Priors and Latent Variables

DD2421

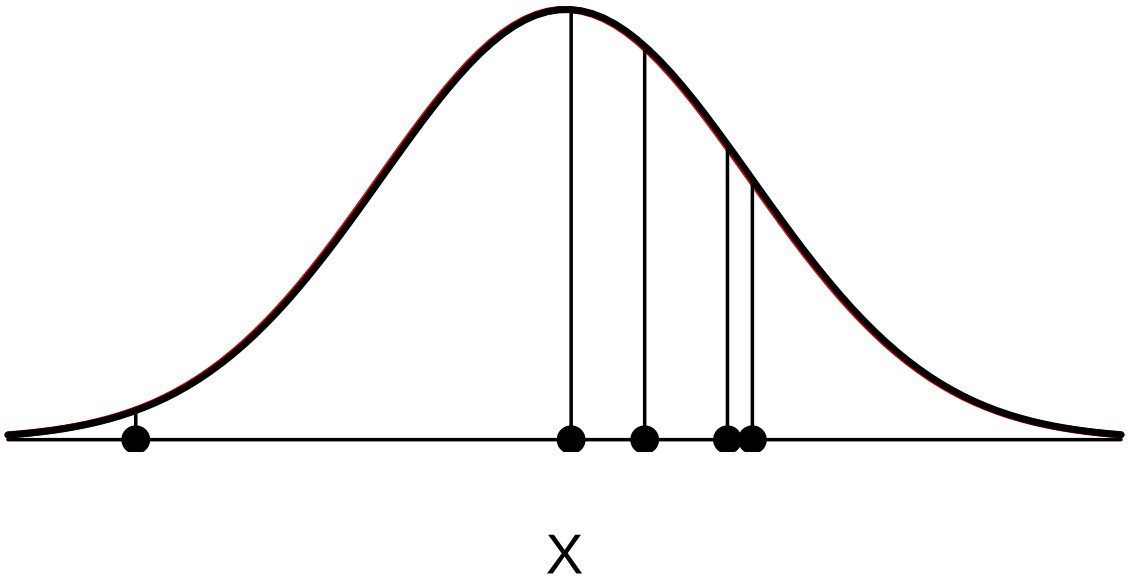
Giampiero Salvi

HT2018

Outline

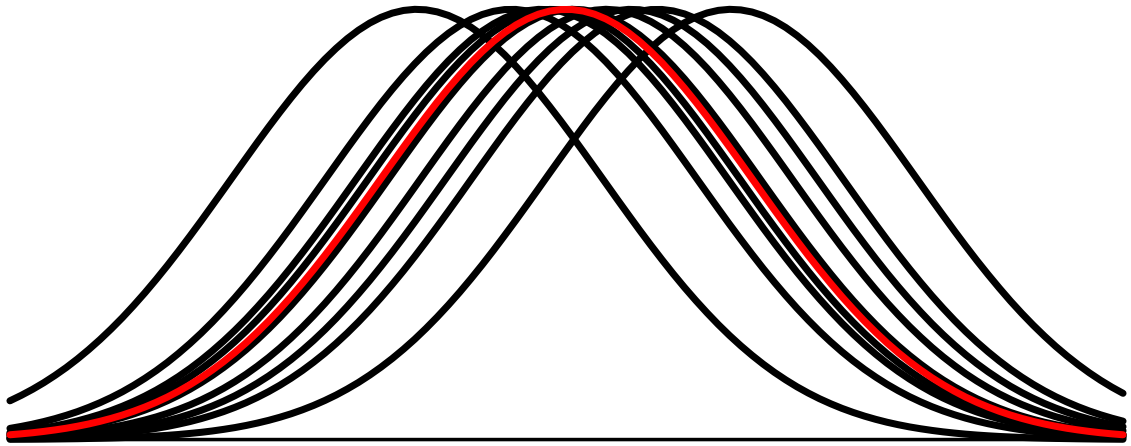
Problem: few data points

10 repetitions with 5 points each



Problem: few data points

10 repetitions with 5 points each



X

Maximum a Posteriori Estimation

$$
\begin{aligned}
\theta_{\text{MAP}} &= \arg\max_{\theta} P(\theta|\mathcal{D}) \\
&= \arg\max_{\theta} \frac{P(\theta)P(\mathcal{D}|\theta)}{P(\mathcal{D})} \\
&= \arg\max_{\theta} P(\theta)P(\mathcal{D}|\theta) \\
&= \arg\max_{\theta} \left[P(\theta) \prod_{i=1}^{N} P(x_i|\theta) \right] \\
&= \arg\max_{\theta} \left[\log P(\theta) + \sum_{i=1}^{N} \log P(x_i|\theta) \right]
\end{aligned}
$$

- $\log P(\theta)$ works as regularization

MAP for Linear Regression

Model (deterministic):

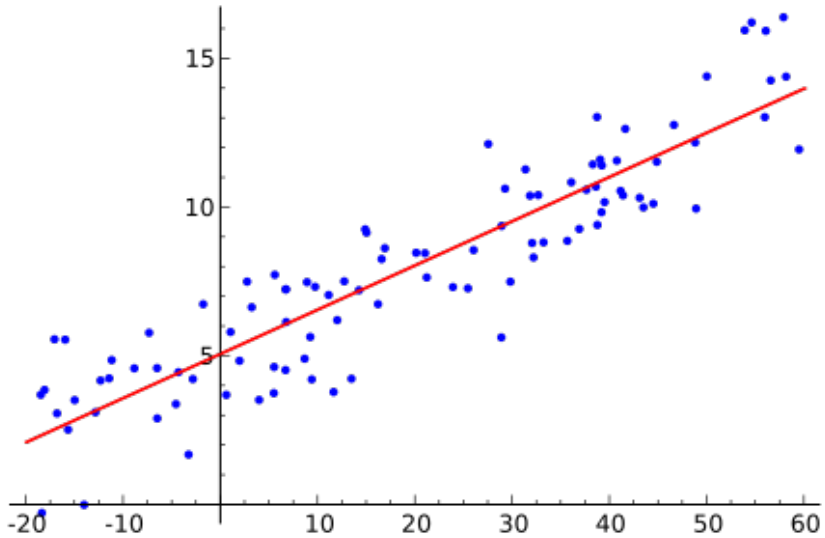
$$y = \mathbf{w}^T \mathbf{x} + \epsilon$$

With:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Therefore:

$$\begin{aligned} \text{💬} &\sim \mathcal{N}(\mu_Y(\mathbf{x}), \sigma_Y^2(\mathbf{x})) \\ &= \mathcal{N}(\mathbf{w}^T \mathbf{x}, \sigma^2) \end{aligned}$$



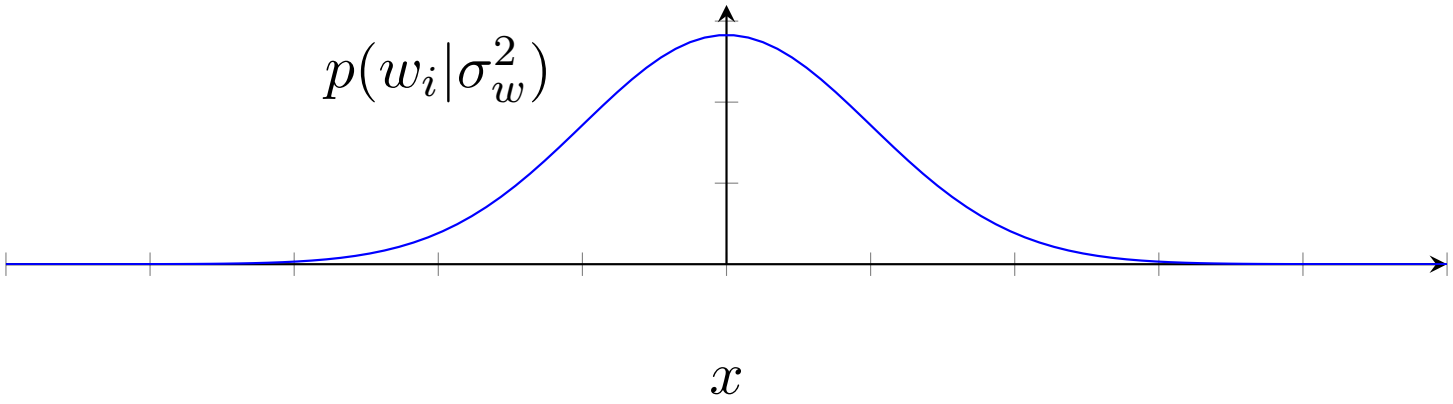
But now we define the a priori probability of $\mathbf{w}$: $P(\mathbf{w})$

Example: zero-mean spherical Gaussian prior

Example: zero-mean spherical Gaussian on $\mathbf{w} = [w_0, \ldots, w_D]$

$$p(\mathbf{w}|\sigma_w^2) = \mathcal{N}(0, \sigma_w^2 \mathbf{I}_D) = \frac{1}{(2\pi\sigma_w^2)^{\frac{D}{2}}} \exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2\sigma_w^2}\right) =$$

$$= \prod_{i=1}^{D} \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{w_i^2}{2\sigma_w^2}\right)$$



Example: zero-mean spherical Gaussian prior

Example: zero-mean spherical Gaussian on $\mathbf{w} = [w_0, \ldots, w_D]$

$$p(\mathbf{w}|\sigma_w^2) = \mathcal{N}(0, \sigma_w^2 \mathbf{I}_D) = \frac{1}{(2\pi\sigma_w^2)^{\frac{D}{2}}} \exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2\sigma_w^2}\right)$$

Instead of $\log p(Y|X, w)$ as in MLE, we optimize $\log p(\mathbf{w}|Y, X)$:

$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} \log p(\mathbf{w}|Y, X) = \arg\max_{\mathbf{w}} \log \left[p(Y|X, \mathbf{w})p(\mathbf{w})\right]$$

$$\ldots = \arg\max_{\mathbf{w}} \sum_n \log p(y_n|\mathbf{x}_n, \mathbf{w}) + \log p(\mathbf{w}) =$$

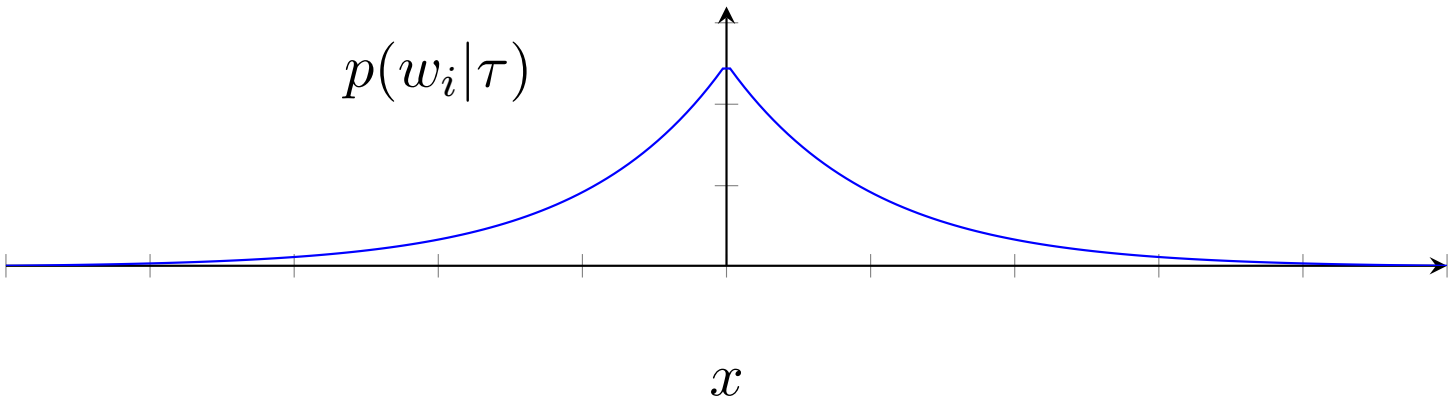
$$\ldots = \arg\min_{\mathbf{w}} \underbrace{\sum_n \left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2}_{\text{fit to the data (ML)}} + \underbrace{\frac{\sigma^2}{\sigma_w^2}\mathbf{w}^T\mathbf{w}}_{\text{keep } \mathbf{w} \text{ simple}}$$

Equivalent to ridge regression with $\lambda = \frac{\sigma^2}{\sigma_w^2}$

Example: Prior for LASSO

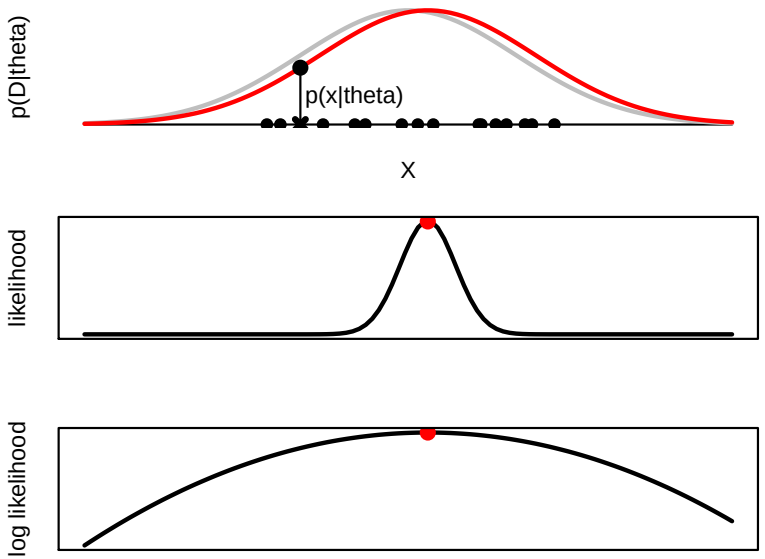
- LASSO: Least Absolute Shrinkage and Selection Operator
- We want the regularization to be $\lambda \sum_i |w_i|$ instead of $\lambda \sum_i w_i^2$.
- Following the same arguments as before, we will need a product of zero-mean Laplace priors:

$$p(\mathbf{w}|\tau) = \prod_i \text{Laplace}(w_i, 0, \tau) = \prod_i \frac{1}{2\tau} \exp\left(-\frac{|w_i|}{\tau}\right)$$



ML, MAP and Point Estimates

- Both ML and MAP produce point estimates of θ
- Assumption: there is a true value for θ
- advantage: once $\hat{\theta}$ is found, everything is known



Limitations (Linear Regression)

- shift problem to defining the parameters of the prior (λ in Ridge and LASSO regression)
- uncertainty in the posterior $p(y|\mathbf{x}, \mathbf{w}_{\text{OPT}})$ is still σ^2 and is independent of $\mathbf{x}$

Bayesian estimation (non-parametric models)

$$\begin{array}{lccccc} \text{ML:} & \mathcal{D} & \rightarrow & \theta_{\text{ML}} & \rightarrow & P(\mathbf{x}_{\text{new}}|\theta_{\text{ML}}) \\ \text{MAP:} & \mathcal{D}, P(\theta) & \rightarrow & \theta_{\text{MAP}} & \rightarrow & P(\mathbf{x}_{\text{new}}|\theta_{\text{MAP}}) \\ \text{Bayes:} & \mathcal{D}, P(\theta) & \rightarrow & P(\theta|\mathcal{D}) & \rightarrow & P(\mathbf{x}_{\text{new}}|\mathcal{D}) \end{array}$$

1. consider θ as a random variable (same as MAP)
2. characterize θ with the posterior distribution $P(\theta|\mathcal{D})$ given the data
3. compute new evidence marginalizing over θ (predictive posterior)

$$P(\mathbf{x}_{\text{new}}|\mathcal{D}) = \int_{\theta \in \Theta} P(\mathbf{x}_{\text{new}}|\theta) P(\theta|\mathcal{D}) d\theta$$

note that we can also vary the number of parameters (model complexity)

Bayesian Linear Regression

$$P(y_{\text{new}}|\mathbf{x}_{\text{new}}, Y, X) = \int_{\mathbf{w}\in W} P(y_{\text{new}}|\mathbf{x}_{\text{new}}, \mathbf{w})P(\mathbf{w}|Y, X)d\mathbf{w}$$

- if prior on $\mathbf{w}$ is Gaussian, then posterior $P(\mathbf{w}|Y, X)$ is still Gaussian
- because the likelihood $P(y_{\text{new}}|\mathbf{x}_{\text{new}}, \mathbf{w})$ is also Gaussian, the predictive posterior is Gaussian as well.

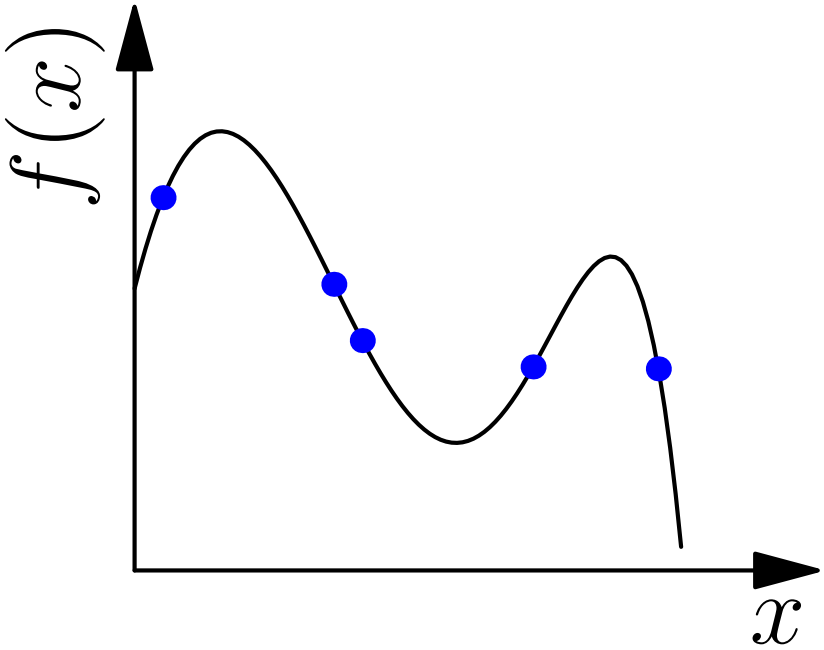
$$P(y_{\text{new}}|\mathbf{x}_{\text{new}}, Y, X) = \mathcal{N}(\mu^T\mathbf{x}_{\text{new}}, \sigma^2 + \mathbf{x}_{\text{new}}^T\mathbf{\Sigma}\mathbf{x}_{\text{new}})$$

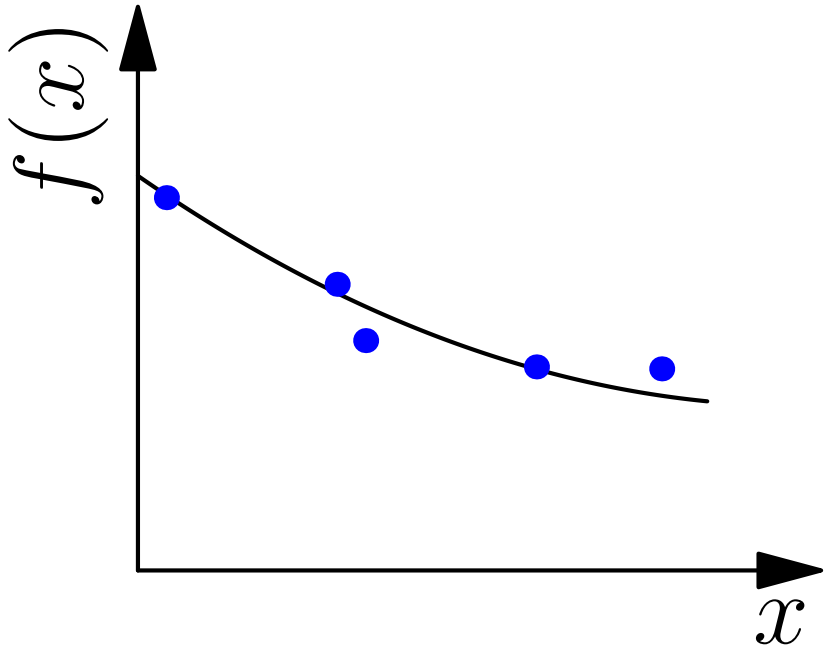
Where μ and $\mathbf{\Sigma}$ are mean and cov. matrix of the posterior $P(\mathbf{w}|Y, X)$

Consequences

- we are considering the uncertainty over the choice of $\mathbf{w}$ as well as the original uncertainty σ^2.
- we have natural means to prevent overfitting

Overfitting





Occam's Razor

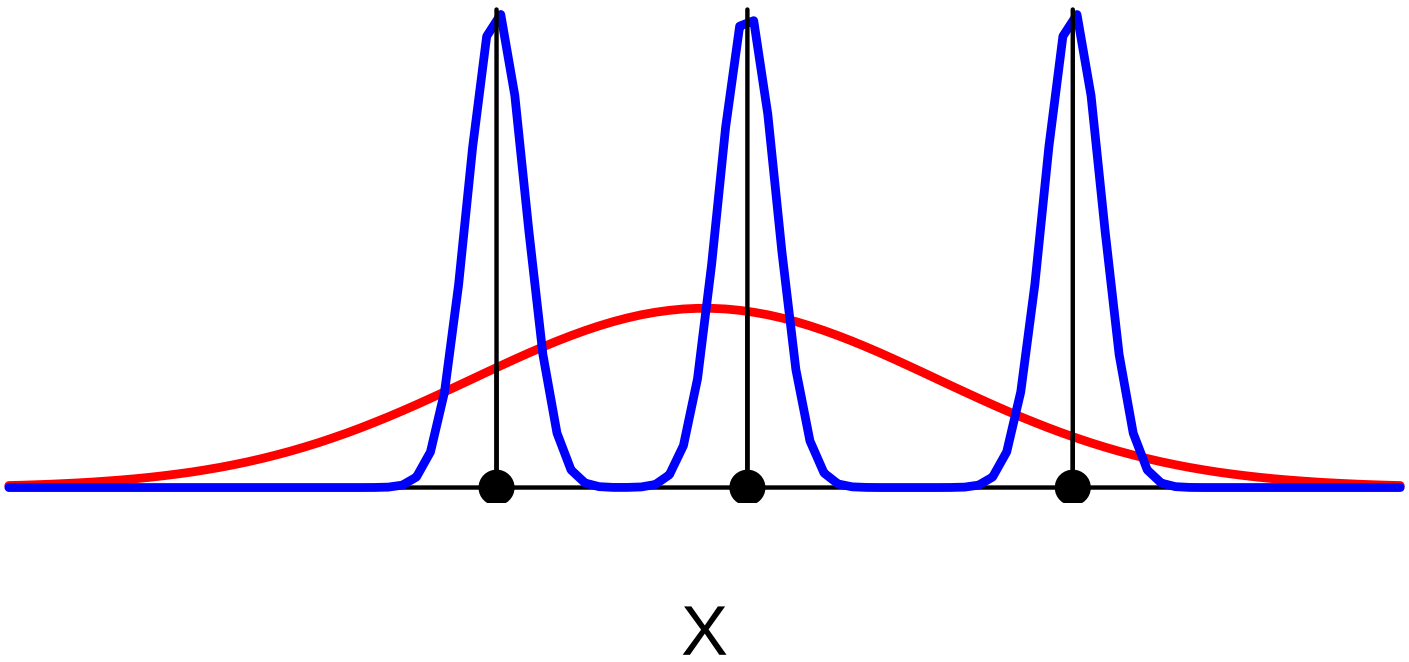
Choose the simplest explanation for the observed data

Important factors:

- number of model parameters
- number of data points
- model fit to the data

Overfitting and Maximum Likelihood

we can make the likelihood arbitrary large by increasing the number of parameters



Occam's Razor and Bayesian Learning

Remember that:

$$P(\mathbf{x}_{\text{new}}|\mathcal{D}) = \int_{\theta \in \Theta} P(\mathbf{x}_{\text{new}}|\theta) P(\theta|\mathcal{D}) d\theta$$

Intuition:

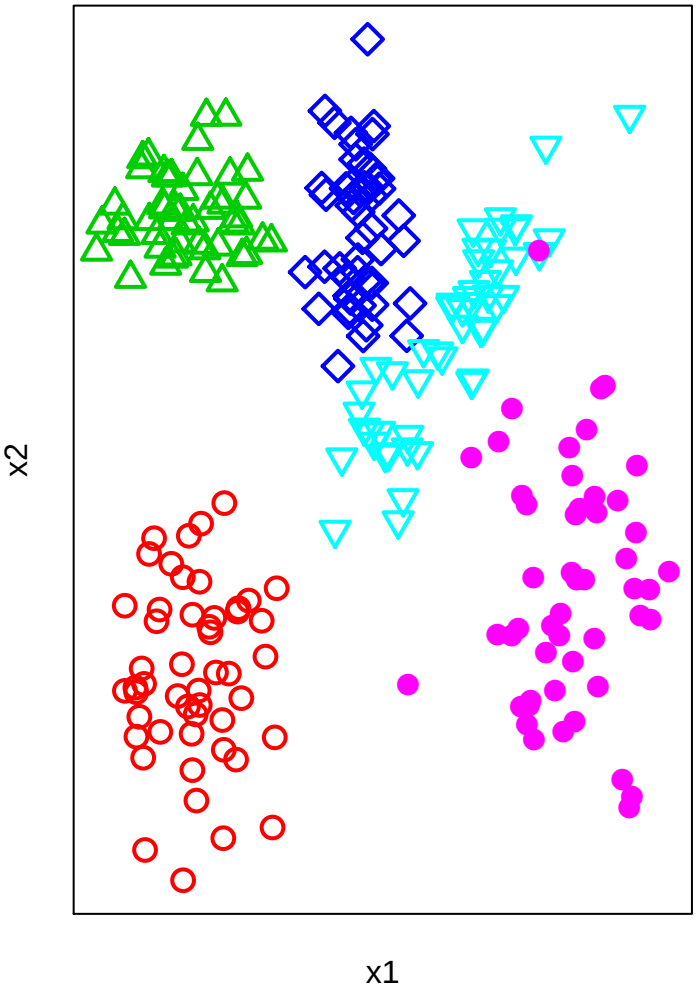
More complex models fit the data very well (large $P(\mathcal{D}|\theta)$ and $P(\theta|\mathcal{D})$ but only for small regions of the parameter space Θ.

Limitations

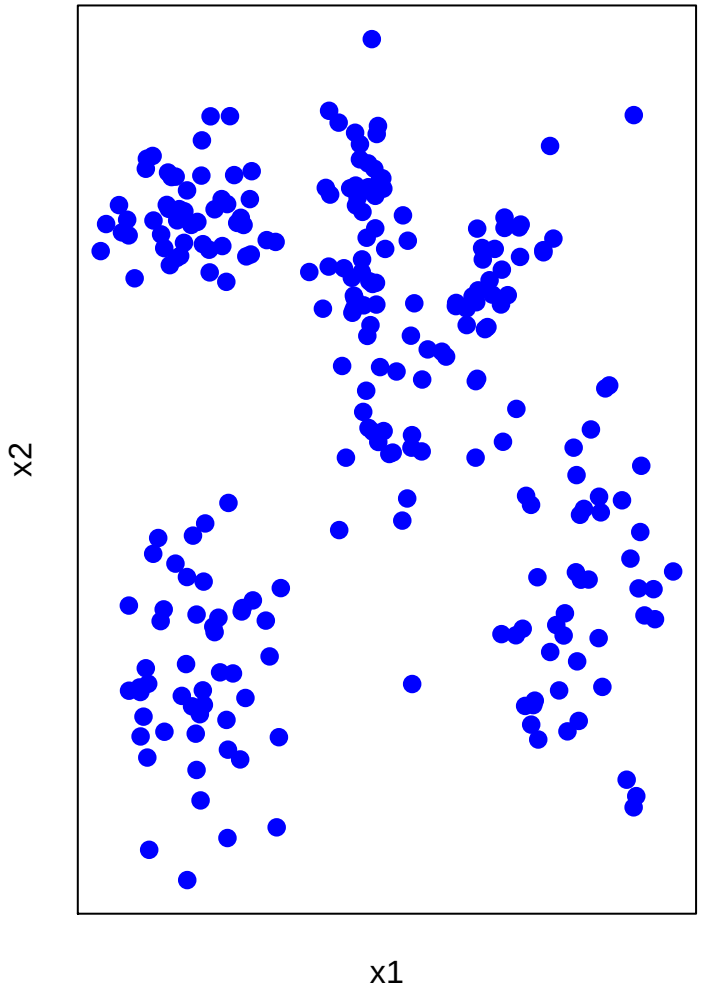
- not always possible to compute posterior (conjugate priors)
- approximations with high computational cost (sampling methods) or complex solutions (variational methods)
- sometime we want to have non-informative priors
- for unbounded continuous variables this can be difficult

Clustering vs Classification

Classification



Clustering



Fitting complex distributions

We can try to fit a mixture of K distributions:

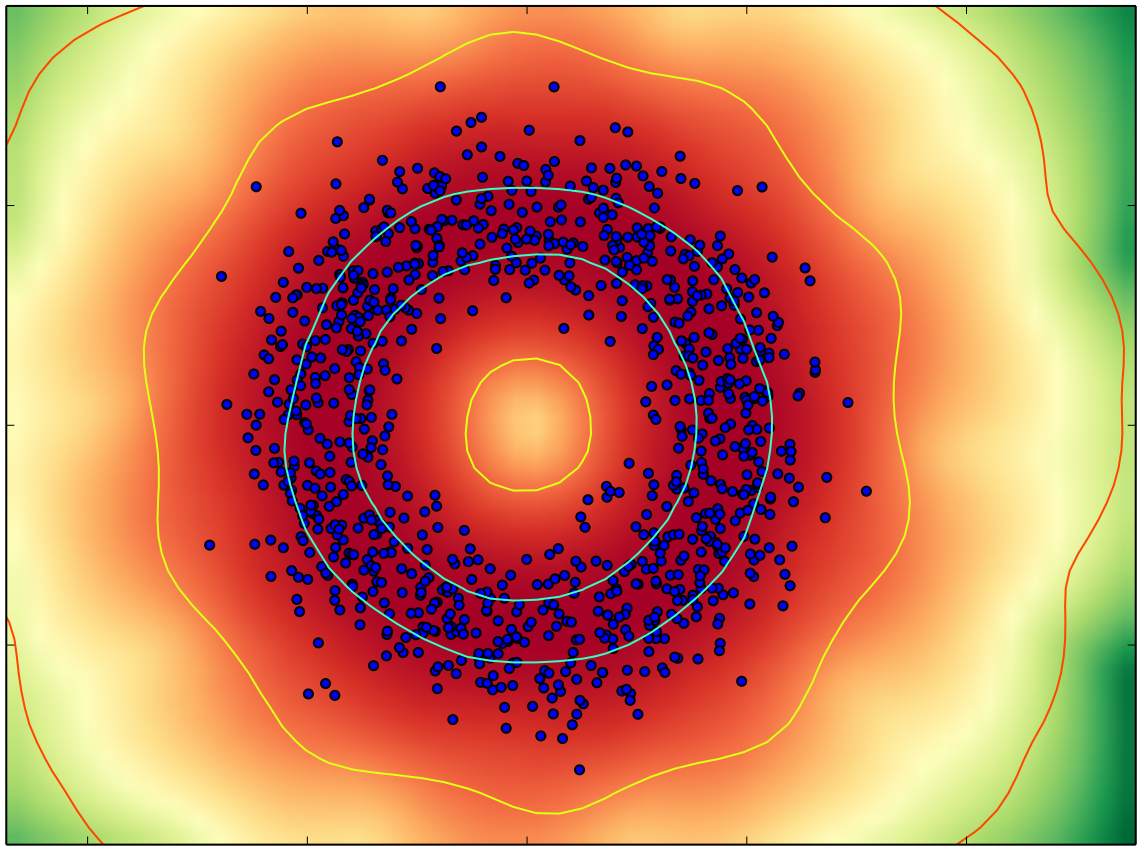
$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

with $\theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$

Example: doughnut data

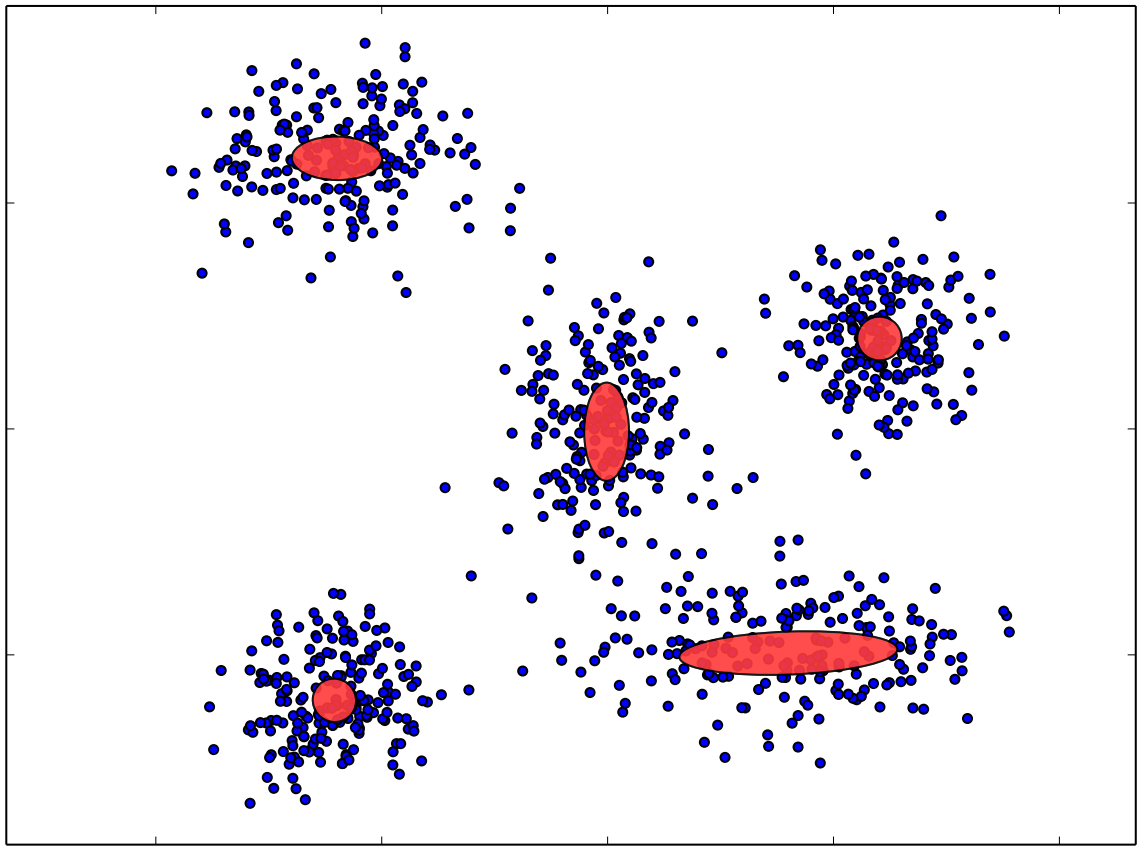
$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$

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



Clustering Example

$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$
$$P(\mathbf{x}|\theta_k), \forall k \in [1, K]$$



Fitting complex distributions

We can try to fit a mixture of K distributions:

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

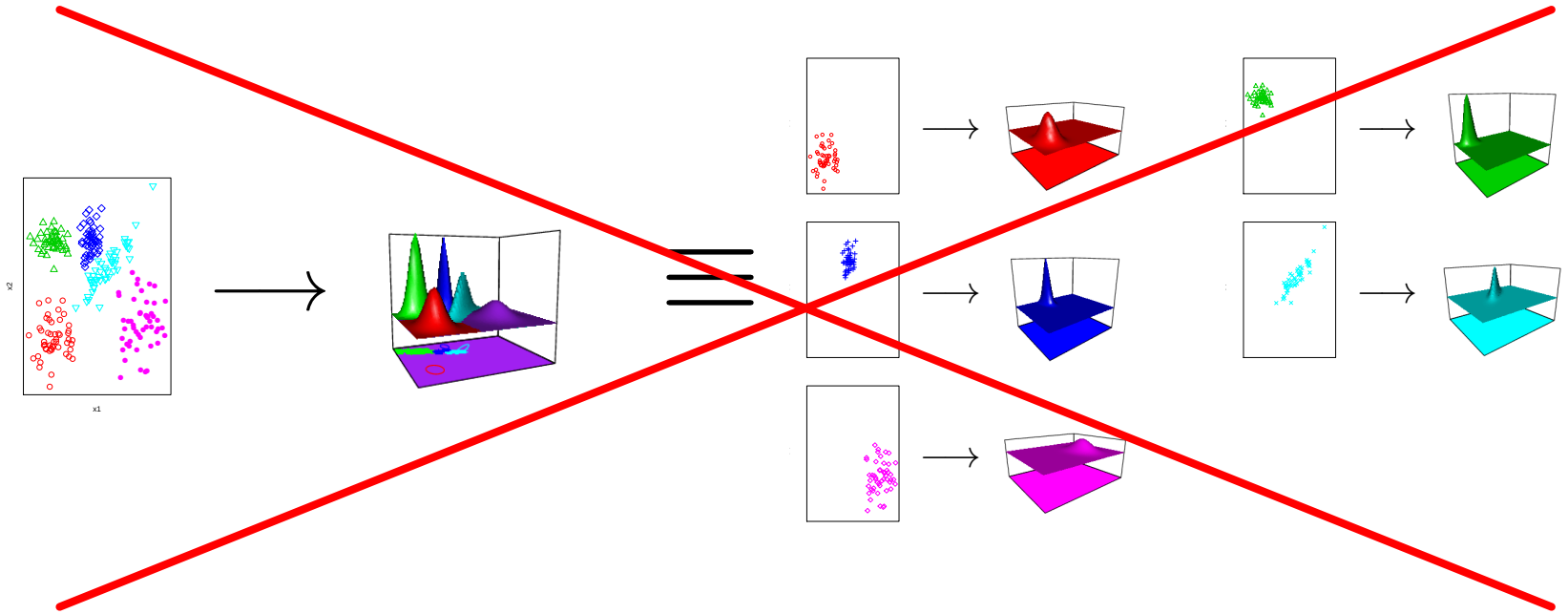
$$\text{with } \theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$$

Problem:

We do not know which point has been generated by which component of the mixture

We cannot optimize $P(\mathbf{x}|\theta)$ directly

No Class Independence Assumption



Solution: Expectation Maximization

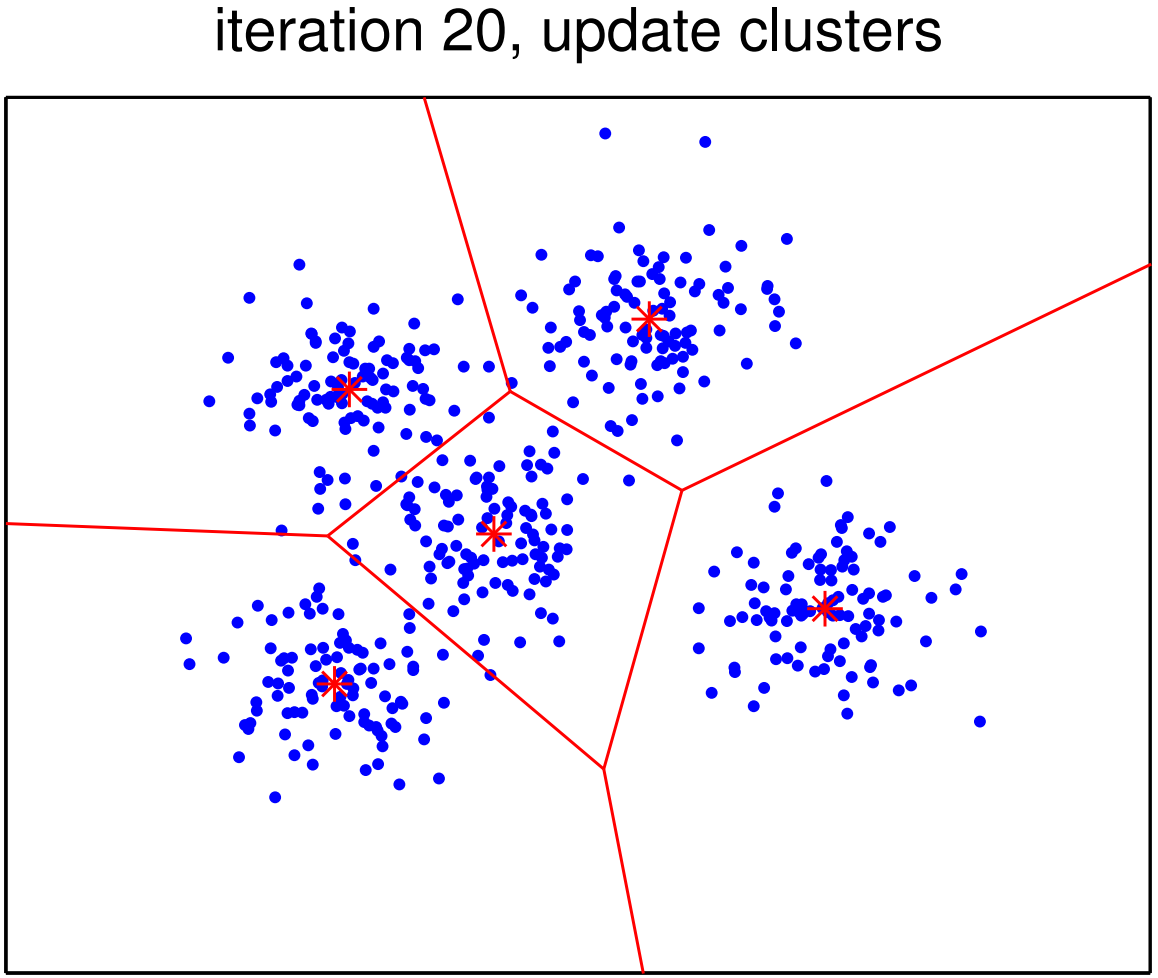
Heuristic Example: K-means

- describes each class with a centroid
- a point belongs to a class if the corresponding centroid is closest (Euclidean distance)
- iterative procedure
- guaranteed to converge
- not guaranteed to find the optimal solution
- used in vector quantization (since the 1950's)

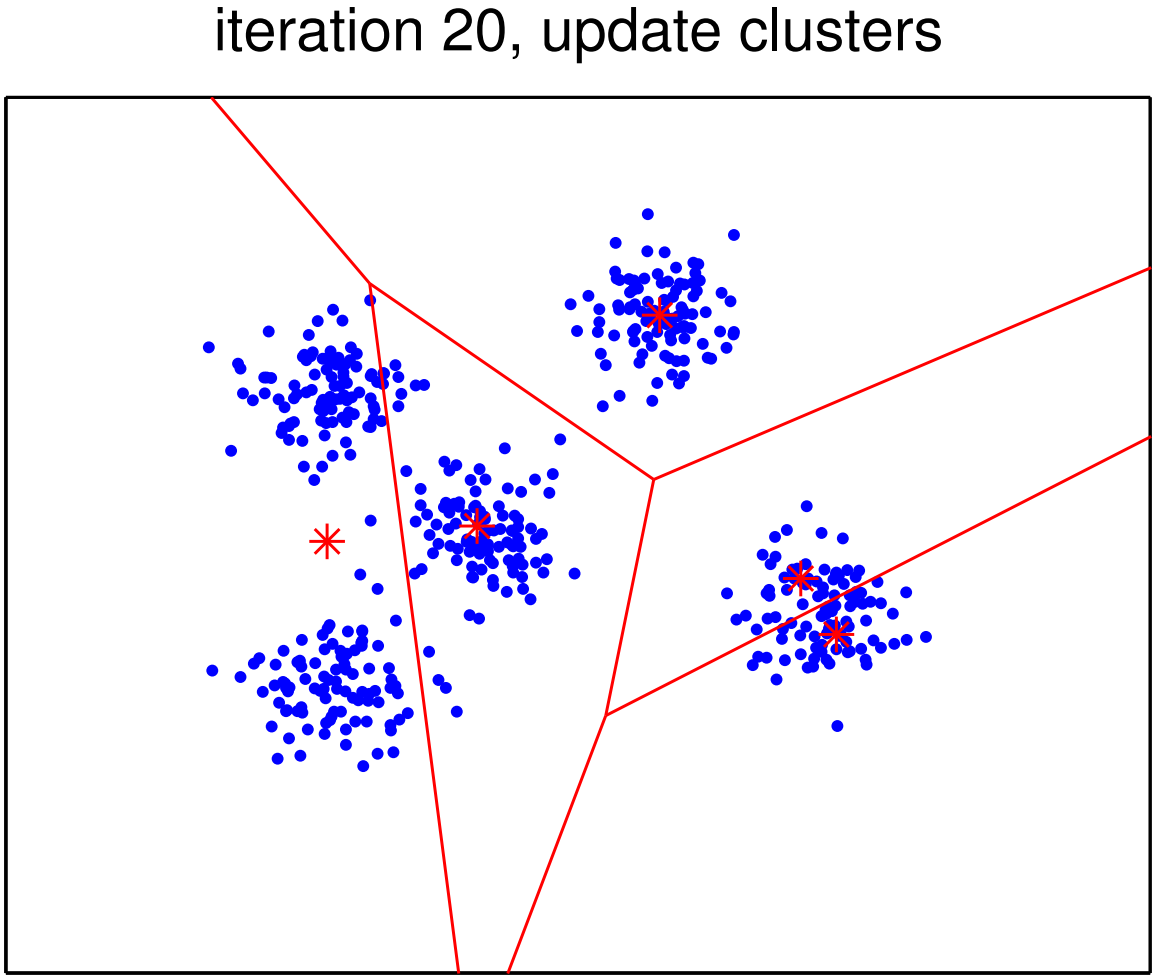
K-means: algorithm

Data: k (number of desired clusters), n data points $\mathbf{x}_i$
Result: k clusters
initialization: assign initial value to k centroids $\mathbf{c}_i$;
repeat
    assign each point $\mathbf{x}_i$ to closest centroid $\mathbf{c}_j$;
    compute new centroids as mean of each group of points;
until *centroids do not change*;
return k clusters;

K-means: example



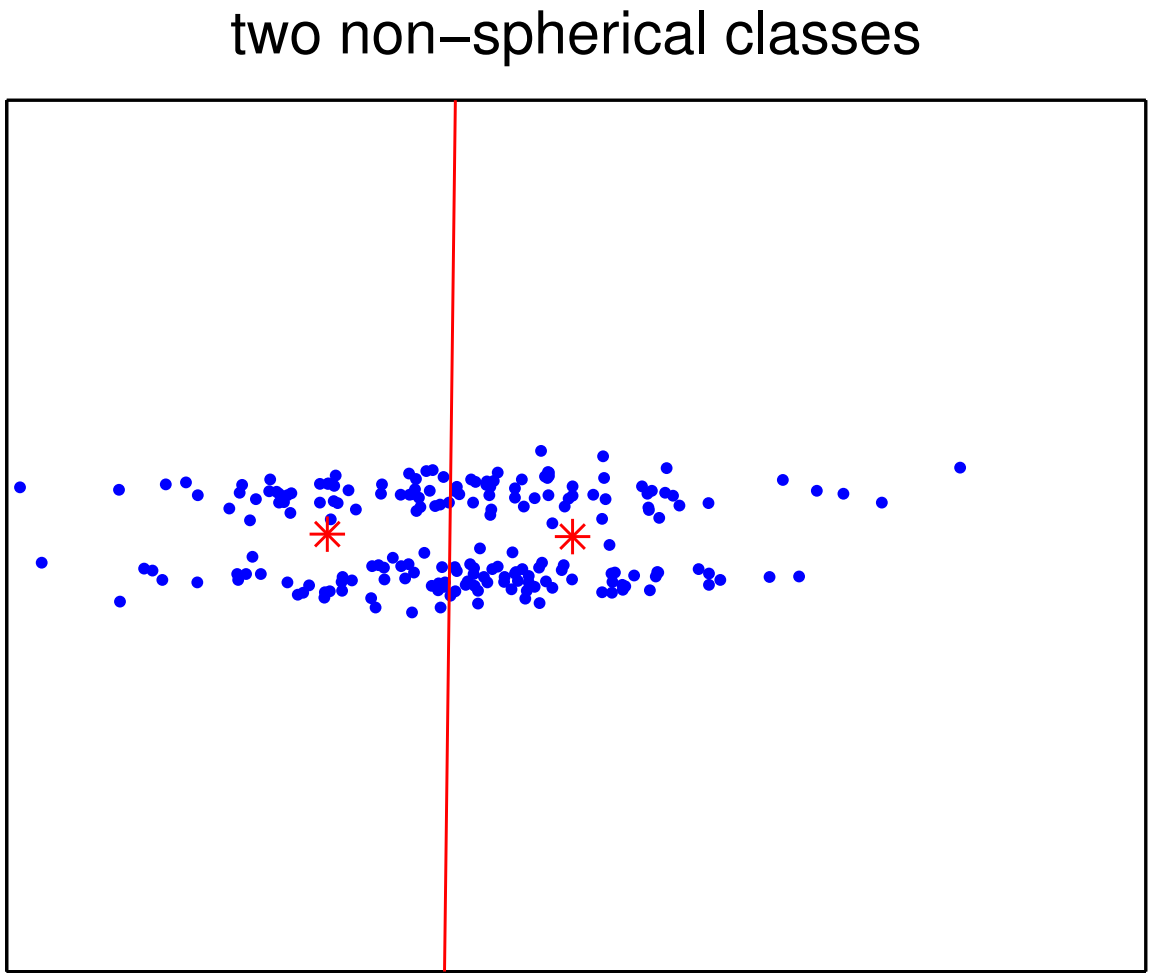
K-means: sensitivity to initial conditions



K-means: limits of Euclidean distance

- the Euclidean distance is isotropic (same in all directions in $\mathbb{R}^p$)
- this favours spherical clusters
- the size of the clusters is controlled by their distance

K-means: non-spherical classes



Expectation Maximization

Fitting model parameters with missing (latent) variables

$$P(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k P(x|\theta_k),$$

$$\text{with } \theta = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_K\}$$

- very general idea (applies to many different probabilistic models)
- augment the data with the latent variables: $h_i \in \{1, \ldots, K\}$ assignment of each data point x_i to a component of the mixture
- optimize the Likelihood of the complete data over N data points

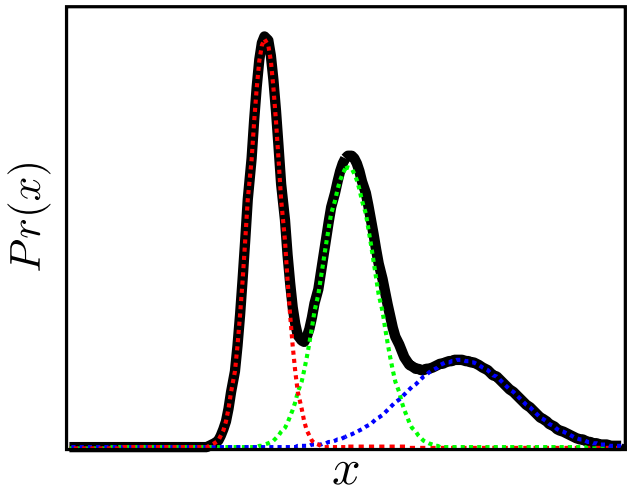
$$P(\mathbf{x}_1, \ldots, \mathbf{x}_N, h_1, \ldots, h_N|\theta)$$

Example: Mixture of Gaussians

This distribution is a weighted sum of K Gaussian distributions

$$P(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$

where $\pi_1 + \cdots + \pi_K = 1$
and $\pi_k > 0$ $(k = 1, \ldots, K)$.



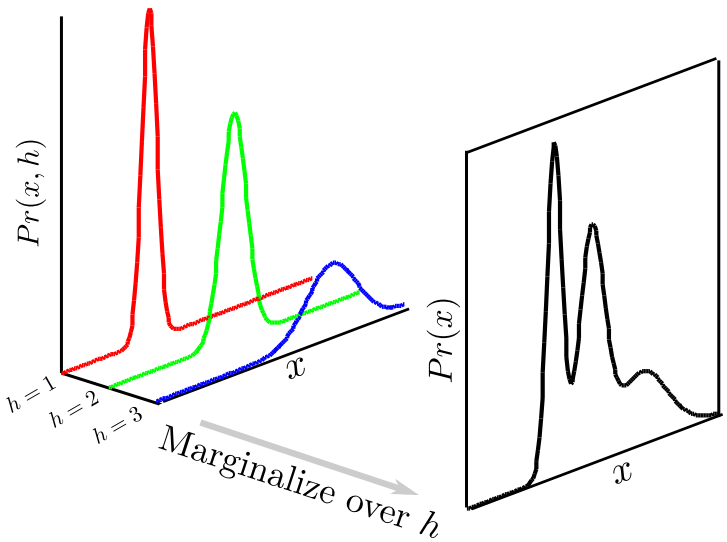
This model can describe **complex multi-modal** probability distributions by combining simpler distributions.

Mixture of Gaussians as a marginalization

We can interpret the Mixture of Gaussians model with the introduction of a discrete hidden/latent variable h and $P(x, h)$:

$$P(x) = \sum_{k=1}^{K} P(x, h = k) = \sum_{k=1}^{K} P(x \mid h = k) P(h = k)$$

$$= \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \sigma_k^2)$$



← **mixture density**

Figures taken from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians

For each sample x_i introduce a *hidden variable* h_i

$$h_i = \begin{cases} 1 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x; \mu_1, \sigma_1^2) \\ 2 & \text{if sample } x_i \text{ was drawn from } \mathcal{N}(x; \mu_2, \sigma_2^2) \end{cases}$$

and come up with initial values

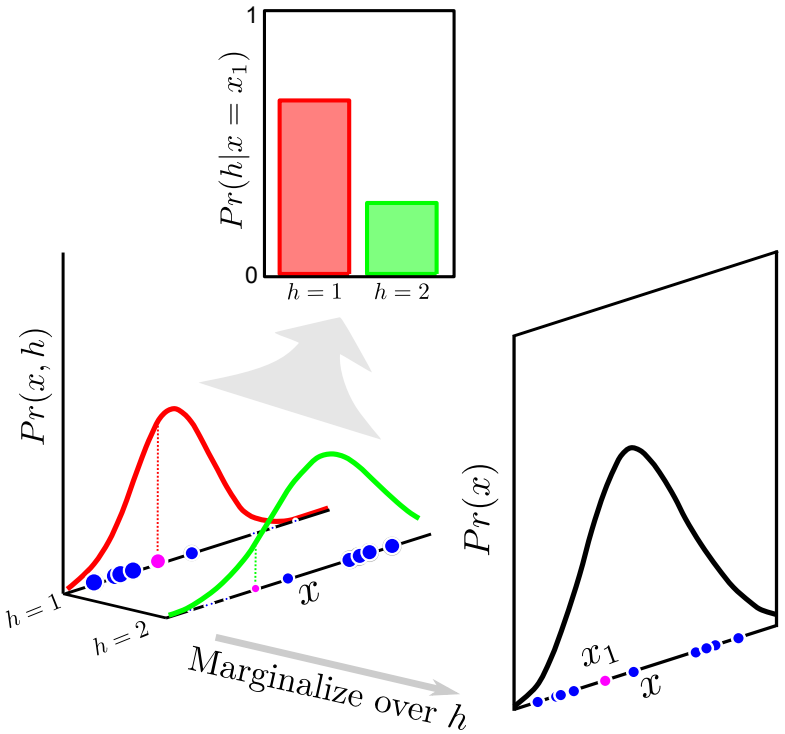
$$\Theta^{(0)} = (\pi_1^{(0)}, \mu_1^{(0)}, \sigma_1^{(0)}, \mu_2^{(0)}, \sigma_2^{(0)})$$

for each of the parameters.

EM is an *iterative algorithm* which updates $\Theta^{(t)}$ using the following two steps...

EM for two Gaussians: E-step

The responsibility of k-th Gaussian for each sample x (indicated by the size of the projected data point)



Look at each sample x along hidden variable h in the E-step

Figure from **Computer Vision: models, learning and inference** by Simon Prince.

EM for two Gaussians: E-step (cont.)

E-step: Compute the *"posterior probability"* that x_i was generated by component k given the current estimate of the parameters $\Theta^{(t)}$. (responsibilities)

for $i = 1, \dots n$

for $k = 1, 2$

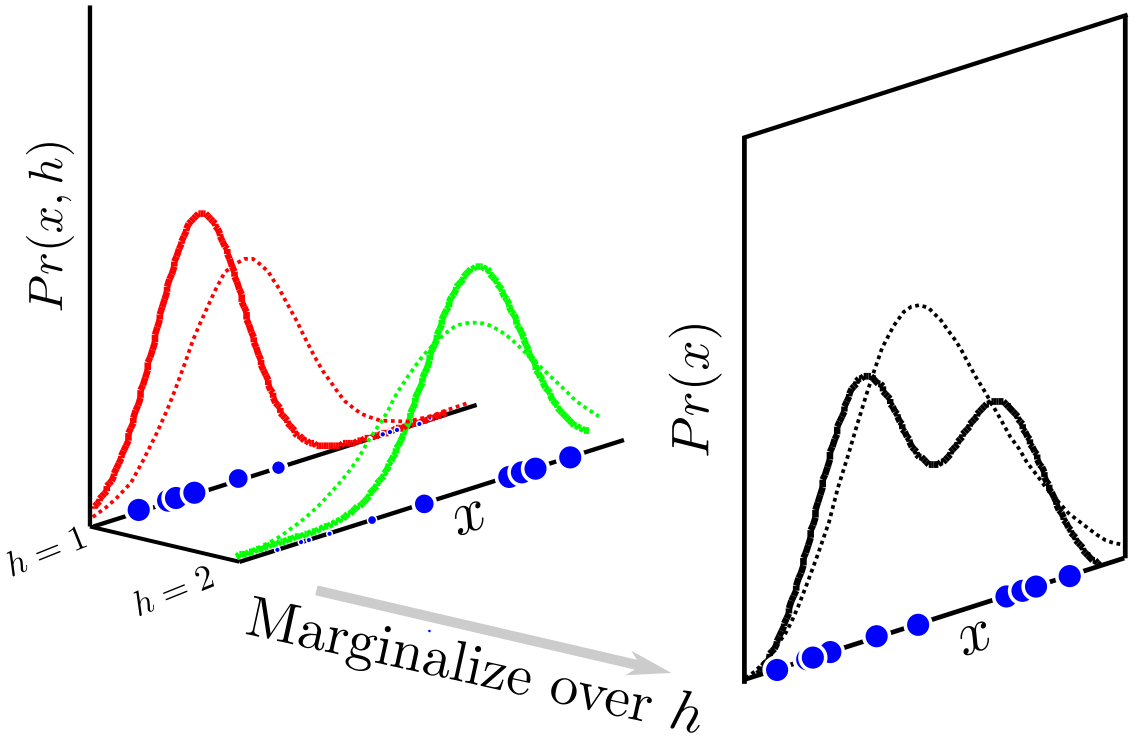
$$\gamma_{ik}^{(t)} = P(h_i = k \,|\, x_i, \Theta^{(t)})$$

$$= \frac{\pi_k^{(t)} \mathcal{N}(x_i; \mu_k^{(t)}, \sigma_k^{(t)})}{\pi_1^{(t)} \mathcal{N}(x_i; \mu_1^{(t)}, \sigma_1^{(t)}) + \pi_2^{(t)} \mathcal{N}(x_i; \mu_2^{(t)}, \sigma_2^{(t)})}$$

Note: $\gamma_{i1}^{(t)} + \gamma_{i2}^{(t)} = 1$ and $\pi_1 + \pi_2 = 1$

EM for two Gaussians: M-step

Fitting the Gaussian model for each of k-th constinuetnt.
Sample x_i contributes according to the responsibility γ_{ik}.



(dashed and solid lines for fit before and after update)

Look along samples x for each h in the M-step

EM for two Gaussians: M-step (cont.)

M-step: Compute the *Maximum Likelihood* of the parameters of the mixture model given out data's membership distribution, the $\gamma_i^{(t)}$'s:

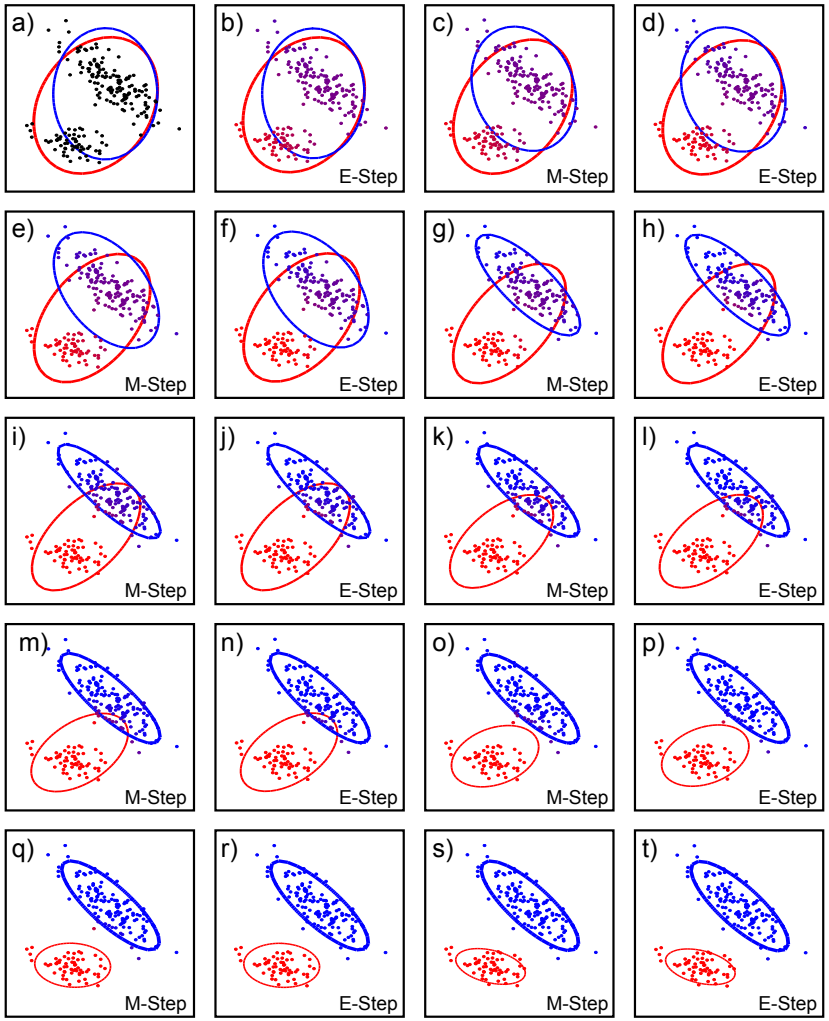
for $k = 1, 2$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)} x_i}{\sum_{i=1}^n \gamma_{ik}^{(t)}},$$

$$\sigma_k^{(t+1)} = \sqrt{\frac{\sum_{i=1}^n \gamma_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^n \gamma_{ik}^{(t)}}},$$

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^n \gamma_{ik}^{(t)}}{n}.$$

EM in practice



EM properties

Similar to K-means

- guaranteed to find a local maximum of the complete data likelihood
- somewhat sensitive to initial conditions

Better than K-means

- Gaussian distributions can model clusters with different shapes
- all data points are smoothly used to update all parameters

Summary

1. Incorporating Priors
   - Maximum a Posteriori Estimation
   - Bayesian Non-Parametric Methods
   - Model Selection and Occam's Razor
2. Unsupervised Learning
   - Classification vs Clustering
   - Heuristic Example: K-means
   - Expectation Maximization

Don't forget to fill the form at
`https://goo.gl/forms/uqo0u9ppUMkhDnUe2`
The link is also in Canvas.