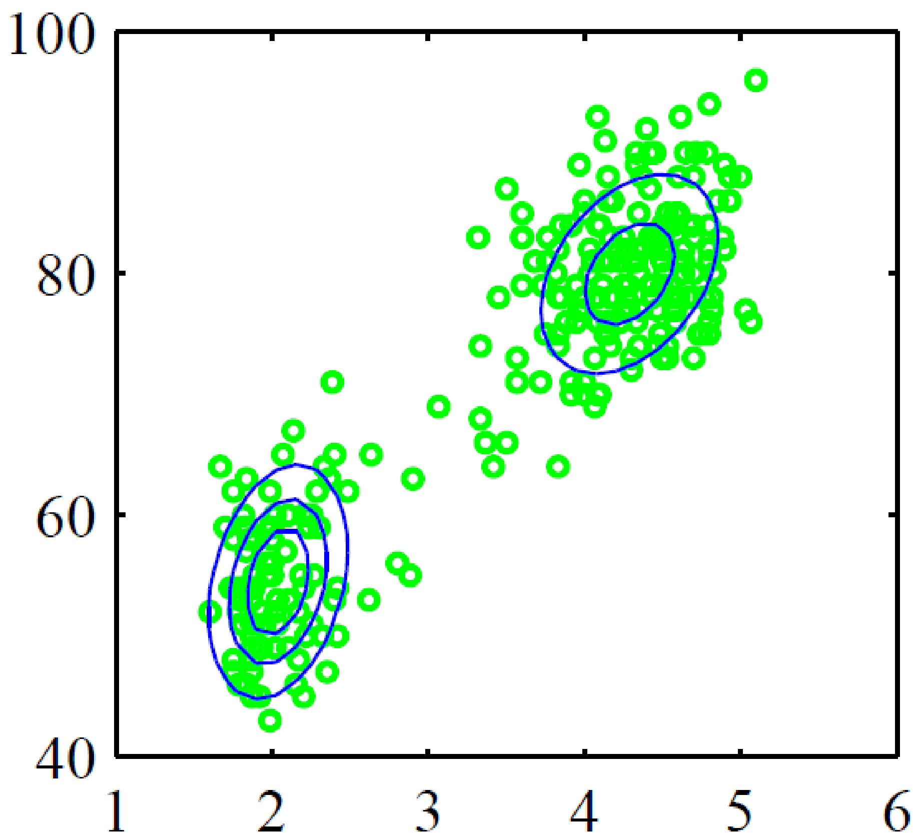


Gaussian Mixture Models (5)

Old Faithful

- A second approach: Modelling the behaviour using a linear combination of two Gaussian distributions.

→ Looks much better ….



Gaussian Mixture Models (6)

Mixture Models

- Linear combination of several basic distributions, e.g. Gaussians.

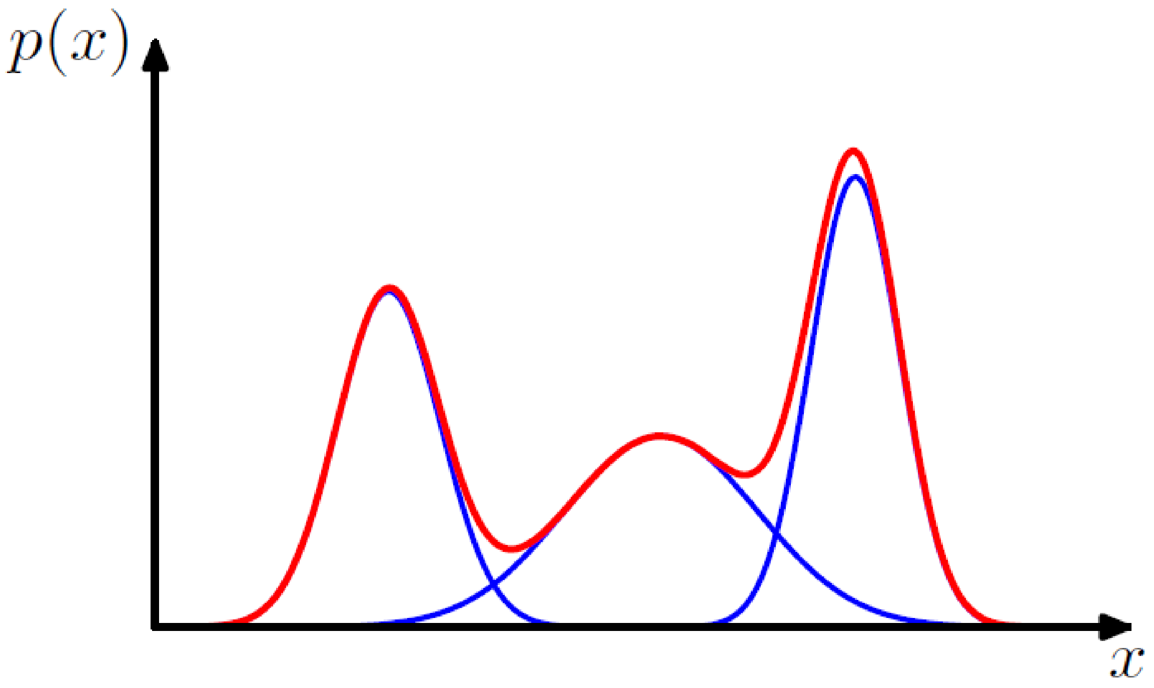
- With an appropriate number of Gaussian functions and their appropriately configured parameters (mean and covariance matrix) as well as an appropriate definition of the weights for the linear combination, we can approximate almost every continuous density function with arbitrary accuracy.

Gaussian Mixture Models (7)

Christian-Albrechts-Universität zu Kiel

Mixture Models

- Example: Gaussian Mixture Model (in red) as a linear combination of three different Gaussian functions (in blue).



Gaussian Mixture Models (8)

Christian-Albrechts-Universität zu Kiel

Mixture Models

- Linear combination of K Gaussian distributions:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Each Gaussian distribution $\mathcal{N}(x|\mu_k, \Sigma_k)$ is called a component of the mixture model.
- Each component has its own mean μ_k and its own covariance matrix Σ_k.
- The parameters π_k are called mixture coefficients.

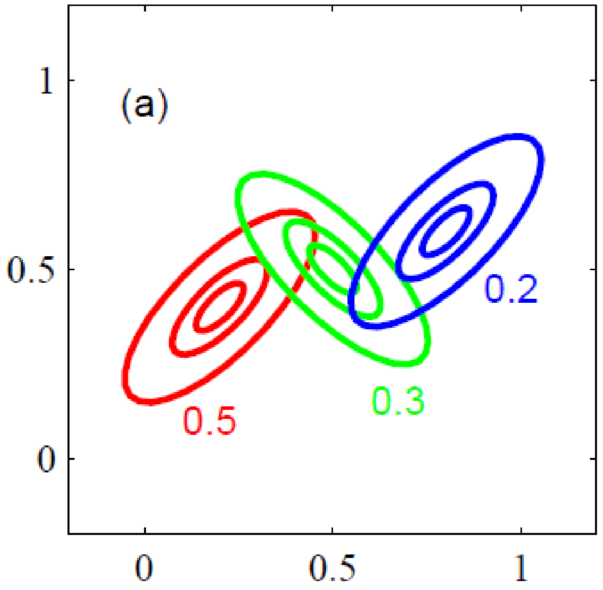
Gaussian Mixture Models (9)

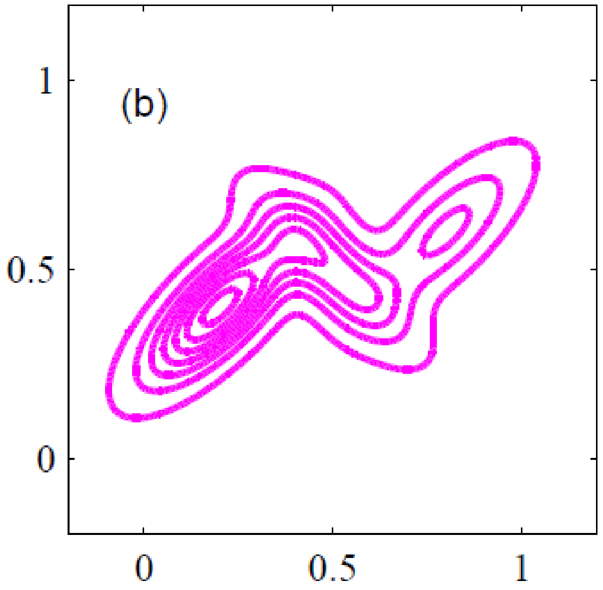
Example: Gaussian Mixture Model

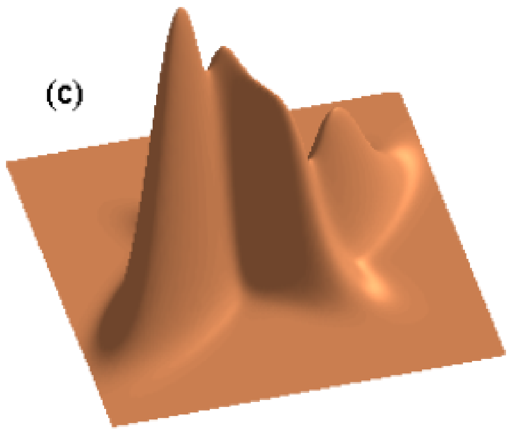
a) Contour lines of the constant density of the three components (or: values of mixture coefficients).

b) Contour lines of the constant density of the mixture model

c) 3-D plot of the mixture model







Gaussian Mixture Models (10)

Christian-Albrechts-Universität zu Kiel

Gaussian Mixture Model (GMM)

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Integrating over both sides while considering that both, components and mixture model are normalised results in:

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

- From $p(x) \geq 0$ and $\mathcal{N}(x|\mu_k, \Sigma_k) \geq 0$ follows that $\pi_k \geq 0$ for all k.

- As a conclusion, this means: $0 \leq \pi_k \leq 1$

Gaussian Mixture Models (11)

GMM and probabilities

- The mixture coefficients fulfil all requirements for probabilities. In consequence, we can write (with $\pi_k = p(k)$):

$$p(x) = \sum_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)p(k)$$

  with $p(x)$ as a priori probability for choosing the k-th component.

- Later on, we'll see that a posteriori probabilities $p(k|x)$ will play a major role (they are called responsibilities).

Gaussian Mixture Models (12)

Responsibilities

- Using Bayes‘ theorem, we can express the responsibilities $\gamma_k(x)$ as:

$$\begin{aligned}\gamma_k(x) &\equiv p(k|x) \\ &= \frac{p(k)p(x|k)}{p(x)} \\ &= \frac{p(k)p(x|k)}{\sum_l p(l)p(x|l)} \\ &= \frac{\pi_l \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_l \pi_l \mathcal{N}(x|\mu_l, \Sigma_l)}\end{aligned}$$

Gaussian Mixture Models (13)

GMM

- Consequently, the shape of the Gaussian Mixture Model is controlled by the parameters π, μ, and Σ, with $\pi \equiv \{\pi_1, \dots, \pi_K\}$, $\mu \equiv \{\mu_1, \dots, \mu_K\}$, and $\Sigma \equiv \{\Sigma_1, \dots, \Sigma_K\}$.
- One possibility to configure these parameters is using a Maximum-Likelihood (ML) approach. This requires the so-called Log-Likelihood-Function for given data $X = \{x_1, \dots, x_N\}$

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}$$

- For a given data set, the ML-based solution is the model with the highest probability (i.e. highest probability of having generated these data).

Gaussian Mixture Models (14)

GMM

- In general, this function has to be maximised by derivation according to different parameters, determining zeros, and proving that this is a maximum.
- Well, within the argument of the logarithm there is a summation over k.
  → Makes it a bit complicated…
- Means: The ML approach cannot be solved analytically in a closed-form.
- However: There are approaches to tackle this problem
  → e.g. Expectation Maximisation

Gaussian Mixture Models (15)

Christian-Albrechts-Universität zu Kiel

Design of a generative classifier

- Notation: $c = 1, \dots, C$ are classes, $k = 1, \dots, K$ are components, x is the input variable (or vector), R_c is the region of the input space associated with class c.
- Sought: $p(c|x)$, i.e. the distribution of classes for given observations

$$
\begin{aligned}
p(c|x) &= \frac{p(x|c)p(c)}{p(x)} \\
&= \frac{\left(\sum_{k=1}^{K} p(x|k)p(k|c)\right)p(c)}{\sum_{k=1}^{K} p(x|k)p(k)} \\
&= \sum_{k=1}^{K} \frac{p(k|c)p(c)}{p(k)} \cdot \frac{p(x|k)p(k)}{\sum_{k=1}^{K} p(x|k)p(k)} \\
&= \sum_{k=1}^{K} \frac{\int_{x \in R_c} p(k|x)dx \cdot p(c)}{p(k)} \cdot \frac{p(x|k)p(k)}{\sum_{k=1}^{K} p(x|k)p(k)}
\end{aligned}
$$

Gaussian Mixture Models (16)

Design of a generative classifier (ctd.)

$$\mathrm{p(c|x)} = \sum_{k=1}^{K} \underbrace{\frac{\int_{x \in R_c} p(k|x)dx \cdot p(c)}{p(k)}}_{p(c|k)} \cdot \underbrace{\frac{p(x|k)p(k)}{\sum_{k=1}^{K} p(x|k)p(k)}}_{p(c|k)}$$

- The conditional distributions $p(x|k)$ are multivariate Gaussian distributions with parameters μ_k and Σ_k, i.e. $\mathcal{N}(x|\mu_k, \Sigma_k)$.
- The product of normalisation factors of these Gaussian distributions and the mixture coefficients $p(k)$, the inverse value of the normalisation factor $\sum_{k=1}^{K} p(x|k)p(k)$, as well as the output probabilities $p(c|k)$ corresponds to the weights w_{kc} within the generic classifier model.

Gaussian Mixture Models (17)

Christian-Albrechts-Universität zu Kiel

Design of a generative classifier (ctd.)

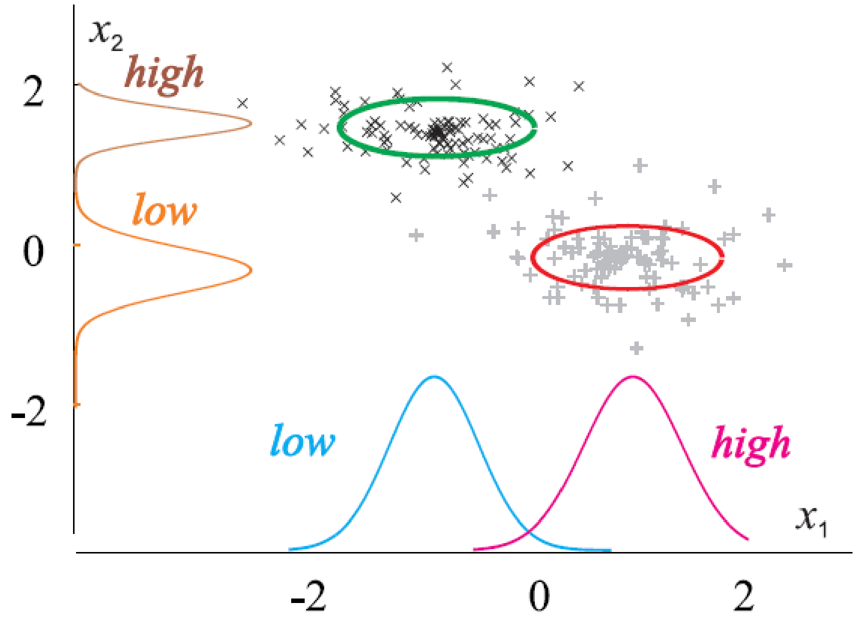
- How can we determine the parameters of the rule premises $p(c|k)$?
  → Maximum-Likelihood approach
  - A solution is not given as closed-form but as an iterative approach.
  - I.e. Expectation Maximisation (EM)!

- How can we determine the parameters of the rule conclusions $p(c|k)$?
  → Maximum-Likelihood approach
  - Here, a solution in closed form is possible.

Gaussian Mixture Models (18)

Christian-Albrechts-Universität zu Kiel

Example:



Rule system:

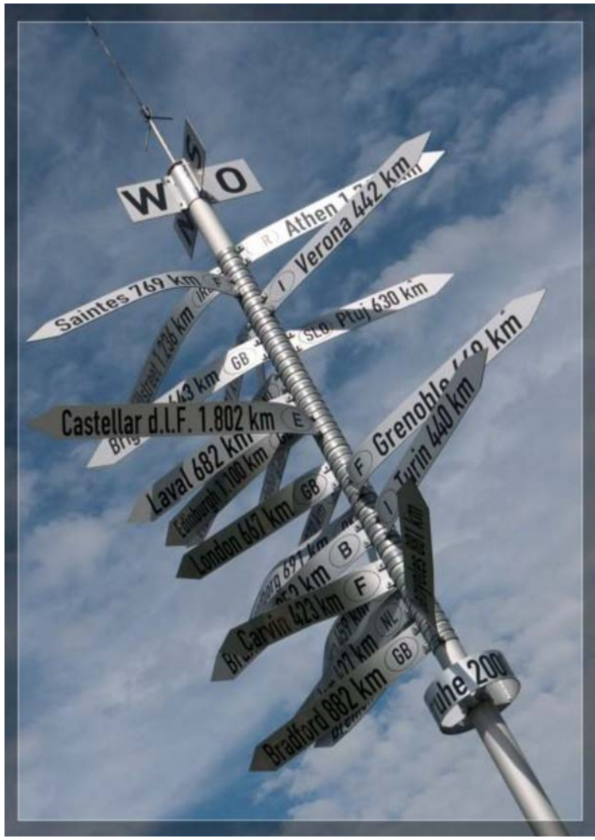
$$\text{if } x_1 \text{ is } low \text{ and } x_2 \text{ is } high \text{ then } Class_1 \text{ is } 0.98 \text{ and } Class_2 \text{ is } 0.02$$
$$\text{if } x_1 \text{ is } high \text{ and } x_2 \text{ is } low \text{ then } Class_1 \text{ is } 0.04 \text{ and } Class_2 \text{ is } 0.96$$

- (if ellipses are axis-oriented, i.e. covariance matrices Σ_k are diagonal)

Agenda

Christian-Albrechts-Universität zu Kiel

- Generative Modelling
- Gaussian Mixture Models
- Training Gaussian Mixture Models
- Applications in intelligent systems
- Conclusion and further readings



Training GMMs

Problem

- Identification of groups or clusters of data points in multi-dimensional input spaces
- Given: data set $\{x_1, \ldots, x_N\}$ consisting of N observations of a D-dimensional random variable x.
- Goal: Partition data into K cluster (assume that K is given)
- By intuition, a cluster is a group of data points with small distances between data points within the cluster (in comparison to distances to data points outside the cluster)

Training GMMs (2)

Christian-Albrechts-Universität zu Kiel

Training GMMs

- We introduce a set μ_k of D-dimensional vectors with $k = 1, \dots, K$ where μ_k is the prototype associated with the k-th cluster.
- By intuition, μ_k defines the centre (i.e. the mass centre/centroid) of the cluster.
- Goal:
  - Find an assignment of data points to clusters and a set of vectors $\{\mu_k\}$ such that the quadratic sum of all distances between all data points and their nearest vector μ_k is minimal.

Training GMMs (3)

Training GMMs

- For describing the assignment of data points to clusters, we introduce a set of binary indicator variables $r_{nk} \in \{0,1\}$ with $k = 1, \dots, K$.
- These indicator variables describe to which cluster the data point belongs to.
- If x_n belongs to cluster k: $r_{nk} = 1$ and $r_{nj} = 0$ for all $j \neq k$.
  → 1-out-of-k encoding scheme
- The goal function (distortion measure) is:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- J describes the quadratic sum of the distances of each data point to the corresponding vector μ_k.

Training GMMs (5)

Christian-Albrechts-Universität zu Kiel

Training GMMs: Goal

- The goal is to determine the values of $\{r_{nk}\}$ and $\{\mu_k\}$
  → This means to minimise J

Approach: Iterative process

- Two alternating steps: optimising r_{nk} and μ_k.
  → Choose initial values for μ_k.
  - E (expectation): In a first step, J is minimised concerning r_{nk}. This means to work with a fixed μ_k.
  - M (maximisation): In a second step, J is minimised concerning μ_k. This means to work with a fixed r_{nk}.
- The two steps are repeated until the process converges.
- We'll see in the following (when considering the actual process) why these steps are called expectation and maximisation.

Training GMMs (6)

Christian-Albrechts-Universität zu Kiel

E step: Determining r_{nk}

- J is a linear function of r_{nk} → an optimisation can be done easily.
- The terms for different n are independent and, consequently, we can choose the value of $r_{nk} = 1$ for each n for that k, which has the minimal value of $||x_n - \mu_k||^2$.
- This means: The n-th data point is assigned to the nearest cluster centre:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \\ 0 & \text{otherwise} \end{cases}$$

Training GMMs (7)

Christian-Albrechts-Universität zu Kiel

M step: Determining μ_k

- J is a quadratic function of μ_k
- Derivation according to μ_k and equating with zero results in:

$$\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

- The denominator is equal to the number of data points assigned to this cluster. This means: The value of μ_k is equal to the centre (i.e. the mean) of all data points x_n belonging to cluster k.
- This is the reason why the approach is also known as the k-means algorithm.

Training GMMs (8)

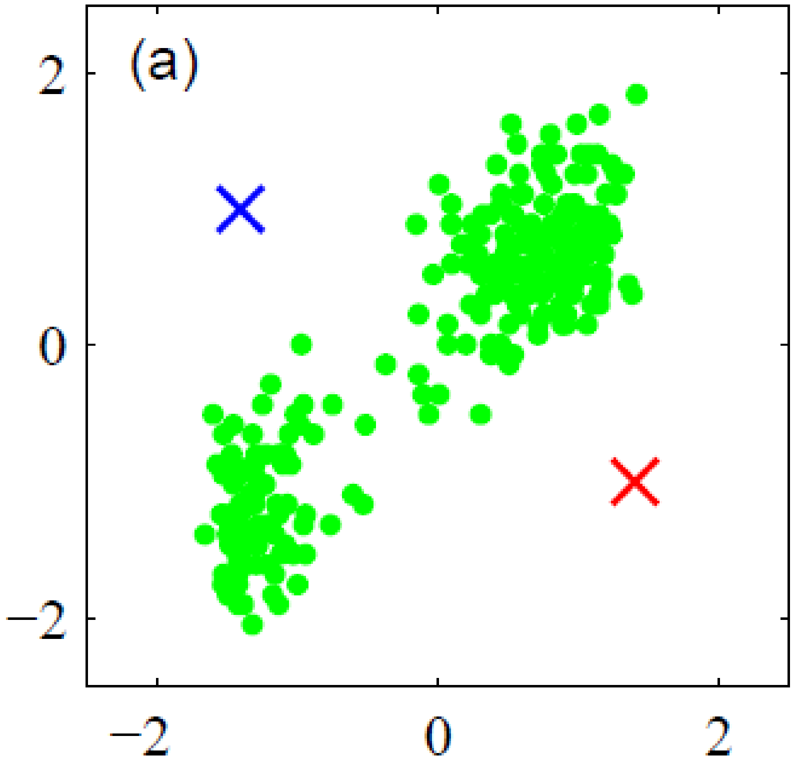
Comments on the approach

- E and M step are repeated until the process converges (i.e. there is no change in the assignments) or a certain stopping criterion is met (e.g. a number of iterations).
- Since each step reduces the value of the goal function J, a converging behaviour is guaranteed.
- However: We may converge to a local minimum rather than to the global minimum!

Training GMMs (9)

Example: Old Faithful again…

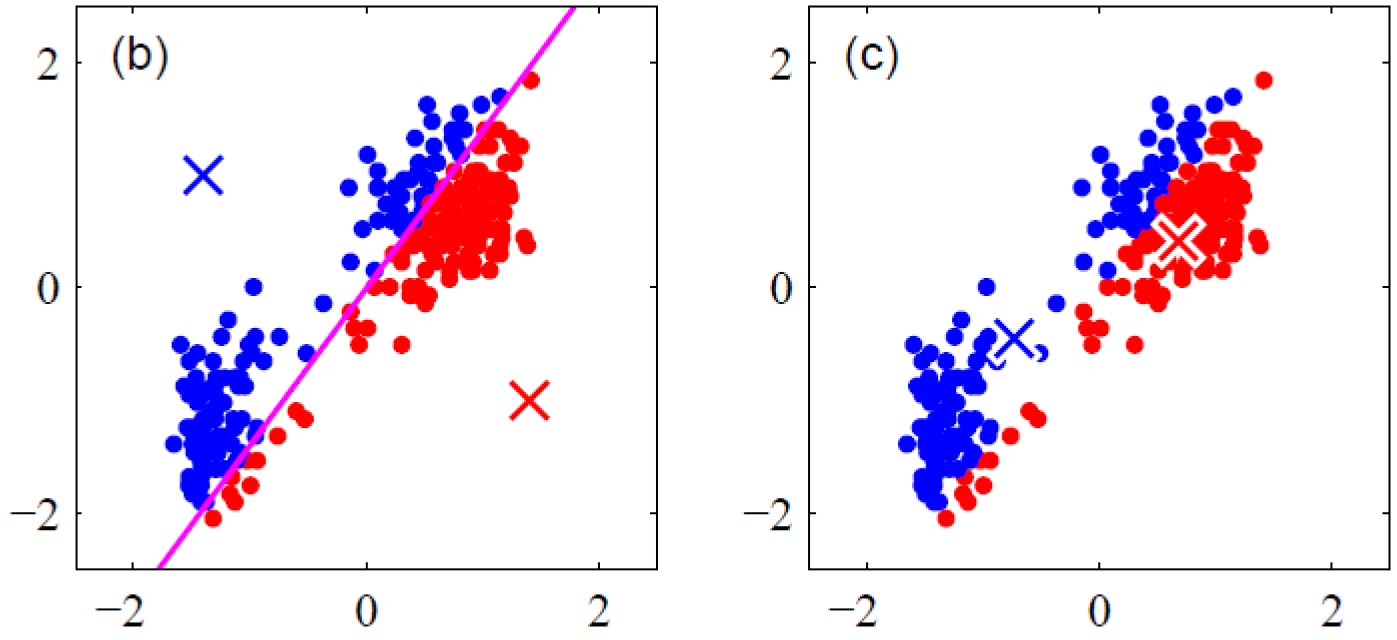
- The data have been standardised, i.e. the variables have a mean of 0 and a standard deviation of 1.
- We used $k = 2$.
- Initial cluster centres are the red and blue crosses.
  $\rightarrow$ By intention, “bad” solutions for the centres have been chosen to demonstrate the process.



Training GMMs (10)

Example: Old Faithful again…

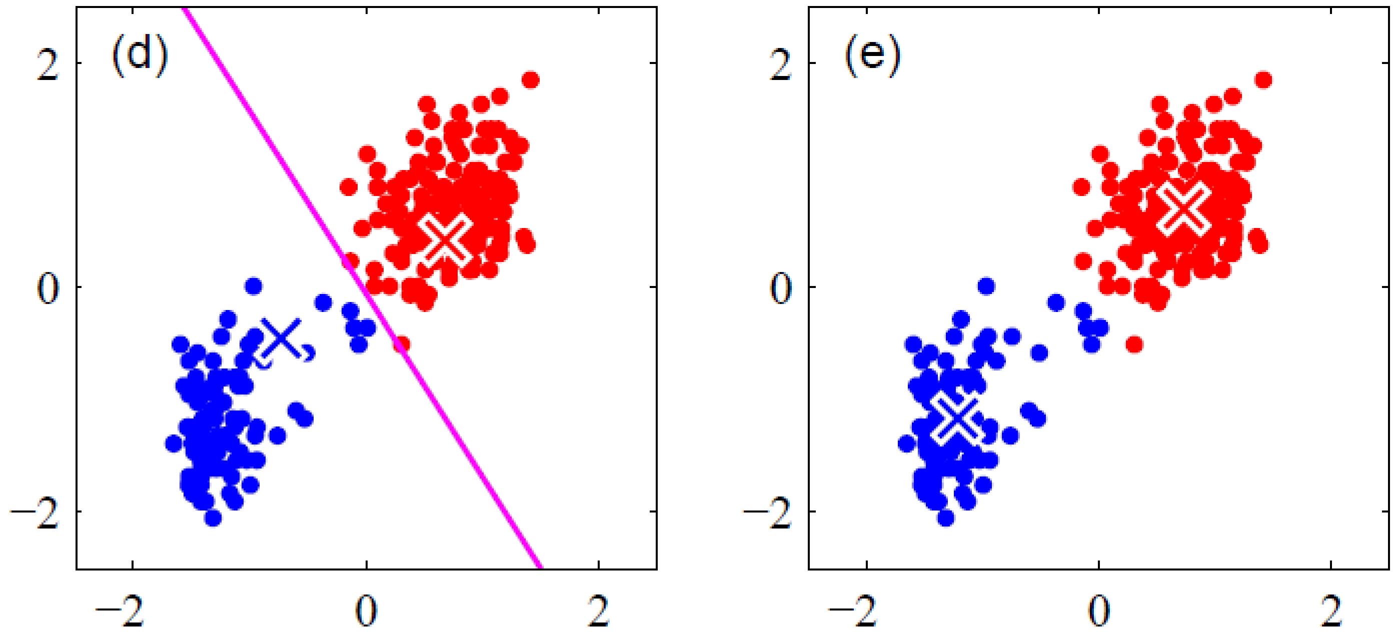
- First E step (left side) and first M step (right side)
- The purple line can be considered to be the decision boundary if a classification decision has to be done. This is the result of assigning the data point to centres.



Training GMMs (11)

Example: Old Faithful again…

- Second E step (left side) and second M step (right side)
- The result is much better…

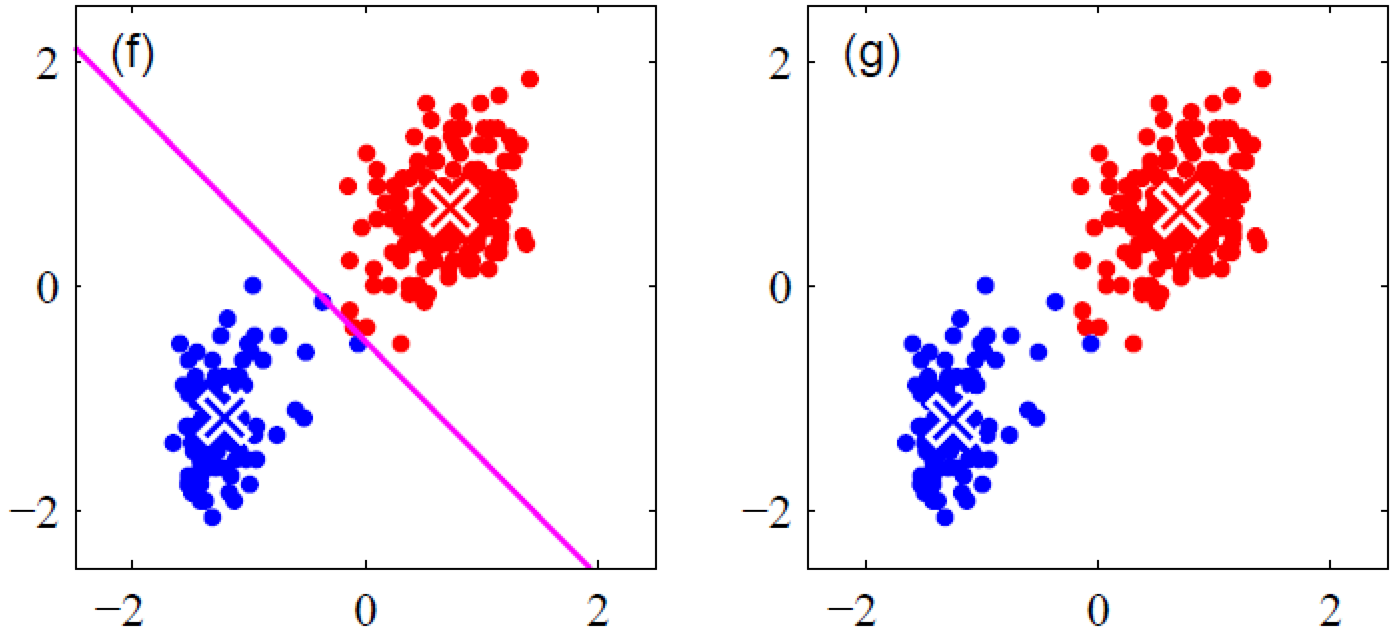


Training GMMs (12)

Christian-Albrechts-Universität zu Kiel

Example: Old Faithful again…

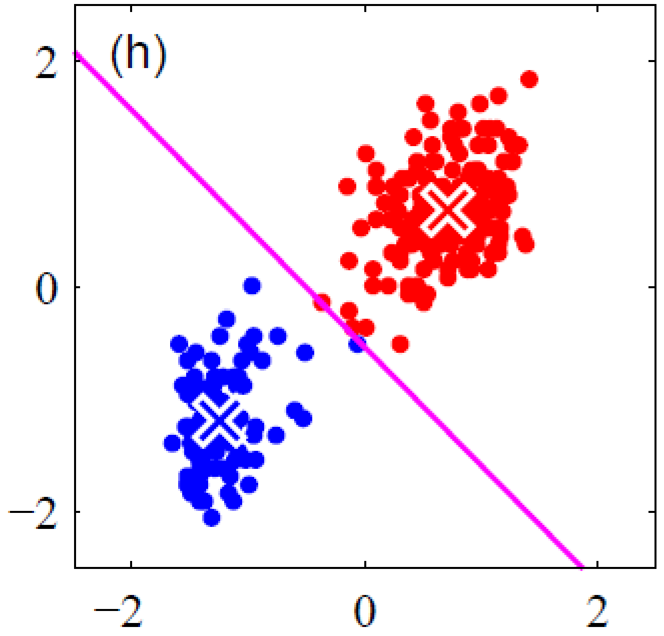
- Third E step (left side) and third M step (right side)
- Only a few data points change their associations…

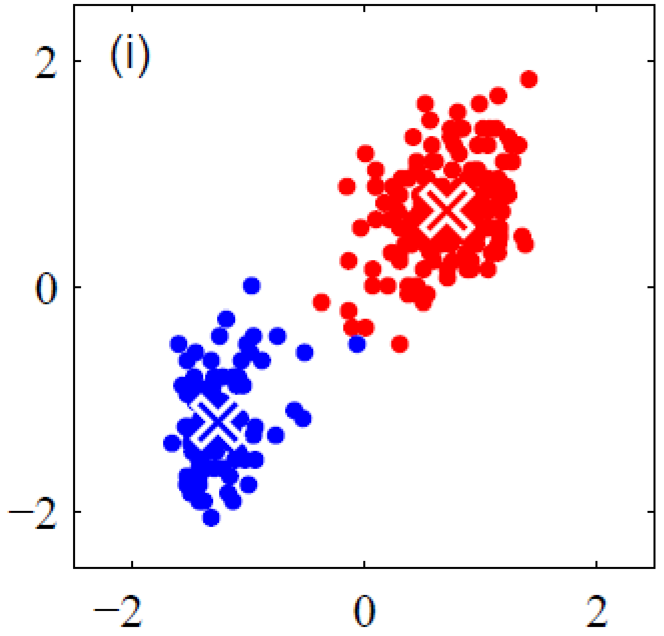


Training GMMs (13)

Example: Old Faithful again…

- Fourth E step (left side) and fourth M step (right side)
- Only one data point changes its association. The process converges. The next iteration would not see any differences…

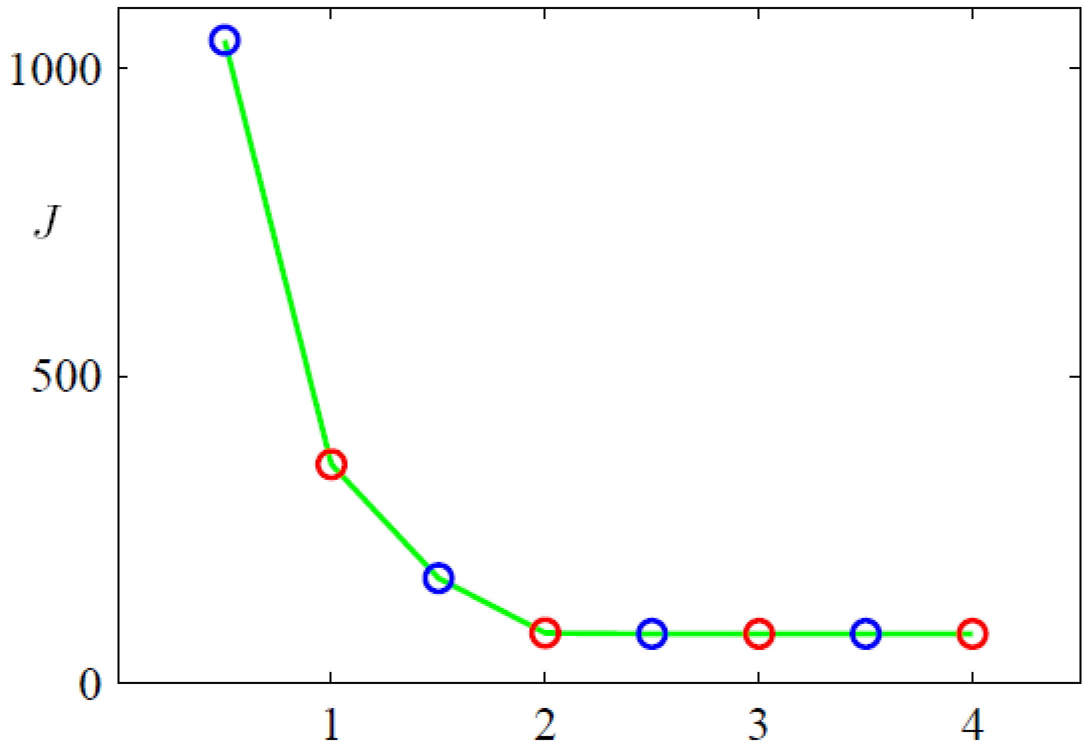




Training GMMs (14)

Example: Old Faithful again…

- Development of the goal function value J after each E step (blue circles) and M step (red circles).



Training GMMs (15)

Some remarks on this k-means approach

- In real-world approaches, more sophisticated initialisation concepts are used.
- For instance, a certain sub-set of the data is chosen as initial centres.
- This k-means approach serves as a basis to determine/initialise the parameters of a GMM → we'll see how this works later on.
- Direct implementations tend to be pretty slow.
  → Appropriate data structures for fast access are based on trees, for instance …

Training GMMs (16)

Batch vs. online version

- Besides the presented batch version, there exist online approaches of the k-means algorithm, i.e. stochastic variants.
- In sequential update steps, an update step is performed for each data point μ_k regarding the closest prototype:

$$\mu_k^{new} = \mu_k^{old} + \eta_n(x_n - \mu_k^{old})$$

- In this formula, η_n is the learning rate.
  → η_n is typically decreased monotonically with the number of data points.

Training GMMs (17)

From k-means to k-medoids clustering

- Originally, k-means is based on the Euclidian distance
  → This is especially bad for categorical variables, but also outliers.
- k-means can be generalised by using an abstract dissimilarity measurement $\mathcal{V}(x, x')$ defined for two vectors x and x':

$$\widetilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\mathbf{x}, \mathbf{x}')$$

- This measure has to be minimised again.
- The resulting algorithm is called k-medoids.
- k-medoids has a complexity of $O(KN)$ – as k-means.

Training GMMs (18)

Christian-Albrechts-Universität zu Kiel

Expectation Maximisation for GMM

- We need the concept of latent variables in the context of GMM.
- A latent variable is not directly observable in experiments.
- Let z be a K-dimensional binary variable using a 1-out-of-K representation. A specific element z_k is 1, all other is 0.
- As a consequence:

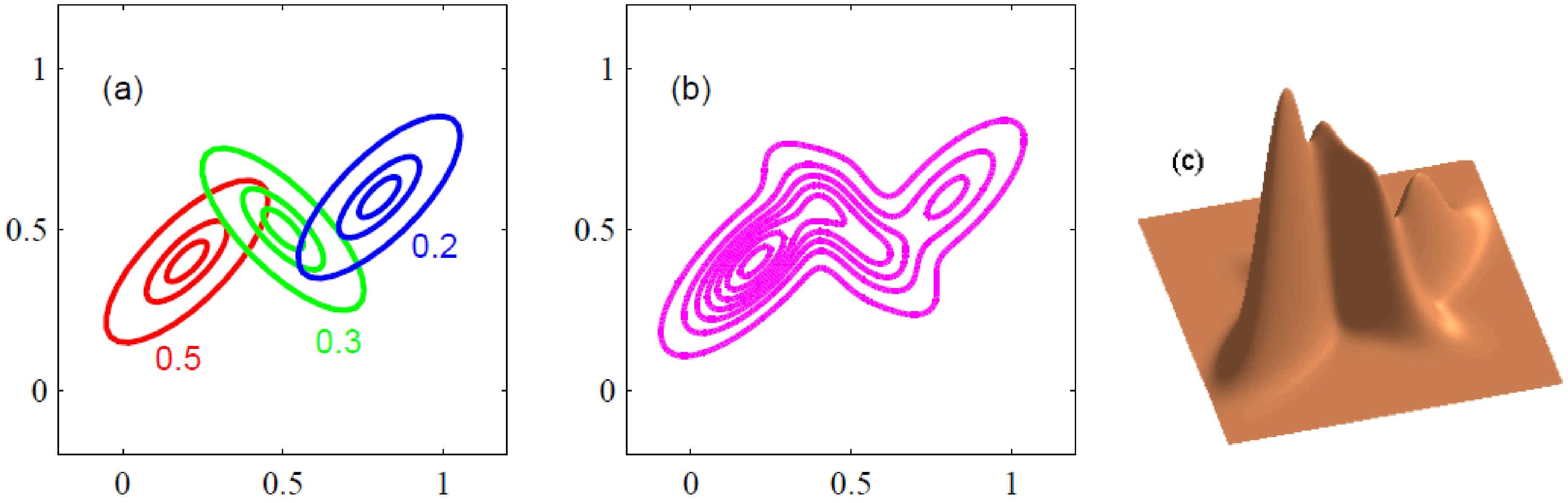
$$z_k \in \{0,1\} \text{ and } \sum_k z_k = 1$$

- z can have K realisations.
- The joint distribution $p(x, z)$ can be expressed using the conditional distribution $p(x|z)$ and the marginal distribution $p(x)$.

Training GMMs (19)

Back to the initial example for Gaussian Mixture Models

- In the following, we'll use this for the generation of data points.

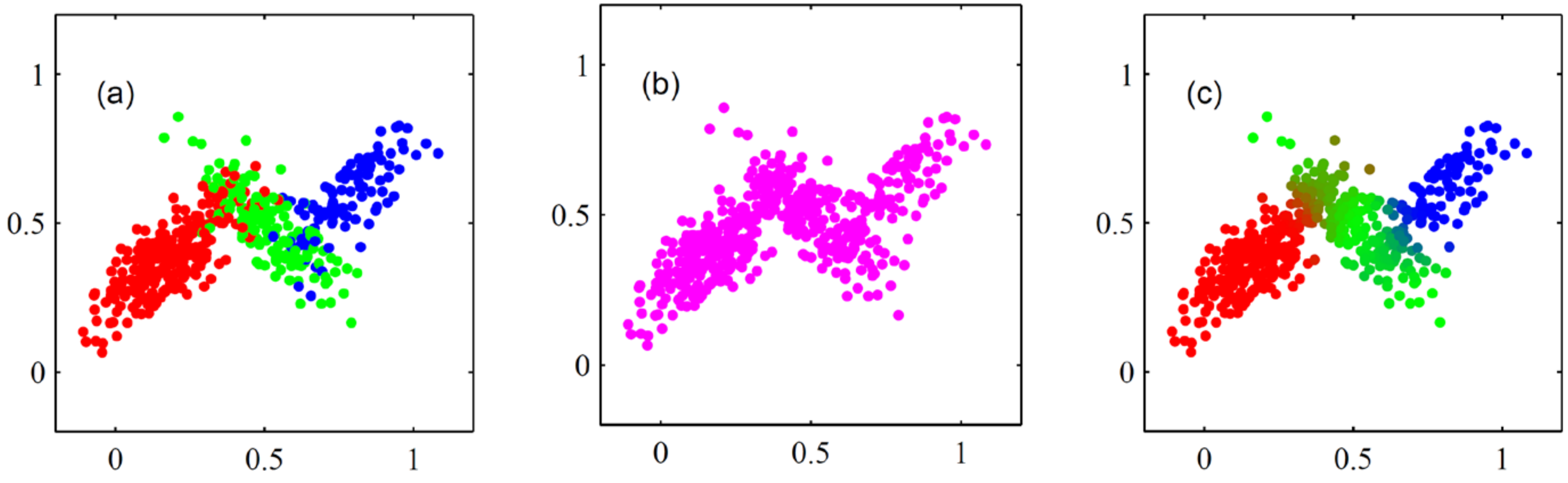


Training GMMs (20)

a) Data points of the joint distribution $p(x, z) = p(z)p(x|z)$, components are coloured accordingly. This data set is called complete.

b) Marginal distribution $p(x)$, emerges by ignoring the values of z. This data set is called incomplete.

c) Colours represent the values of the responsibilities $\gamma(z_{nk})$ associated with the data points x_n. Colours are chosen depending on the fraction of red, green, blue.



Training GMMs (21)

Maximum Likelihood

- Given: Set of observations $\{x_1, \ldots, x_N\}$. These data shall be modelled using a GMM. The data set can be represented using an $(N \times D)$-matrix X with the n-th cell is given by x_n^T.
- We assume that these data have been derived independently of each other and from the investigated distribution (i.i.d.).
- With

$$p(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x|\mu_k, \Sigma_k)$$

holds for the Log-Likelihood-Function:

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \right\}$$

Training GMMs (22)

Singularities

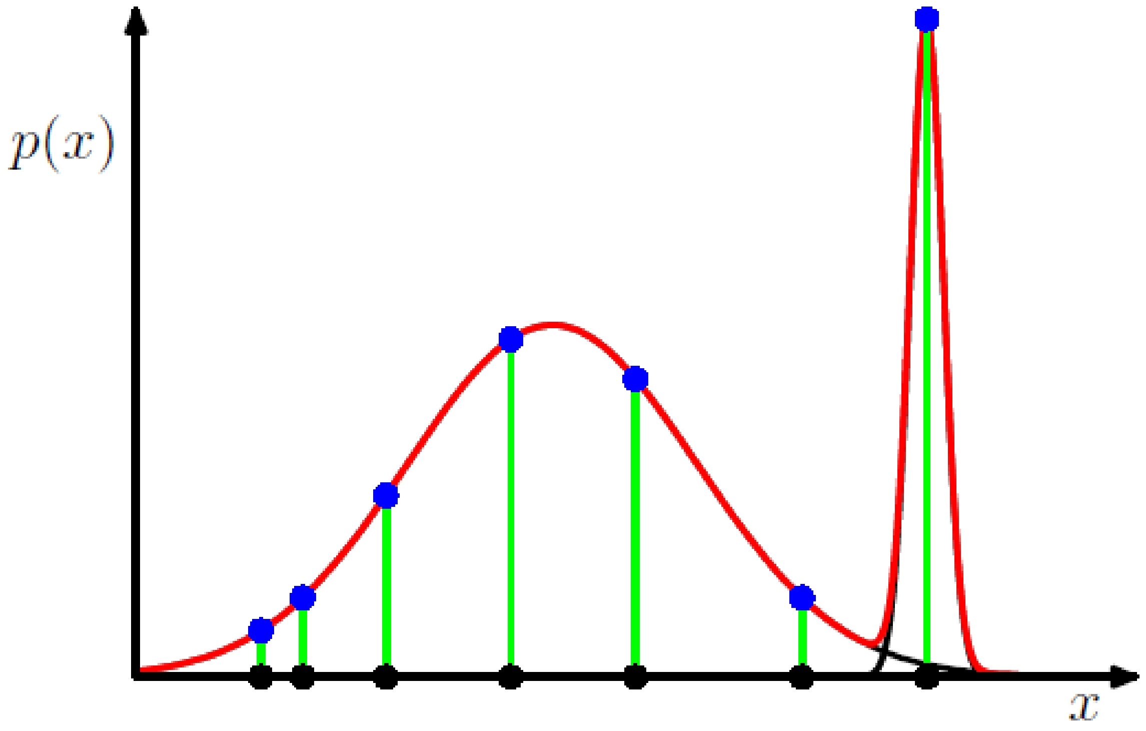
- ML approaches for GMMs can result in severe problems: singularities.
- Example:
  - In GMM, the covariance matrices have the form $\Sigma_k = \sigma_k^2 \, I$ with I being the identity/unit matrix (holds also for general covariance matrices).
  - Assume that for one component (here: j) the mean μ_j is equal to one data point, i.e. $\mu_j = x_n$ for one n.
  - This data point provides one term for the Log-Likelihood-Function:

$$\mathcal{N}\left(x_n \middle| x_n, \sigma_j^2, I\right) = \frac{1}{(2\pi)^{\frac{1}{2}}} \frac{1}{\sigma_j}$$

  - For the one-sided limit of $\sigma_j \to \infty$ this term approaches ∞ and, in consequence, the Log-Likelihood-Function approaches ∞…

Training GMMs (23)

Example for singularities



Training GMMs (24)

Remarks

- Consequently, the Log-Likelihood-Function is problematic.
  → Singularities are observed quite often in real-world data.
- We call this effect “collapsing of a component within a data point“.
- This case does not occur for an individual Gaussian, only for GMMs.
- How to avoid this effect?
  → There are heuristics that can detect singularities. After detection, set the mean to a random value and continue the optimisation process.

Training GMMs (25)

Christian-Albrechts-Universität zu Kiel

Remarks

- Maximising the Log-Likelihood-Function for a GMM is more difficult than for individual Gaussian distribution, since we sum up over k in the logarithm.
- Derivation and equalisation to zero does not lead to a solution in closed form.
- We can apply gradient-based techniques to solve the problem.
- In general, the EM algorithm is applied here.
- The EM (Expectation Maximisation) algorithm is a generally applicable approach – however, we'll focus on GMM here.

Training GMMs (26)

Christian-Albrechts-Universität zu Kiel

EM for GMMs

- Derivation according to μ_k and setting to zero results in:

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}}_{\gamma(z_{nk})} \sum_k (x_n - \mu_k)$$

- The responsibilities appear on the right-hand side.
- By multiplication with Σ_k^{-1} (needs to be non-singular), we get:

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})\ x_n$$

- with:

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Training GMMs (27)

Christian-Albrechts-Universität zu Kiel

EM for GMMs

- The term N_k can be considered to describe the “effective” number of data points assigned to the cluster.
- The mean μ_k is the weighted mean of all data points in the cluster k.
- The weighting factor for each data point x_n is given by the responsibility $\gamma(z_{nk})$, i.e. the a-posteriori probability that component k has been responsible for generating this data point.
- Derivation according to Σ_k and setting to zero results in:

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})\,(x_n - \mu_k)(x_n - \mu_k)^T$$

- Again, each data point is weighted by the responsibility and we again divide by the number of effective data points.

Training GMMs (28)

Christian-Albrechts-Universität zu Kiel

Mixture coefficients

- We have to further keep in mind that the sum of all mixture coefficients has to sum up to 1.
- As a consequence, we receive:

$$\pi_k = \frac{N_k}{N}$$

- We can interpret this as the averaged responsibility of component k for the data points.

Training GMMs (29)

- Since, in turn, the responsibilities depend on these parameters via

$$\gamma(z_k) = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}$$

- the result is not representable in a closed-form.
- But, well, it indicates already that an iterative approach can be applied again…

Training GMMs (30)

Christian-Albrechts-Universität zu Kiel

EM for GMMs

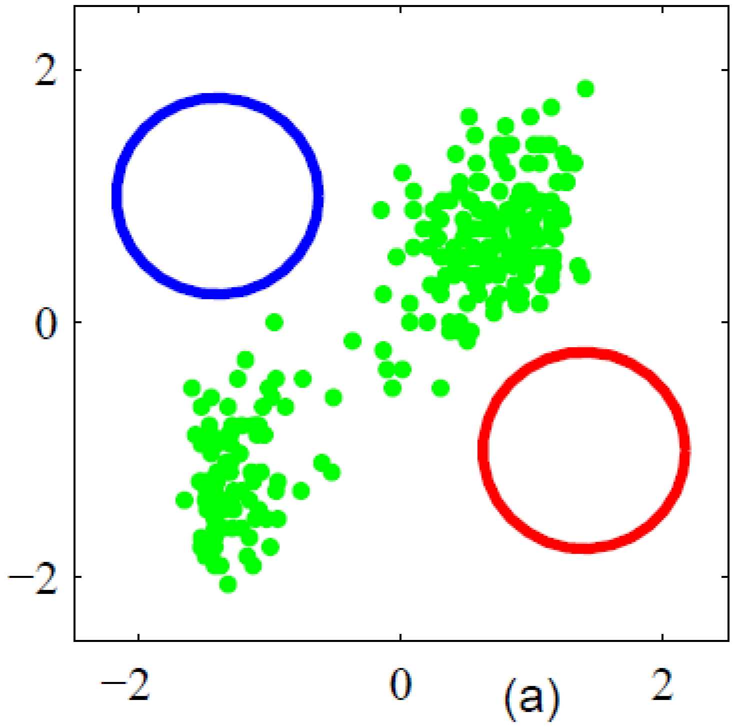
- In a first step, we have to define the initial configuration of the parameters for mean, covariances, and mixture coefficients.
- Afterwards, these two update steps are performed alternatingly:
  - E step (expectation): Using the current values of the parameters, we estimate the responsibilities.
  - M step (maximisation): Using the responsibilities, we estimate the parameters.
- Proofs are showing that the value of the Log-Likelihood-Function is guaranteed to increase in each update step (neglected here).

Training GMMs (31)

Christian-Albrechts-Universität zu Kiel

EM for GMMs: Example

- Old Faithful again…
- Data points and initial values for centres and covariances:

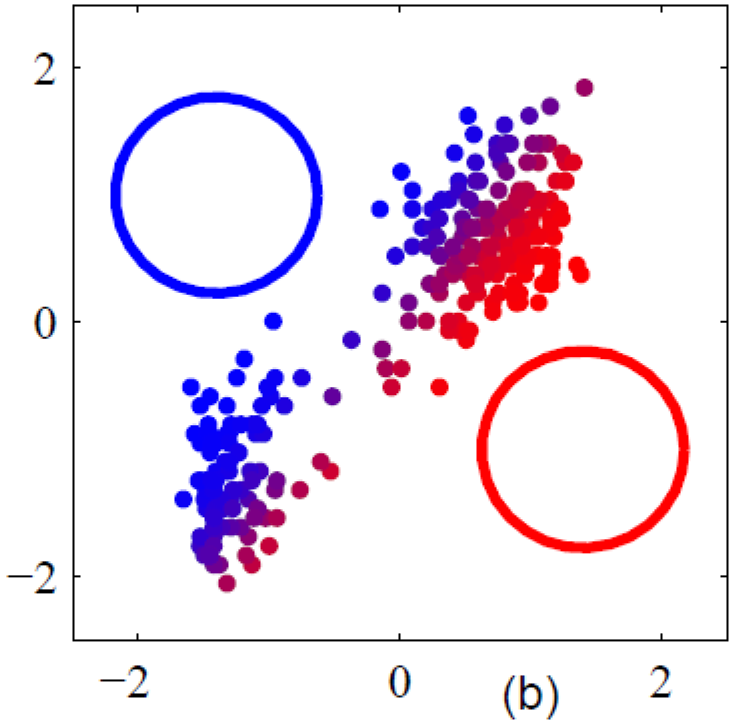


Training GMMs (32)

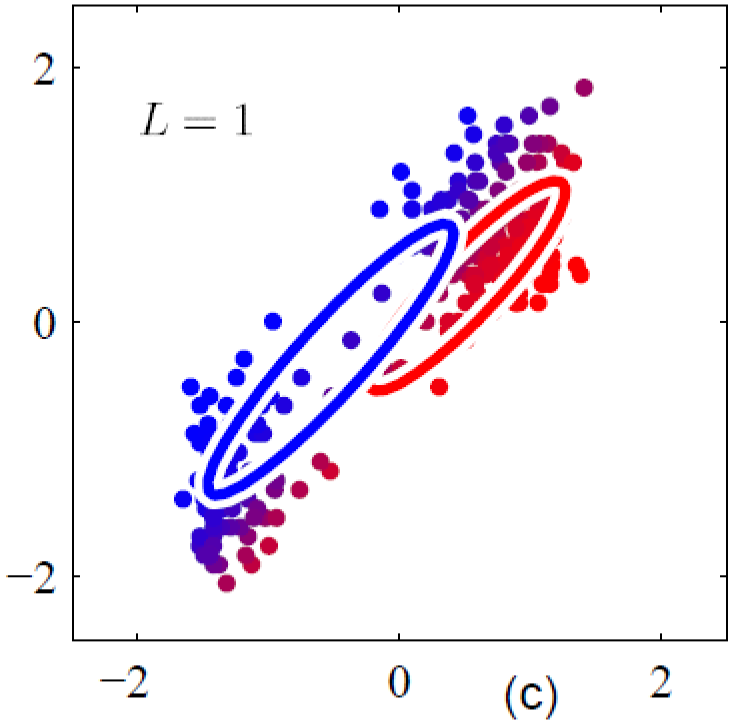
Christian-Albrechts-Universität zu Kiel

EM for GMMs: Example

- Initial E step (responsibilities, left) and initial M step (parameters, right)



(b)

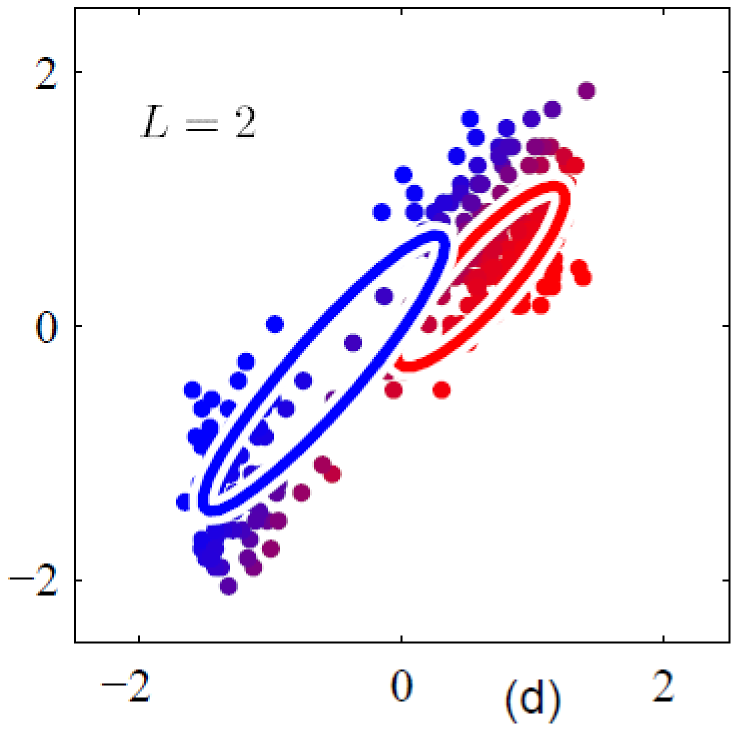


(c)

Training GMMs (33)

EM for GMMs: Example

- Result of EM after two steps

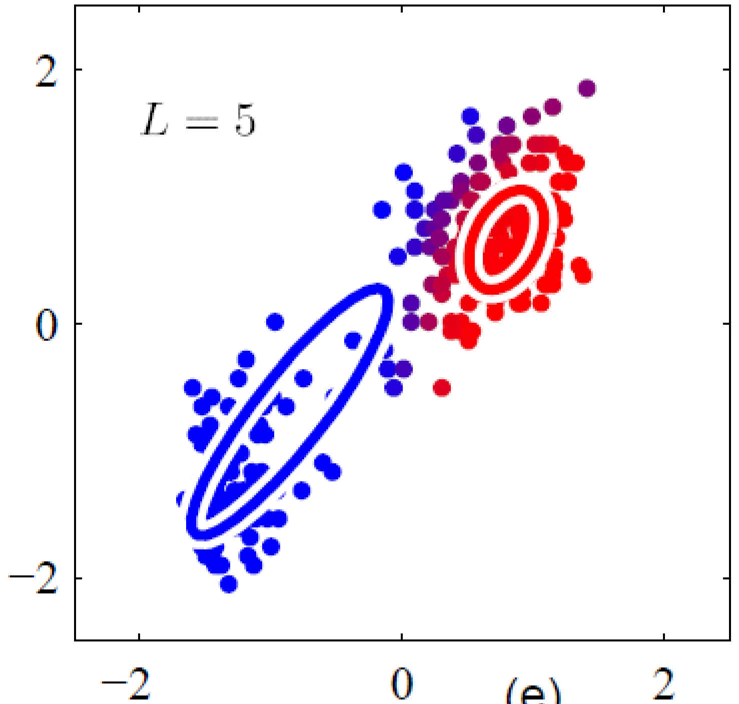


Training GMMs (34)

Christian-Albrechts-Universität zu Kiel

EM for GMMs: Example

- Result of EM after five steps

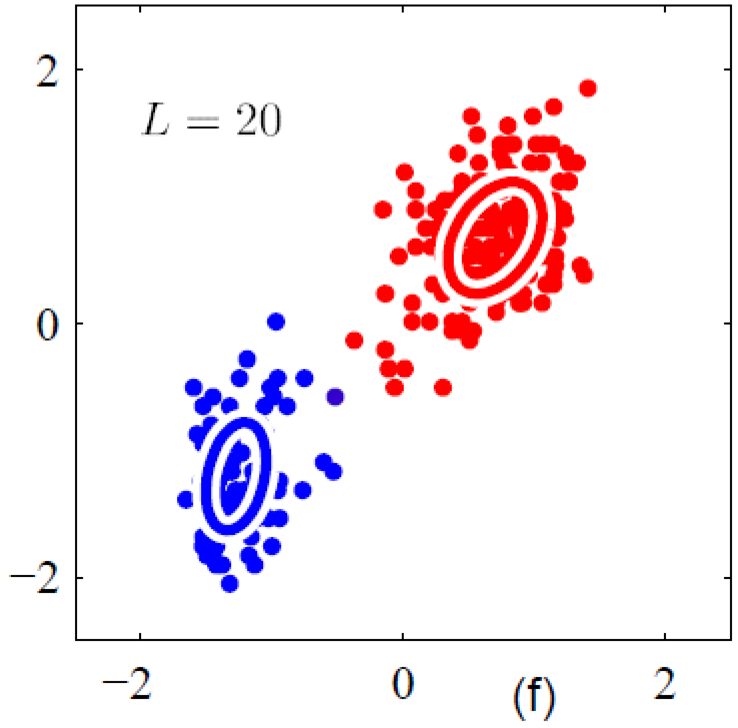


(e)

Training GMMs (35)

EM for GMMs: Example

- Result of EM after twenty steps



Training GMMs (36)

Remarks

- In general, EM needs more steps for convergence than k-means and each step requires more computational effort.
- As a consequence, k-means is typically used to initialise the EM algorithm:
  - Centres are initialised according to the centres found by k-means.
  - Covariances are determined from the variances of the detected clusters.
  - Mixture coefficients are defined according to the fraction of data points assigned to the clusters.
- There is no guarantee that EM will identify the global optimum. However, there are typically more than 1 optima.
- We need to process appropriate techniques to avoid singularities.

Summary: Training GMMs

Christian-Albrechts-Universität zu Kiel

Summary: Training GMMs

- The goal of a GMM is to maximise the Log-Likelihood-Function using the parameters (means, covariances, mixture coefficients).

1. Initialise the means μ_k, the covariances Σ_k, and the mixture coefficients π_k. Determine the initial value of the Log-Likelihood-Function.
2. E step: Determine the responsibilities using the current parameters:

$$\gamma(z_k) = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}$$

Summary: Training GMMs (2)

3. M step: Determine the parameters with the current responsibilities:

$$\mu_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\,x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\,(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

with:

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk})$$

Summary: Training GMMs (3)

4. Log-Likelihood-Function: Evaluate the Log-Likelihood-Function

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}$$

and test for convergence or stopping criterion (either parameter or Log-Likelihood-Function).

→ If stopping criterion or convergence are not reached, continue with step 2!

Summary: Training GMMs (4)

Comparison of k-means and EM for GMM

- Both approaches are pretty similar:
  - k-means is based on a “hard“ assignment of data points to clusters. In particular, each data point is assigned to exactly one cluster.
  - EM for GMM is based on a “soft“ assignment of data points to clusters. In particular, each data point is assigned to clusters using the a-posteriori probabilities.
- At an abstract level, k-means can be understood as a special instance of the EM algorithm for GMMs.

Summary: Training GMMs (5)

When using the estimation functions of the EM algorithm, the following holds:

- The estimation of centres is in certain (extreme) cases equal to k-means.
- The covariances are not estimated.
- The mixture coefficients are not estimated.

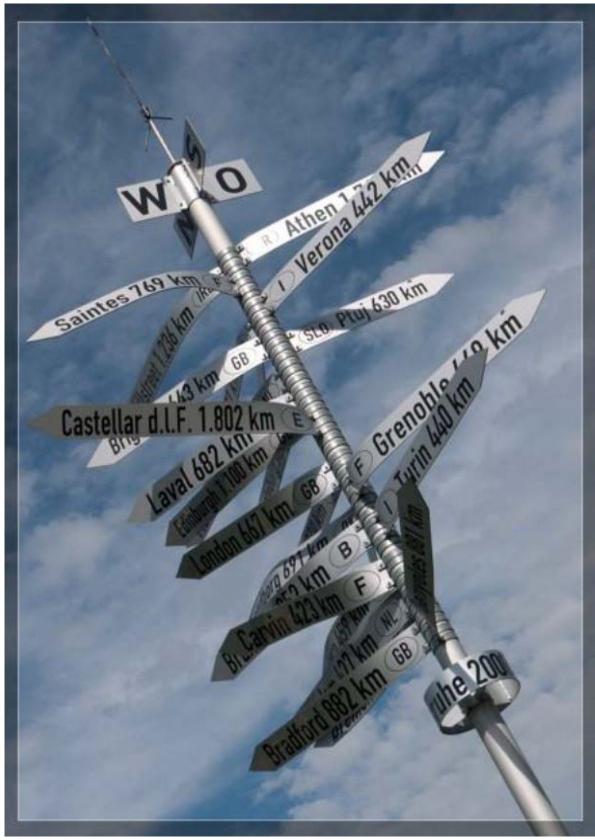
Remarks:

- If the mixture coefficients were set to the fraction of data points this does not have any meaning. The reason is that the mixture coefficients do not play an active role in the EM algorithm anymore.
- A variant of the algorithm with a static (hard) assignment of data points but general covariance matrices is called elliptic k-means algorithm.

Agenda

Christian-Albrechts-Universität zu Kiel

- Generative Modelling
- Gaussian Mixture Models
- Training Gaussian Mixture Models
- Applications in intelligent systems
- Conclusion and further readings



Applications of GMM in OC systems

What are application scenarios for the GMM approach?

- Classification of input data as a pre-processing step
- Classification of environmental states (e.g. as part of the situation description)
- Classification of internal states (e.g. as part of the situation description)
- Novelty/anomaly detection: Changes in the behaviour of some processes that cannot be explained by the knowledge encoded in the GMM.
- …

Applications of GMM in OC systems (2)

Christian-Albrechts-Universität zu Kiel

Construction of a probabilistic classifier

- The EM algorithm can be used to determine the parameters of a GMM.
- As a consequence, we have estimates of the parameters of $p(k|x)$ available in the probabilistic classifier (unsupervised training step):

$$\mathrm{p(c|x)} = \sum_{k=1}^{K} \underbrace{\frac{\int_{x \in R_c} p(k|x)dx \cdot p(c)}{p(k)}}_{p(c|k)} \cdot \underbrace{\frac{p(x|k)p(k)}{\sum_{k=1}^{K} p(x|k)p(k)}}_{p(c|k)}$$

Applications of GMM in OC systems (3)

Construction of a probabilistic classifier

- Within this probabilistic classifier, we estimate the output weights using the parameters $\xi_{k,c}$ of a multinominal distribution $p(c|k)$ for each component.
- For this unsupervised training step, we need the set of targets T for the data set X.
- The term I_c describes the index set of all data points (observations) that belong to class c. This results in:

$$\begin{aligned} \xi_{k,c} &= \frac{\frac{1}{|I_c|}\sum_{n\in I_c}\gamma(z_{n,k})\cdot\frac{|I_c|}{N}}{\frac{N_k}{N}} \\ &= \frac{1}{N_k}\sum_{n\in I_c}\gamma(z_{n,j}) \end{aligned}$$

Applications of GMM in OC systems (4)

Remarks

- We may use other distributions for these components.
  - E.g. the Student distribution if several outliers are within the data set.
  - E.g. the multi-nominal distribution for categorical variables.
- In different dimensions of the input space, different distributions could be applied.
  → This results in hybrid mixture models.
- There are alternatives to EM that come with advantages (e.g. being more robust in terms of singularities or automatically identifying the number of required components).

Applications of GMM in OC systems (5)

Comparison to other classifiers

- A probabilistic classifier comes with the different advantages of generative classifiers.
- A discriminative classifier (e.g. a Support Vector Machine) typically achieves a better classification result.
- A classifier such as a Fuzzy classifier consists of several readable rules (although they might be really difficult to understand)
- A Neural Network (such as a Radial Basis Function Network) can have different properties depending on the target function and the optimisation approach.

→ All of these four classifiers can be realised in the same manner, i.e. they have the same functional form.

Applications of GMM in OC systems (6)

Anomaly/novelty detection

- Can be understood as unknown patterns in time series or unexpected behaviour of some processes generating data (unexpected means that is does not fit to the model that the system has been derived from previous observations).
- Is closely related to the detection of repeating patterns (motif detection).
- Basic idea is to model everything that is known (or has been observed before) and use this knowledge to detect behaviour that is not covered by these training data.
- It depends largely on the particular application how such an anomaly looks like or how it is distinguished from the extreme cases of normal behaviour.
- Similar to the detection of outliers, a threshold for the detection of non-normal behaviour is needed next to the basic model of expected behaviour.

Applications of GMM in OC systems (7)

The basic process of detecting anomalies

1. Model all known behaviour using training data
   - Ideally: Fully cover the input space with existing data
   - Within the initial model, we should consider that outlier, noise, and disturbances are available in the data
   - A model of the "normal conditions" could – but does not necessarily need to – cover explicitly the existing data
   - Only needed is a complete explanation of the existing data (e.g. thresholds are chosen in a way that normal variations or disturbances or deviations are not considered as anomalies)
2. Determine boundaries or thresholds from the training data to assess normal conditions and the maximally accepted deviation.
3. Apply online and search for anomalies
   - i.e. use in a real system with novel data

Applications of GMM in OC systems (8)

Several different approaches are known in the literature

- One-Class Support Vector Machines, …
- GMM-based approaches
  - Use GMM to model your observations (at design-time under test and/or at runtime, e.g. at system start-up in a supervised mode)
  - Use divergence measures to compare distributions (and consider thresholds of acceptable deviations between these distributions)

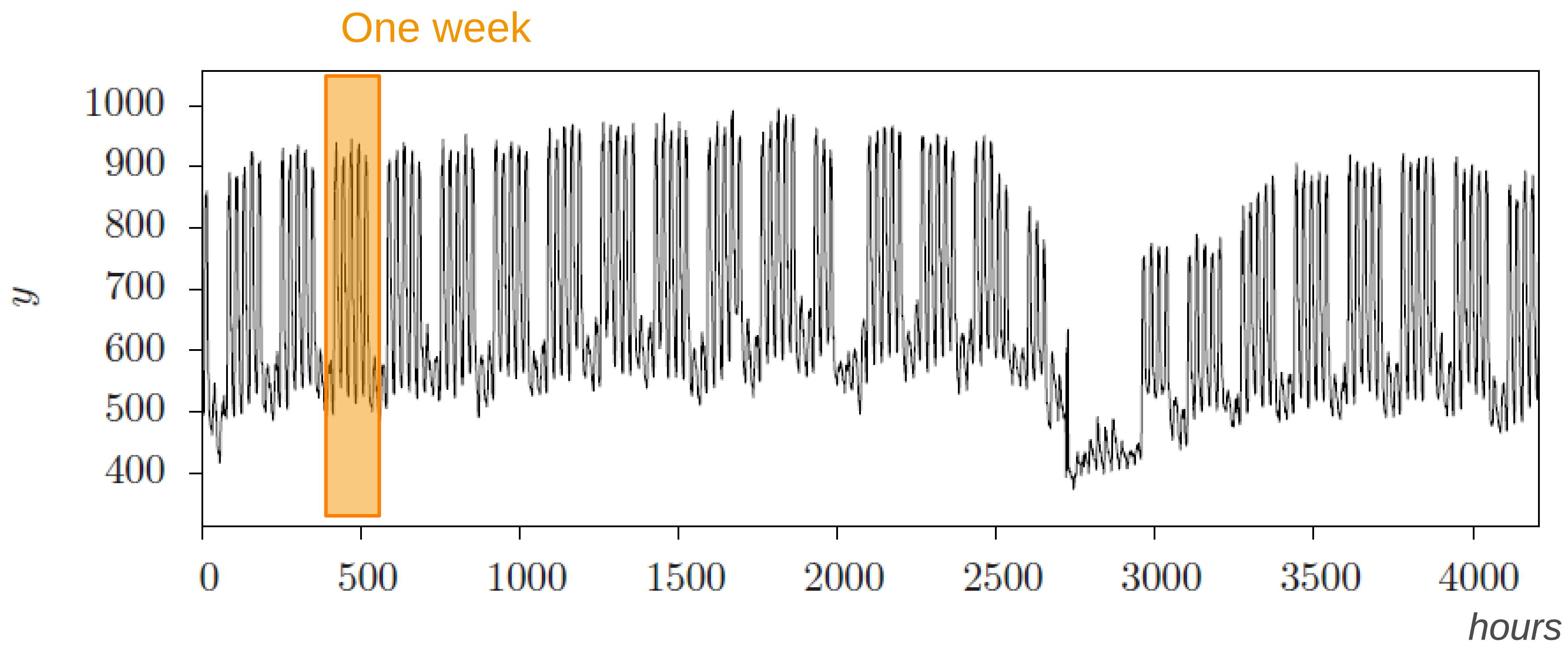
In general, we distinguish between two different goals:

- Search for anomalies in comparison to normal data (e.g. threshold-based approaches / automatically defined by One-Class SVM)
- Search for patterns that are maximally dissimilar to the normal behaviour (e.g. top-10 dissimilar sub-sequences on time series)

Applications of GMM in OC systems (9)

Christian-Albrechts-Universität zu Kiel

Example 1: Search for anomalies in the energy consumption of a building



Source: Prechelt, *PROBEN 1 - a set of benchmarks and benchmarking rules for neural network training algorithms*, 1994

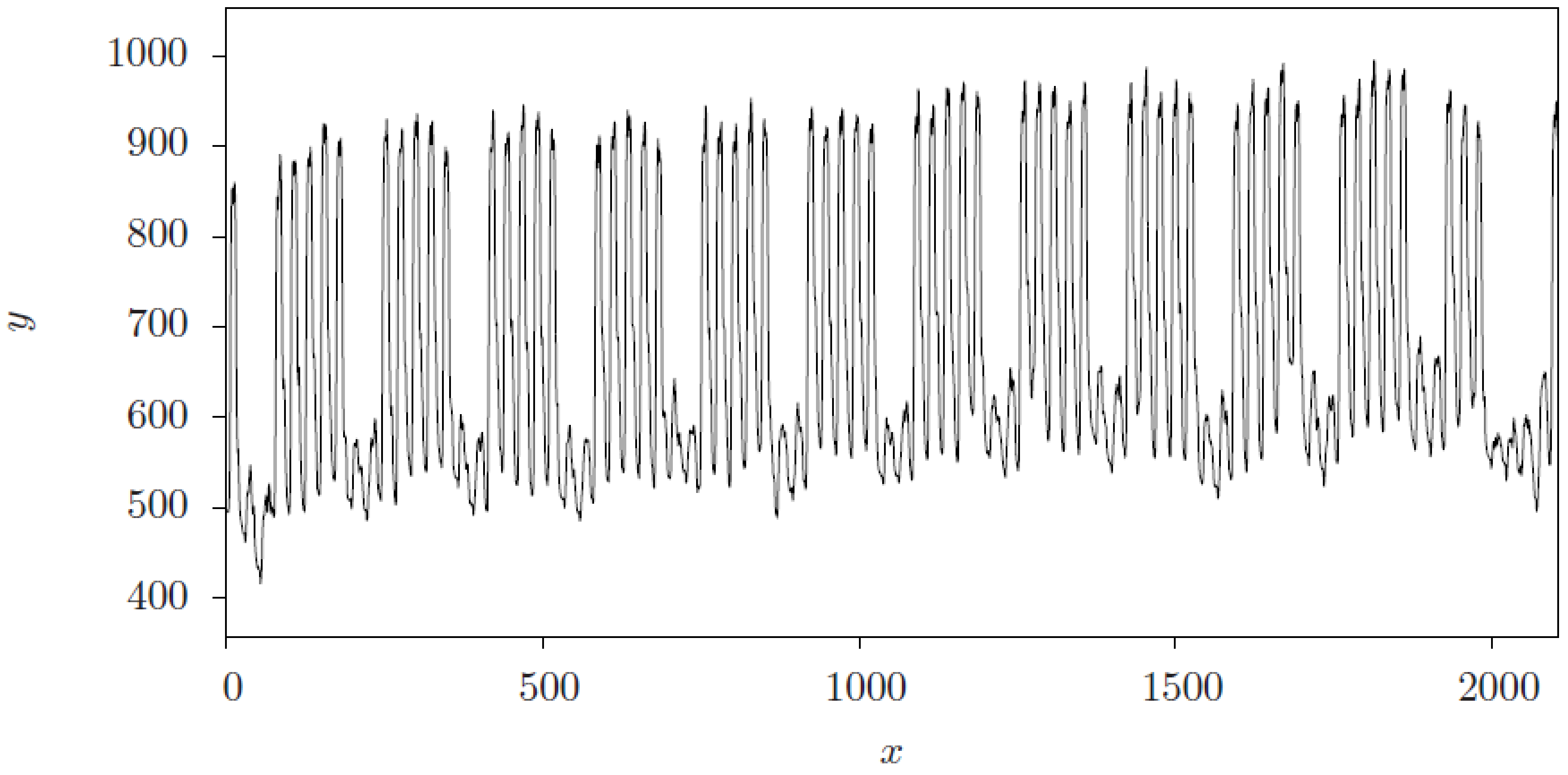
Applications of GMM in OC systems (10)

Approach

- Divide time series into parts for training and test (here: 50:50)
- Generate models for normal weekdays and weekends / bank holiday from training data.
- Search for abnormal days in the test data
- Sort the test days according to the similarity with the existing knowledge (i.e. the model as a result of the training data)

Applications of GMM in OC systems (11)

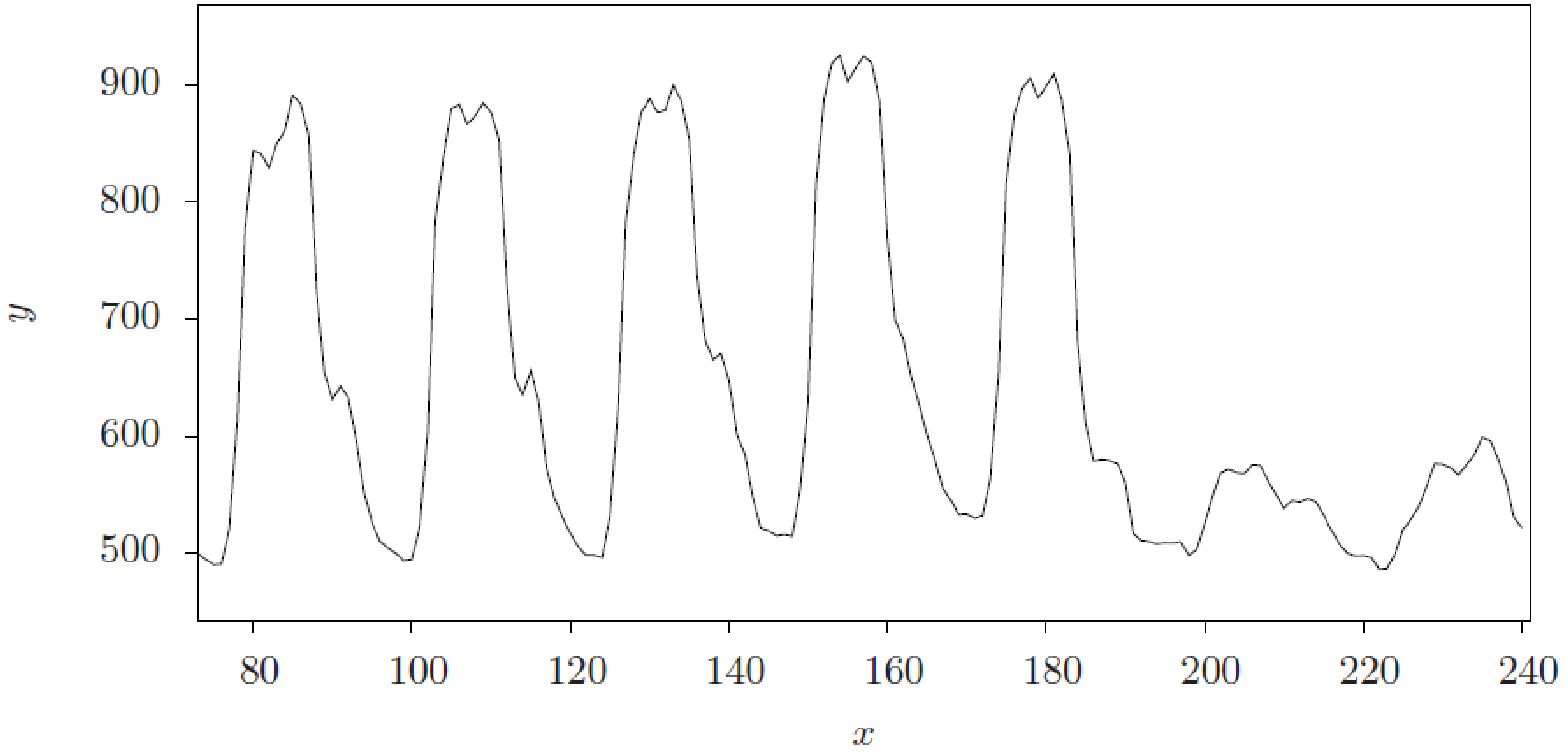
Training data



Applications of GMM in OC systems (12)

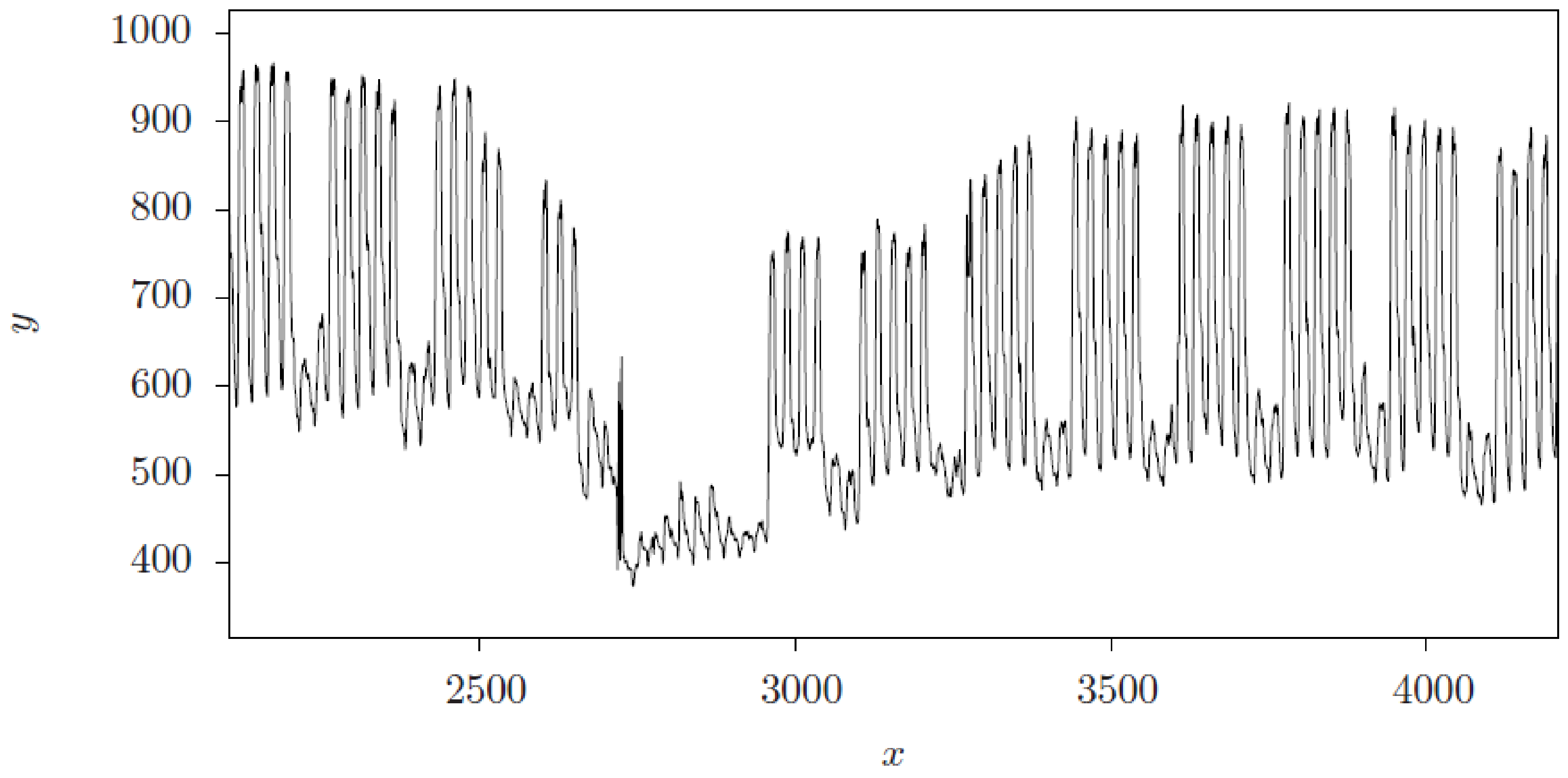
Christian-Albrechts-Universität zu Kiel

Training data (view on seven days only)



Applications of GMM in OC systems (13)

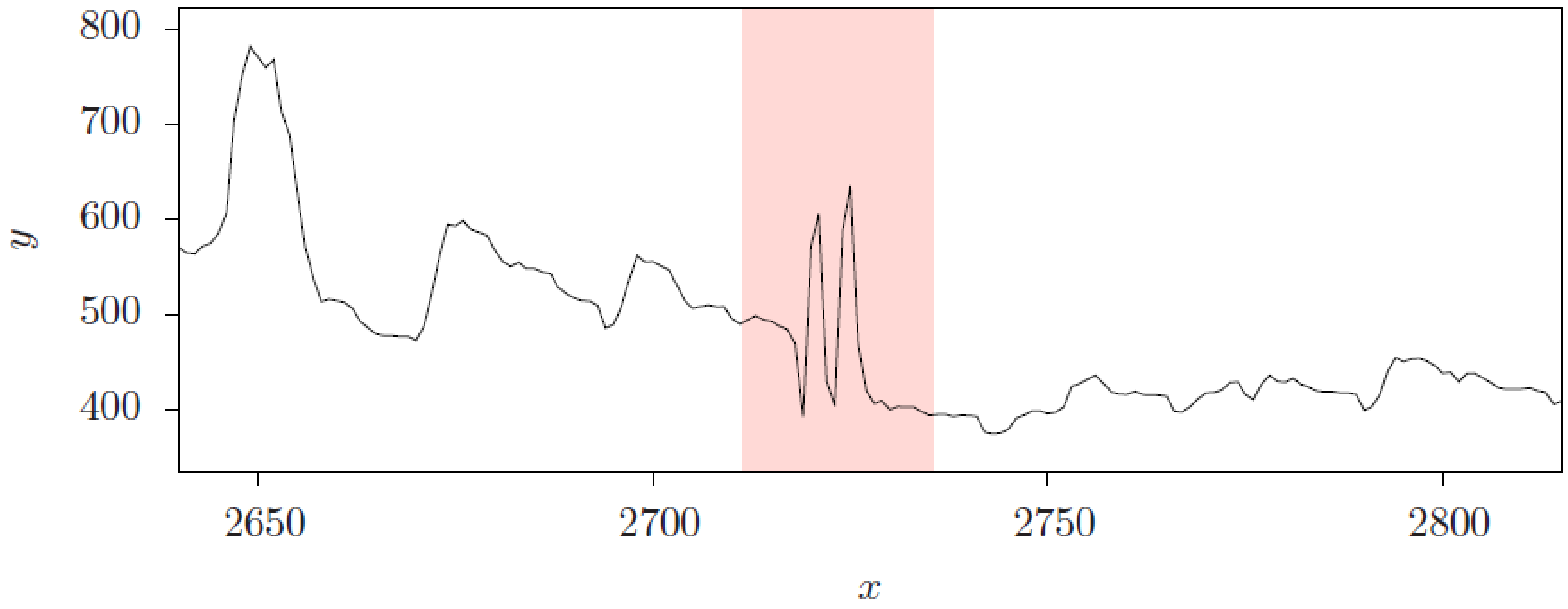
Test data



Applications of GMM in OC systems (14)

Christian-Albrechts-Universität zu Kiel

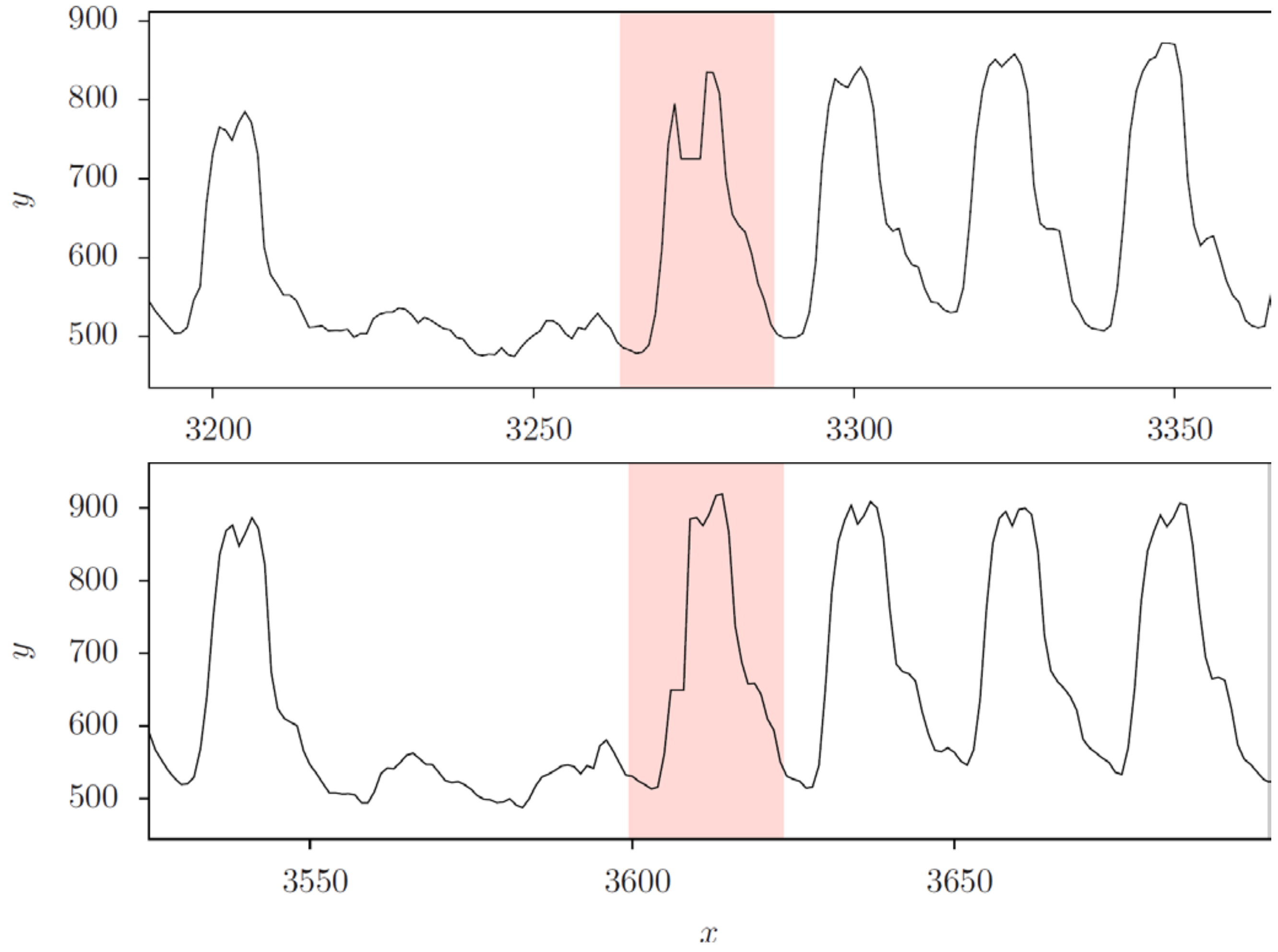
Detected anomaly



Applications of GMM in OC systems (15)

Christian-Albrechts-Universität zu Kiel

Further anomalies

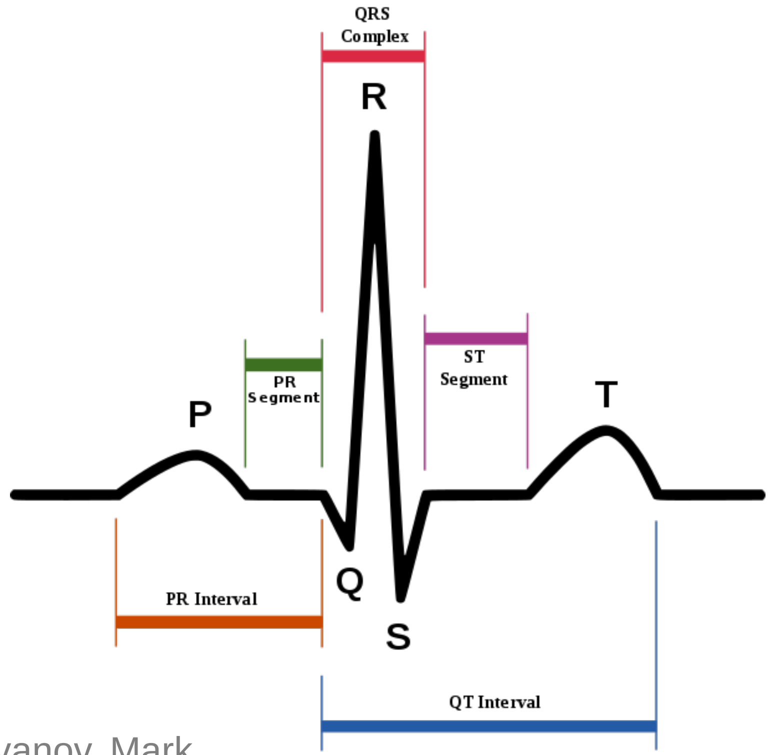


Applications of GMM in OC systems (16)

Christian-Albrechts-Universität zu Kiel

Example 2: Search for anomalies in ECG time series

- Challenges:
  - No fixed length
  - Several (harmless) disturbances
    → Caused by movements of
    the person



Source: Goldberger, Amaral, Glass, Hausdorff, Jeffrey, Ivanov, Mark, Mietus, Moody, Peng, Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals, 2000

Applications of GMM in OC systems (17)

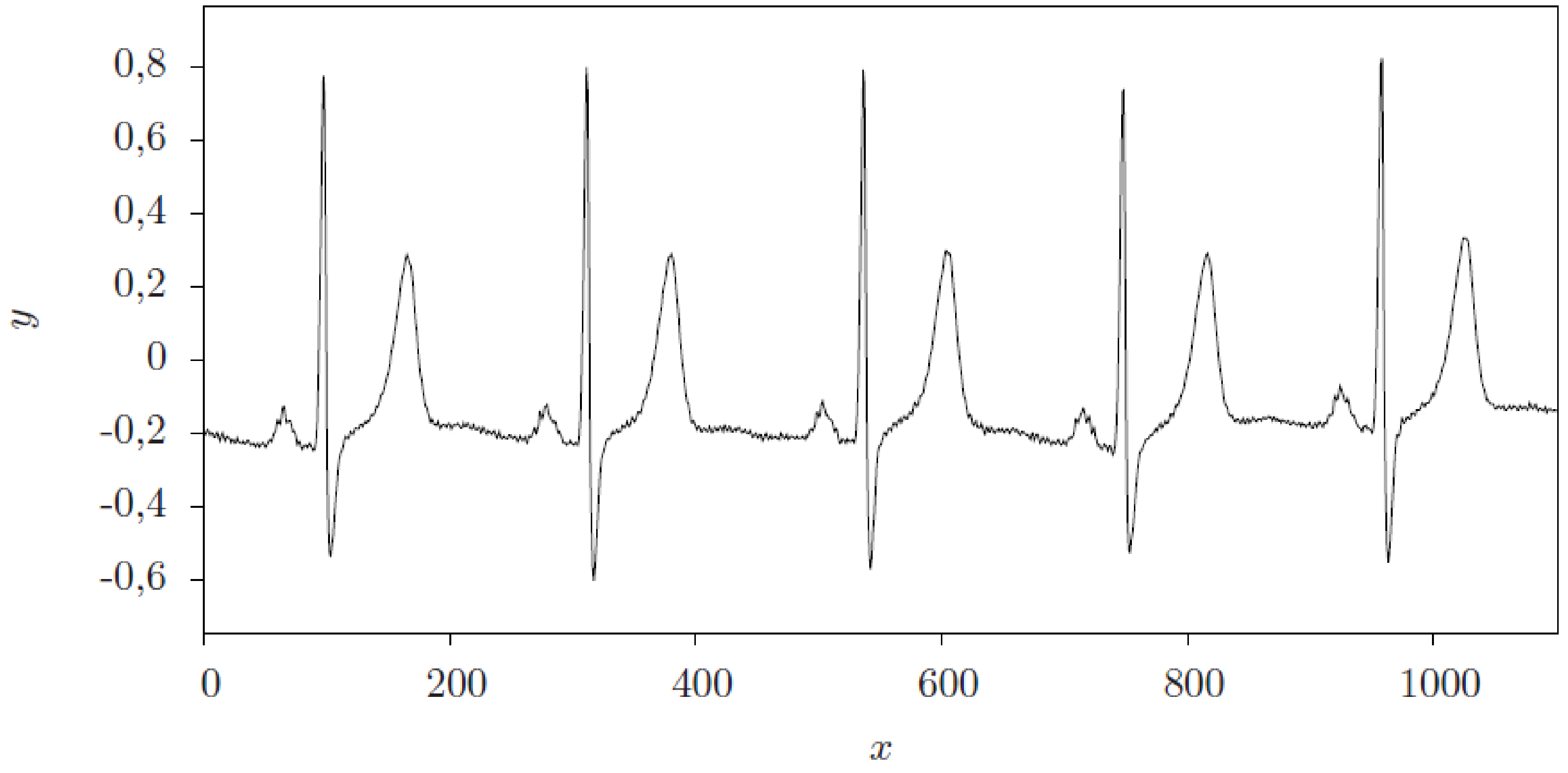
Example 2: Approach

- Consistent segmentation of time series
  - Segmentation into one piece is trivial
  - Segmentation into several pieces requires more sophisticated approaches (and a search for deviations)
- Determining models for each piece/segment of the time series
- Classification of each piece/segment of the time series
- Detection of deviations from the “normal“ pattern of the time series (i.e. the expected sequence of P-Q-R-S-T)
- Problem: Each misclassification results in an anomaly
- Solution: Only accept strong deviations as anomalies (e.g. more than eight mistakes in ten subsequent segments)

Applications of GMM in OC systems (18)

Christian-Albrechts-Universität zu Kiel

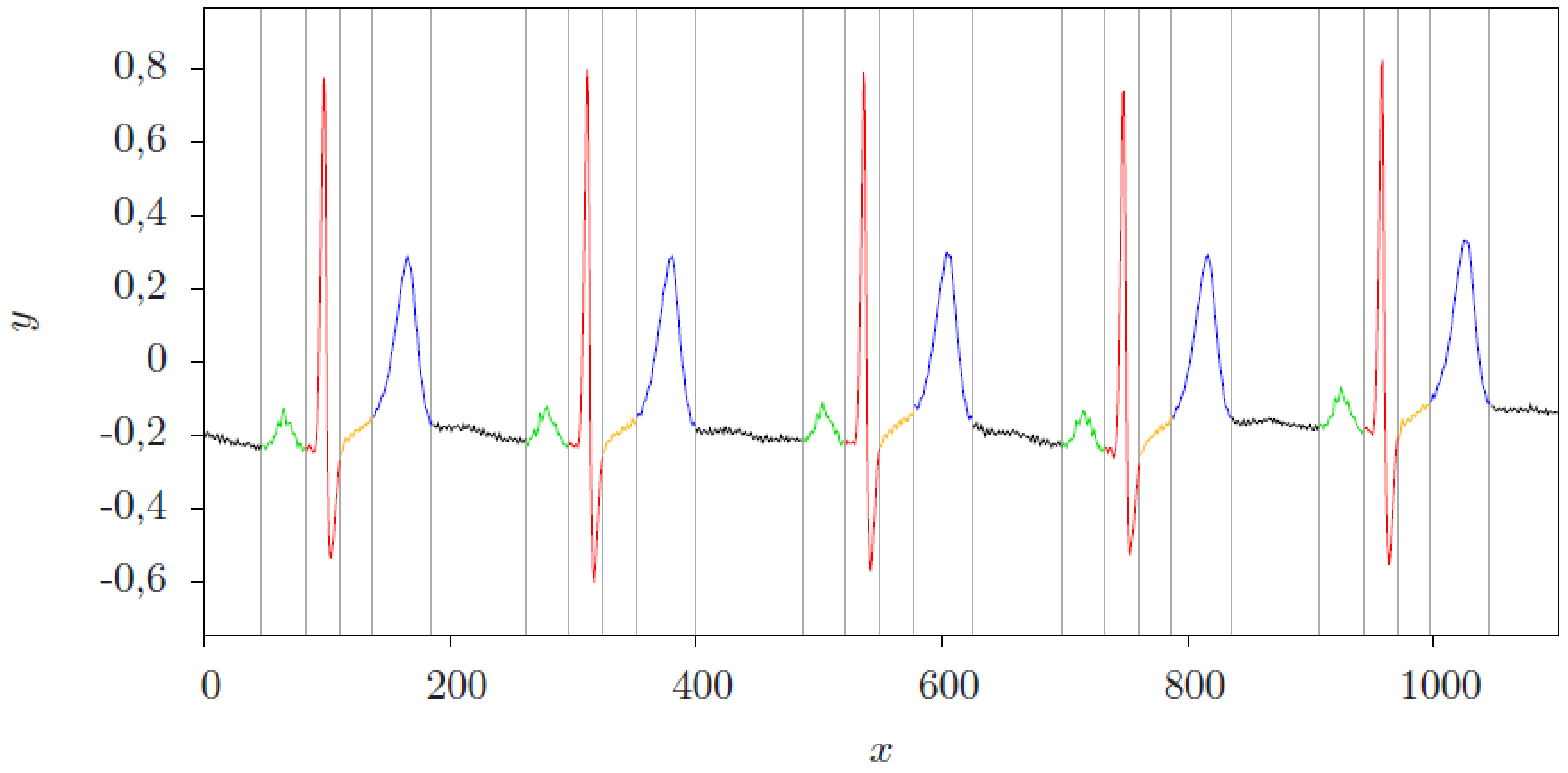
Part of the training data



Applications of GMM in OC systems (19)

Christian-Albrechts-Universität zu Kiel

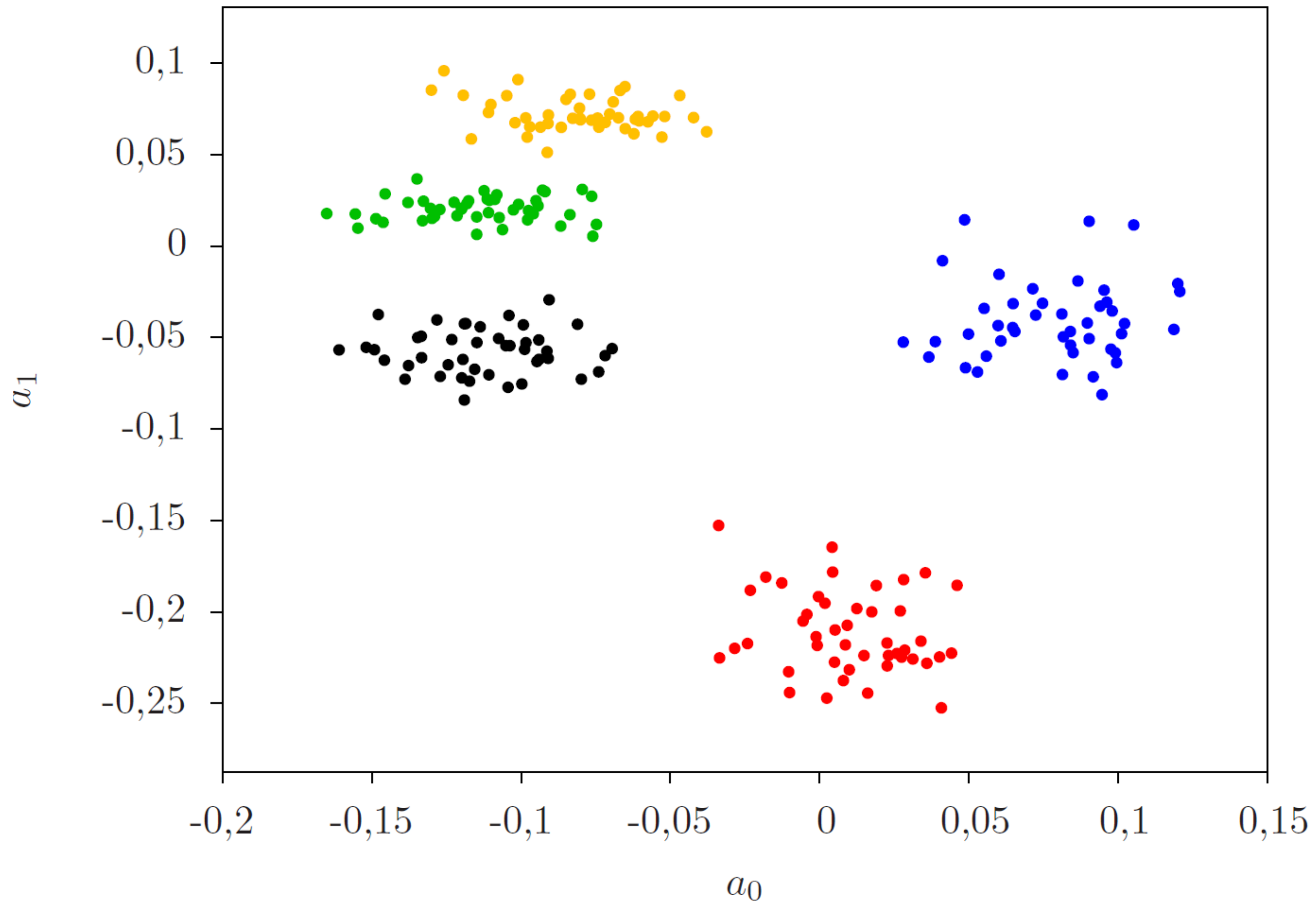
Result after segmentation



Applications of GMM in OC systems (20)

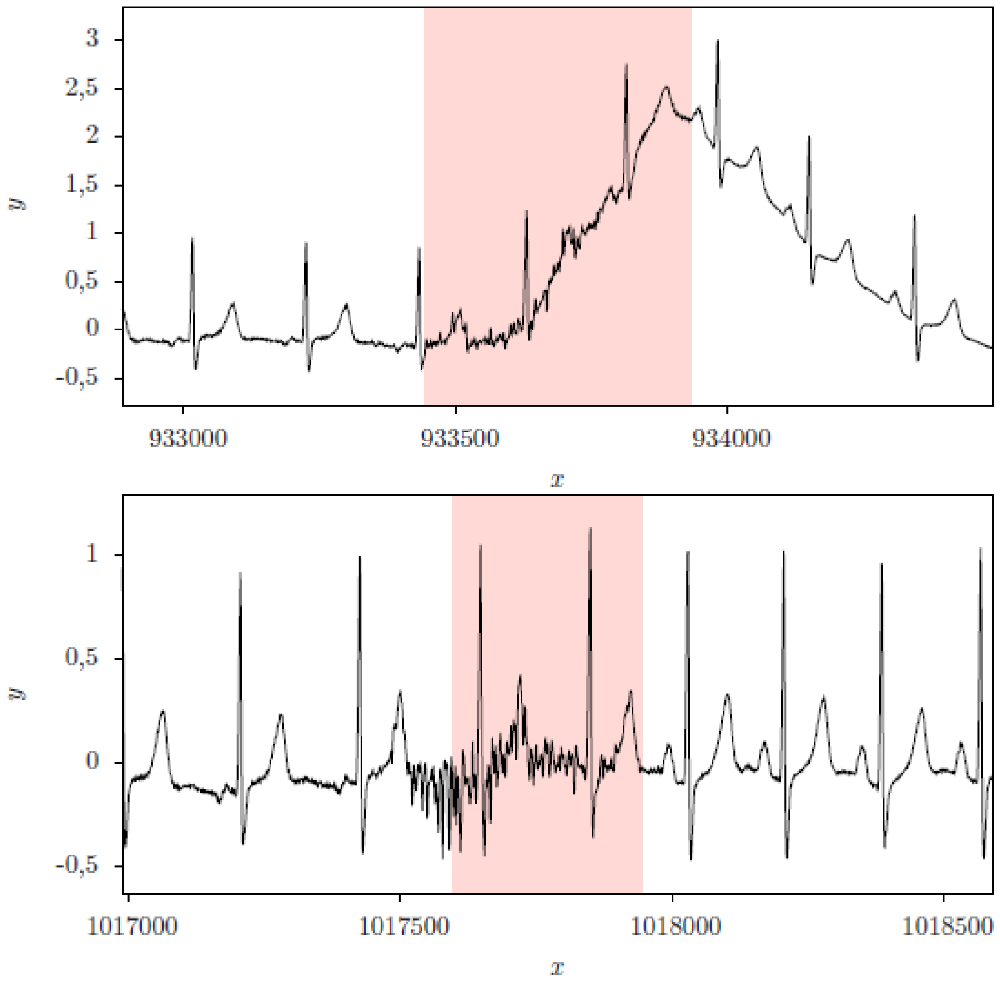
Christian-Albrechts-Universität zu Kiel

Clustering of segments



Applications of GMM in OC systems (21)

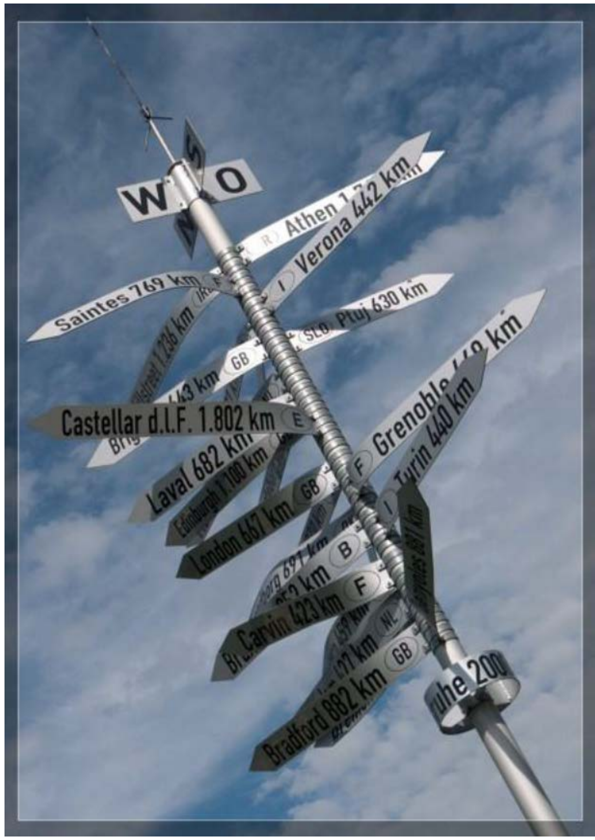
Detected anomalies



Agenda

Christian-Albrechts-Universität zu Kiel

- Generative Modelling
- Gaussian Mixture Models
- Training Gaussian Mixture Models
- Applications in intelligent systems
- Conclusion and further readings



Conclusion

Summary

- Students should be able to:
  - Define the terms of machine learning and model learning
  - Explain how generative modelling works
  - Understand the concept of Gaussian Mixture Models and define all contained parts
  - Describe how a GMM is trained using k-means and Expectation Maximisation
  - Explain for which tasks GMMs can be used in OC systems
  - In particular, identify classification tasks for observed data and define the concept of anomaly detection

References

Christian-Albrechts-Universität zu Kiel

- Goldberger, Amaral, Glass, Hausdorff, Jeffrey, Ivanov, Mark, Mietus, Moody, Peng, Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals, 2000
- Prechelt, PROBEN 1 - a set of benchmarks and benchmarking rules for neural network training algorithms, 1994
- Mitchell, Tom M. "Machine learning. 1997." Burr Ridge, IL: McGraw Hill 45.37 (1997): 870-877

End of chapter

Christian-Albrechts-Universität zu Kiel

- Any questions ...?