



# **7. Mixture Models and Expectation-Maximization**

# Motivation

- Often the introduction of latent (unobserved) random variables into a model can help to express complex (marginal) distributions
- A very common example are **mixture models**, in particular Gaussian mixture models (GMM)
- Mixture models can be used for clustering (unsupervised learning) and to express more complex probability distributions
- As we will see, the parameters of mixture models can be estimated using maximum-likelihood estimation such as expectation-maximization



# K-means Clustering

- Given: data set  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , number of clusters  $K$
- Goal: find cluster centers  $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$  so that

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

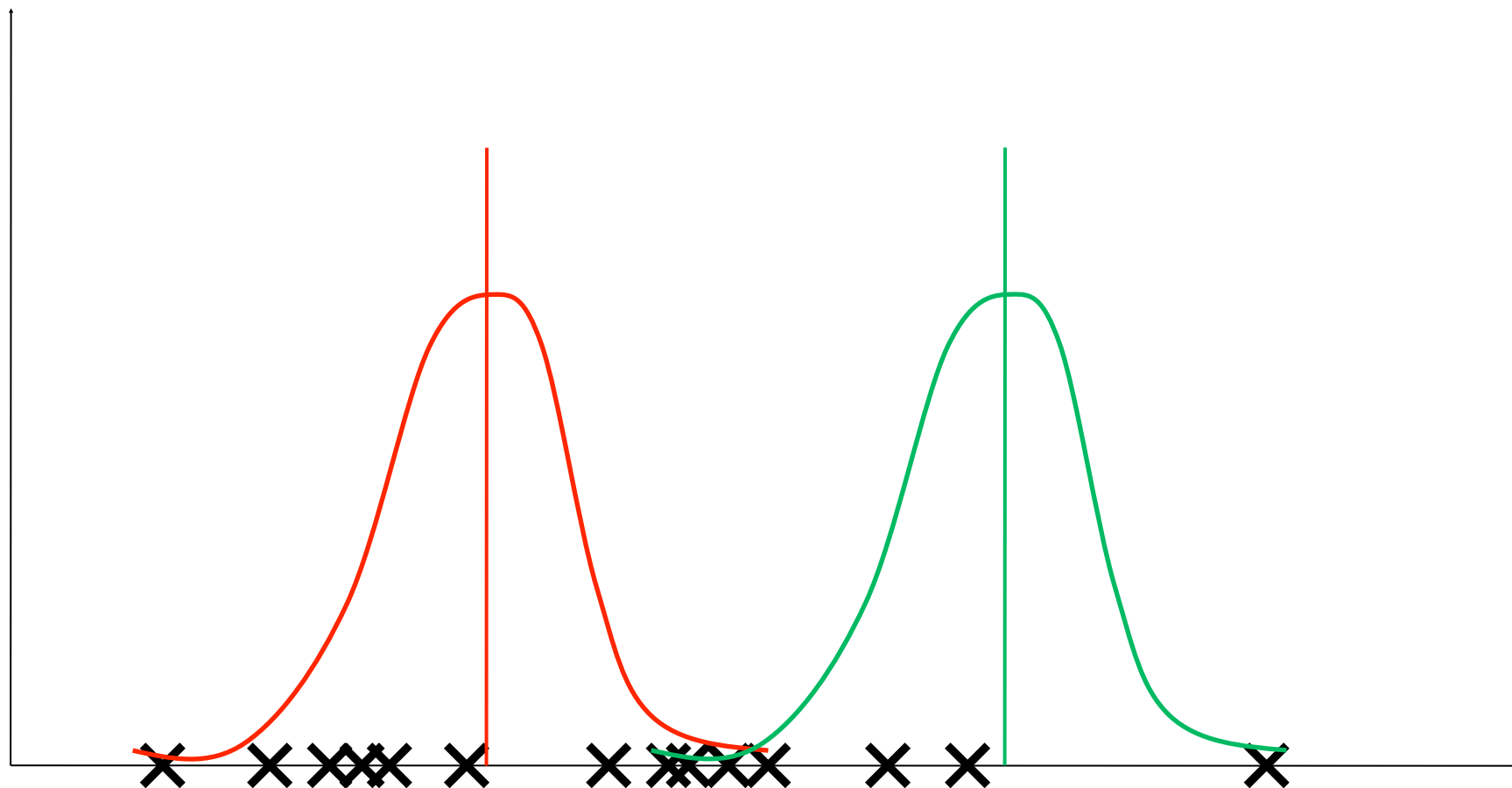
is minimal, where  $r_{nk} = 1$  if  $\mathbf{x}_n$  is assigned to  $\boldsymbol{\mu}_k$

- Idea: compute  $r_{nk}$  and  $\boldsymbol{\mu}_k$  iteratively
- Start with some values for the cluster centers
- Find optimal assignments  $r_{nk}$
- Update cluster centers using these assignments
- Repeat until assignments or centers don't change



# K-means Clustering

Initialize cluster means:  $\{\mu_1, \dots, \mu_K\}$

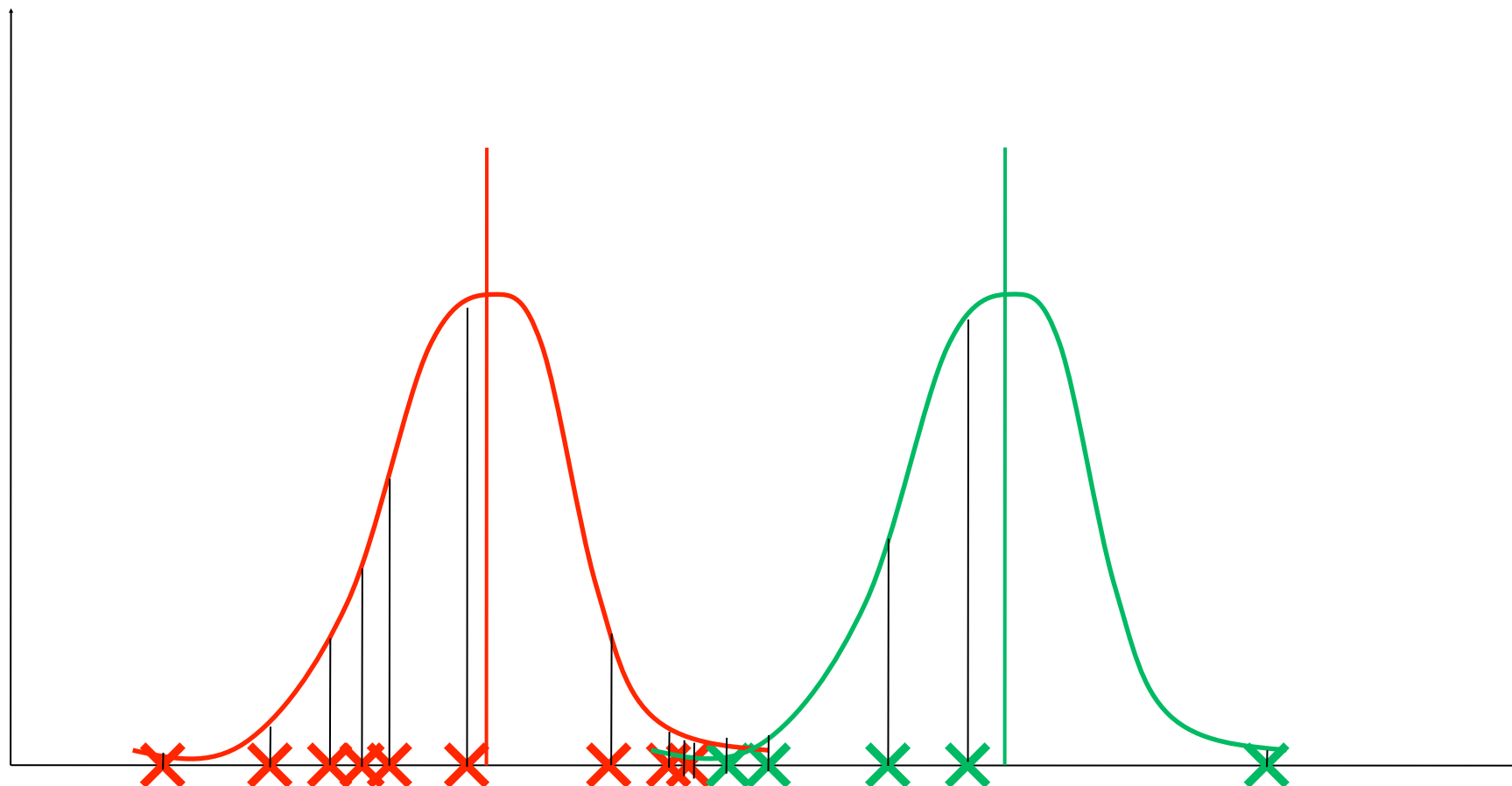




# K-means Clustering

Find optimal assignments:

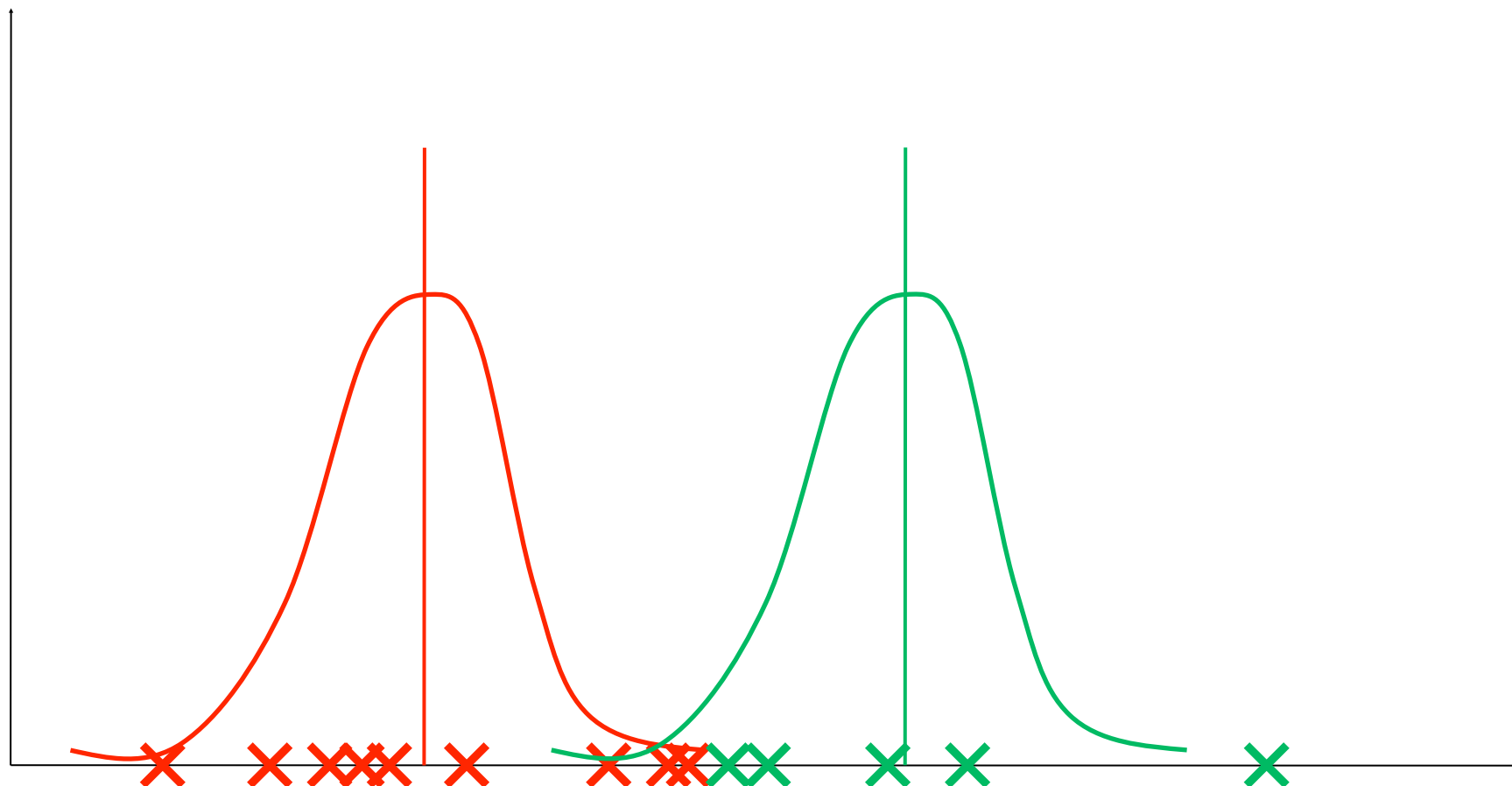
$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\| \\ 0 & \text{otherwise} \end{cases}$$



# K-means Clustering

Find new optimal means:  $\frac{\partial J}{\partial \mu_k} = 2 \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \mu_k) \stackrel{!}{=} 0$

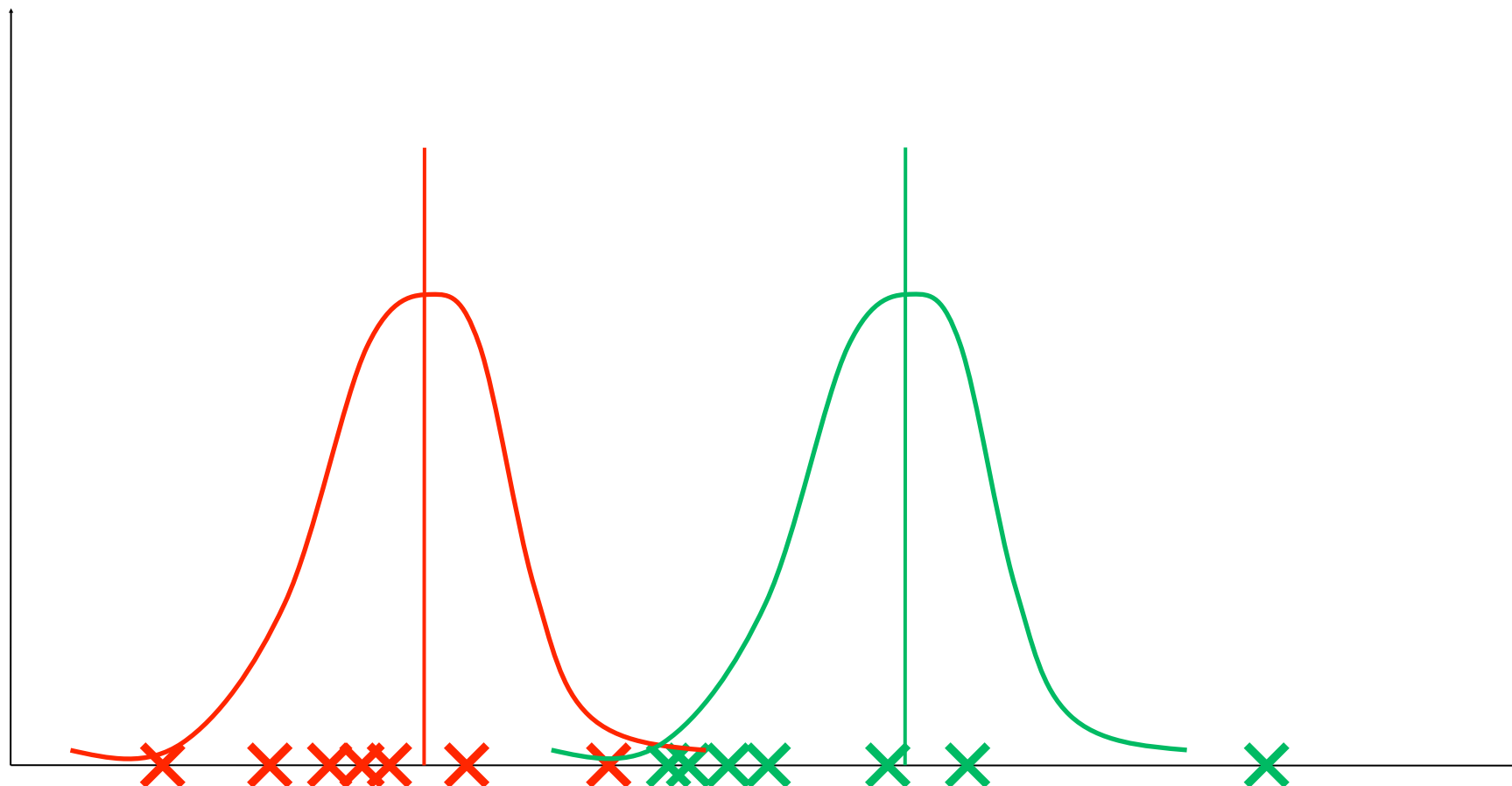
$$\Rightarrow \mu_k = \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{\sum_{n=1}^N r_{nk}}$$



# K-means Clustering

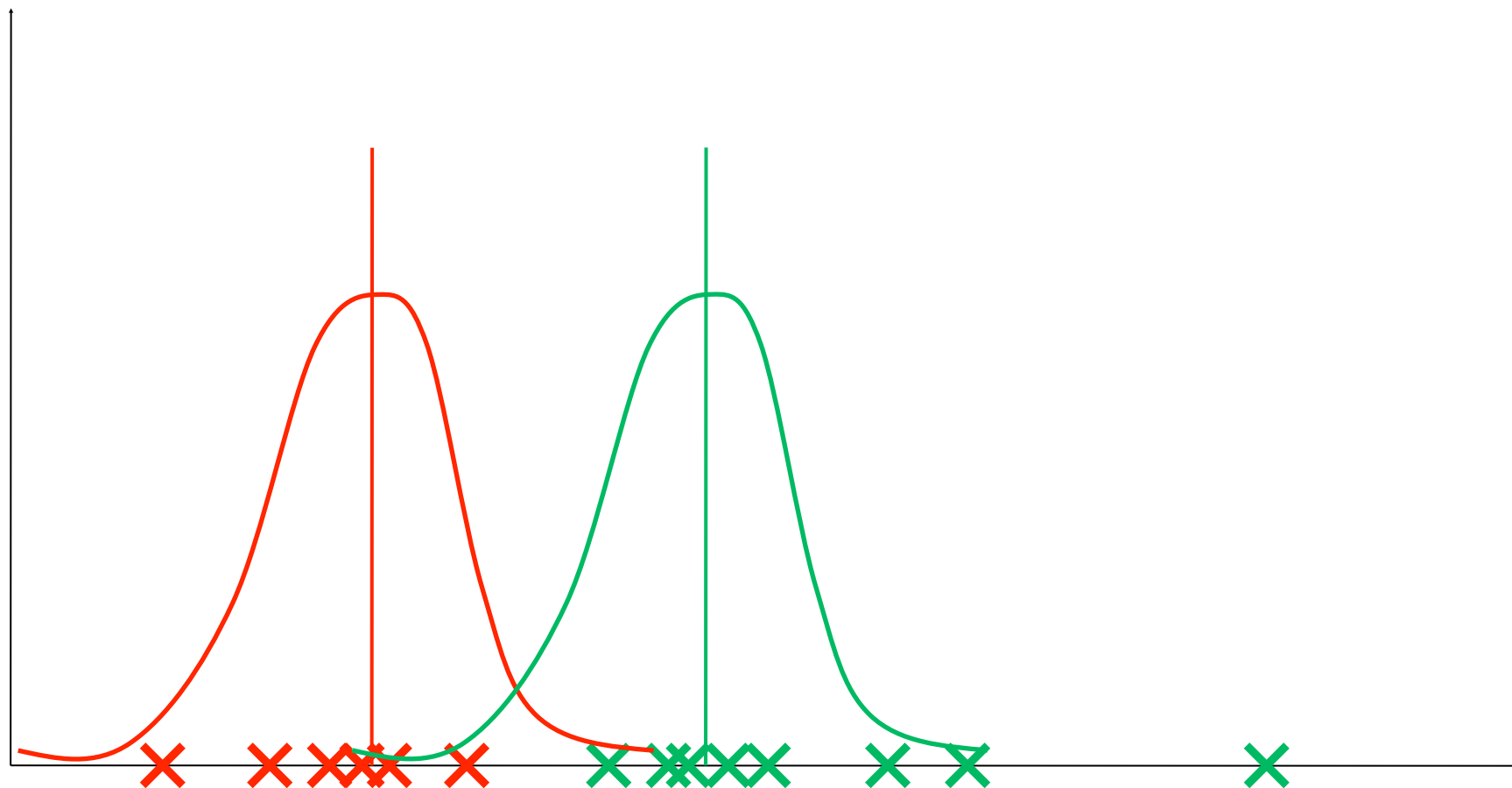
Find new optimal assignments:

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

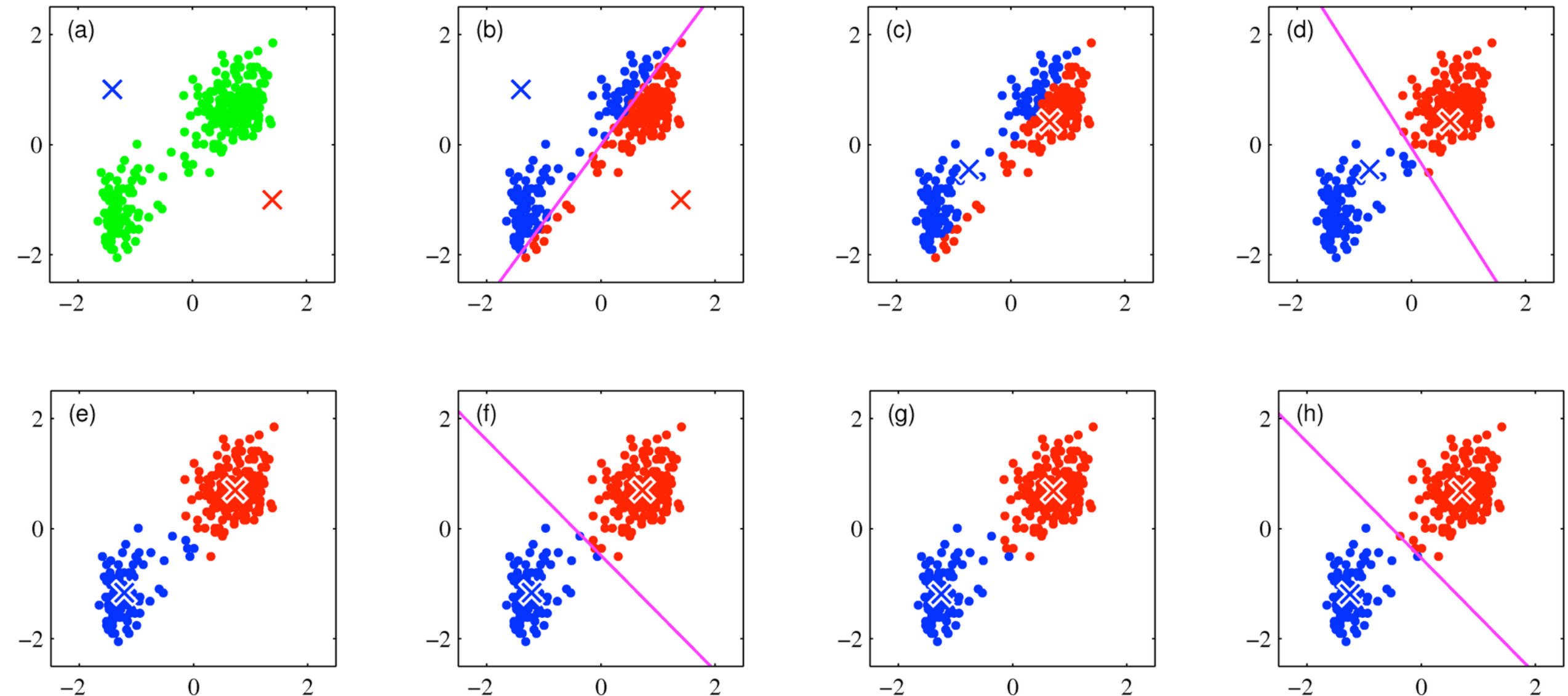


# K-means Clustering

Iterate these steps until means and assignments do not change any more



# 2D Example

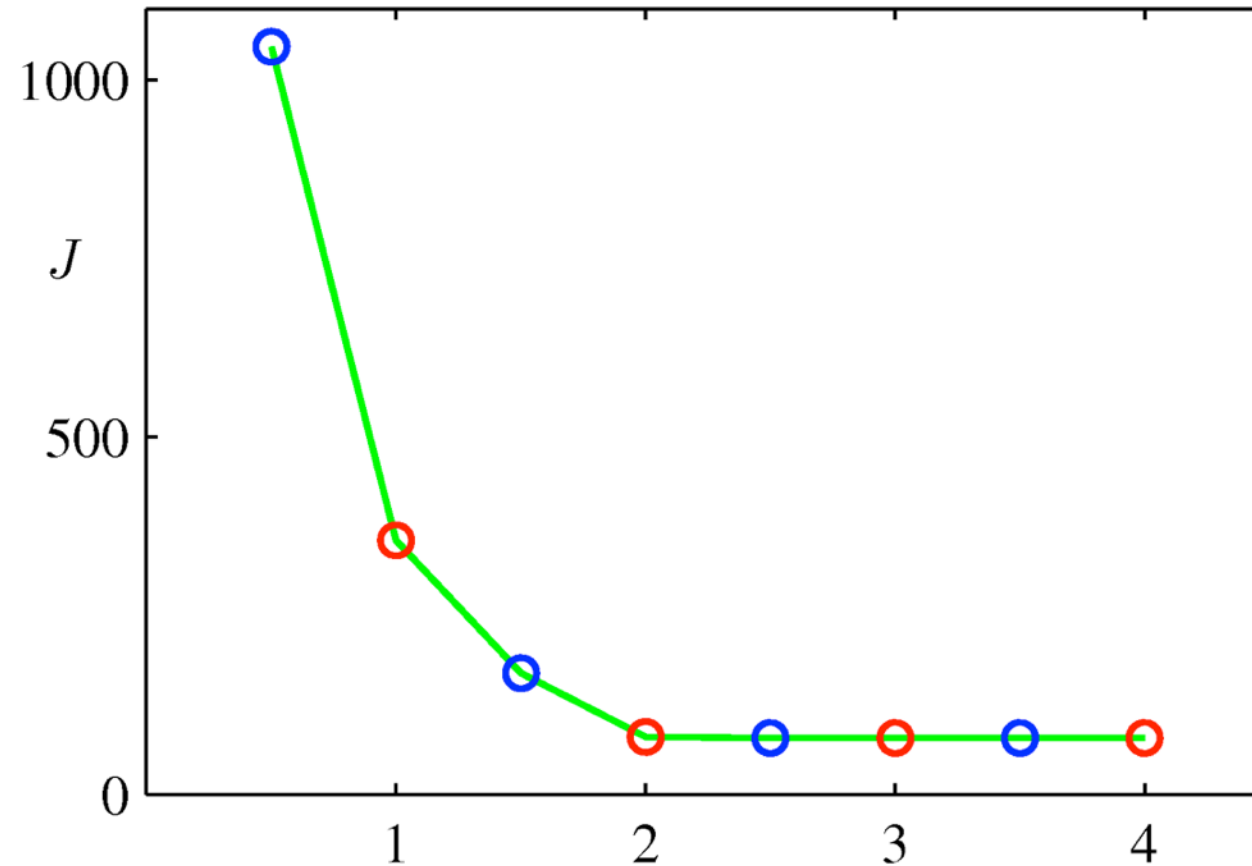


- Real data set
- Random initialization

- Magenta line is “decision boundary”



# The Cost Function



- After every step the cost function  $J$  is minimized
- Blue steps: update assignments
- Red steps: update means
- Convergence after 4 rounds



# K-means for Segmentation

$K = 2$



$K = 3$



$K = 10$



Original image





# K-Means: Additional Remarks

- K-means converges always, but the minimum is not guaranteed to be a global one
- There is an **online** version of  $K$ -means
  - After each addition of  $\mathbf{x}_n$ , the nearest center  $\mu_k$  is updated:
$$\mu_k^{\text{new}} = \mu_k^{\text{old}} + \eta_n (\mathbf{x}_n - \mu_k^{\text{old}})$$
- The  **$K$ -medoid** variant:
  - Replace the Euclidean distance by a general measure  $V$ .

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





# Mixtures of Gaussians

- Assume that the data consists of  $K$  clusters
- The data within each cluster is Gaussian
- For any data point  $\mathbf{x}$  we introduce a  $K$ -dimensional binary random variable  $\mathbf{z}$  so that:

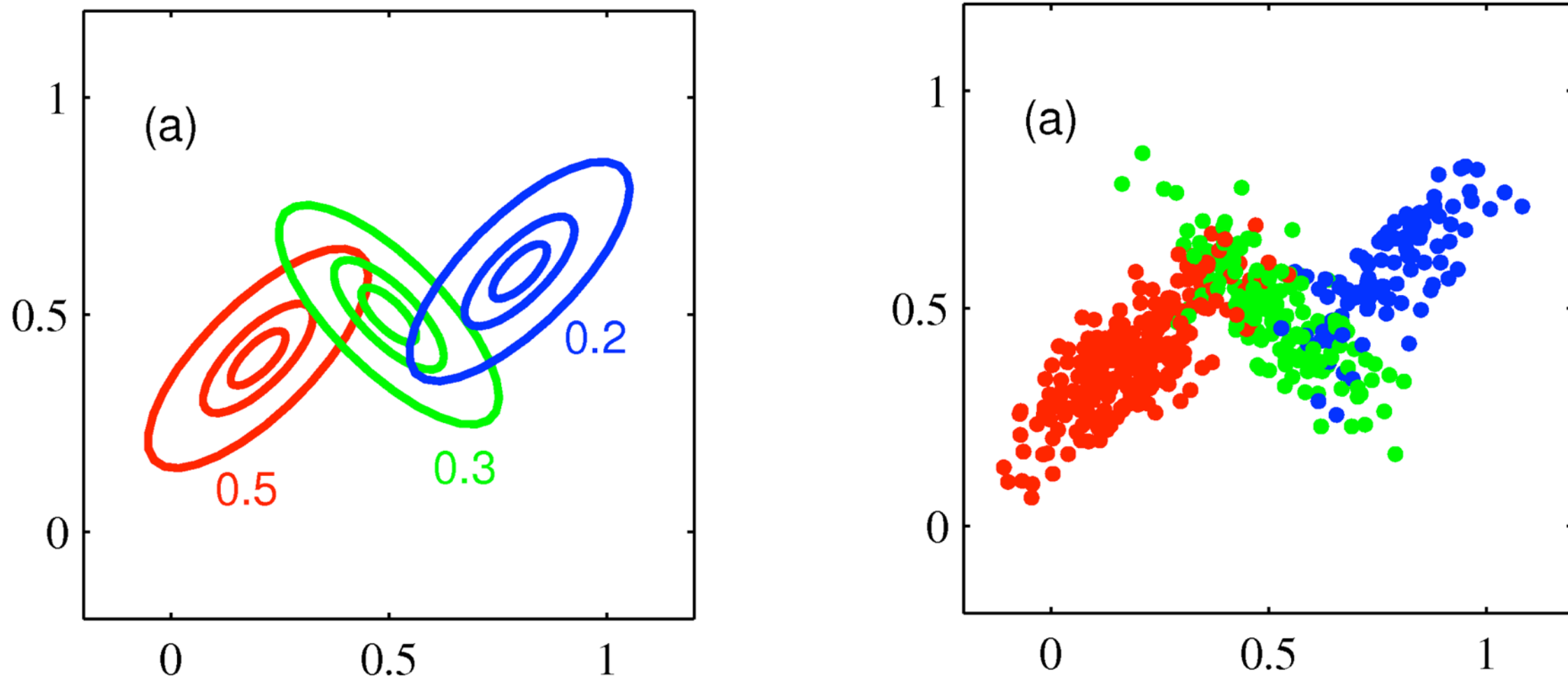
$$p(\mathbf{x}) = \sum_{k=1}^K \underbrace{p(z_k = 1)}_{=:\pi_k} \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$$

where

$$z_k \in \{0, 1\}, \quad \sum_{k=1}^K z_k = 1$$



# A Simple Example



- Mixture of three Gaussians with mixing coefficients
- Left: all three Gaussians as contour plot
- Right: samples from the mixture model, the red component has the most samples



# Parameter Estimation

- From a given set of training data  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  we want to find parameters  $(\pi_{1,\dots,K}, \boldsymbol{\mu}_{1,\dots,K}, \Sigma_{1,\dots,K})$  so that the likelihood is maximized (MLE):

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N \mid \pi_{1,\dots,K}, \boldsymbol{\mu}_{1,\dots,K}, \Sigma_{1,\dots,K}) = \prod_{n=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)$$

or, applying the logarithm:

$$\log p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \Sigma) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)$$

- However: this is not as easy as maximum-likelihood for single Gaussians!

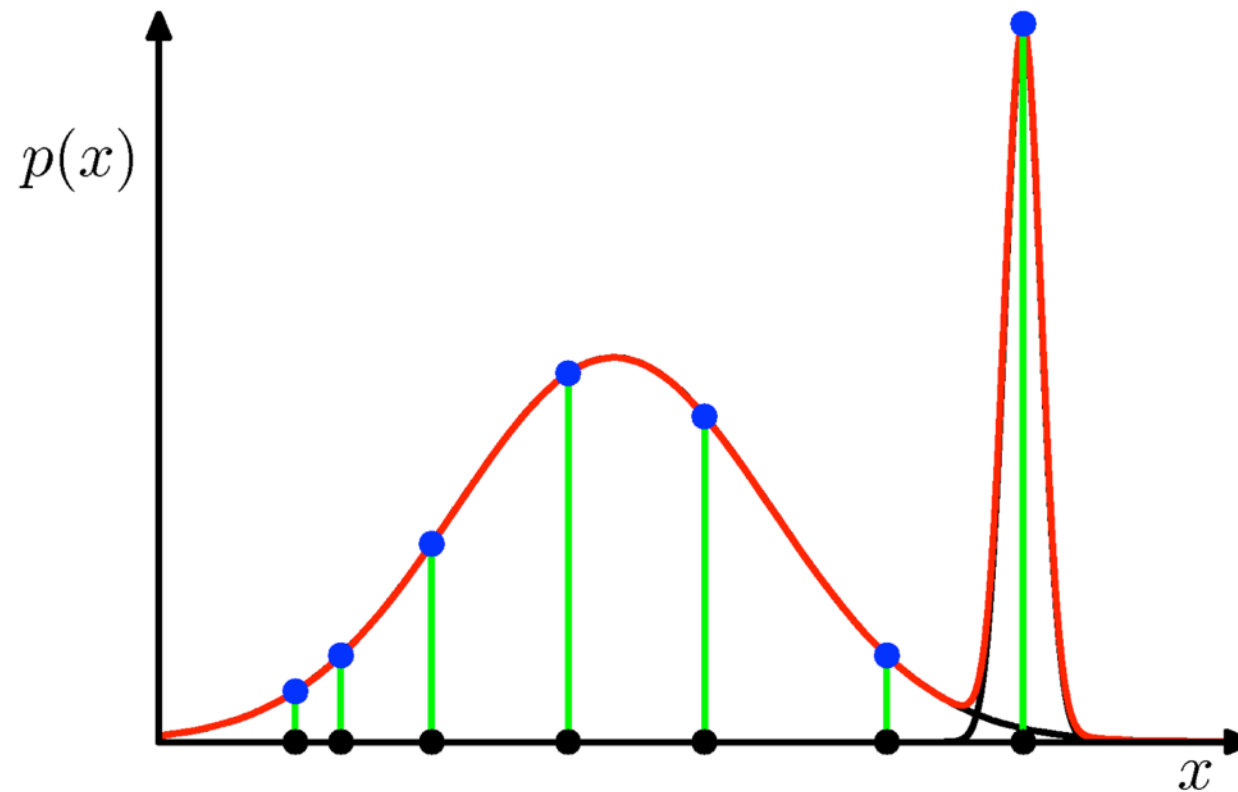


# Problems with MLE for Gaussian Mixtures

- Assume that for one  $k$  the mean  $\mu_k$  is exactly at a data point  $\mathbf{x}_n$ 
  - For simplicity: assume that  $\Sigma_k = \sigma_k^2 I$
  - Then: 
$$\mathcal{N}(\mathbf{x}_n \mid \mathbf{x}_n, \sigma_k^2 I) = \frac{1}{\sqrt{2\pi} \sigma_k^D}$$
  - This means that the overall log-likelihood can be maximized arbitrarily by letting  $\sigma_k \rightarrow 0$  (**overfitting**)
- Another problem is the **identifiability**:
  - The order of the Gaussians is not fixed, therefore:
  - There are  $K!$  equivalent solutions to the MLE problem



# Overfitting with MLE for Gaussian Mixtures



- One Gaussian fits exactly to one data point
- It has a very small variance, i.e. contributes strongly to the overall likelihood
- In standard MLE, there is no way to avoid this!



# Expectation-Maximization

- EM is an elegant and powerful method for MLE problems with latent variables
- Main idea: model parameters and latent variables are estimated iteratively, where average over the latent variables (expectation)
- A typical example application of EM is the Gaussian Mixture model (GMM)
- However, EM has many other applications
- First, we consider EM for GMMs



# Expectation-Maximization for GMM

- First, we define the **responsibilities**:

$$\gamma(z_{nk}) = p(z_{nk} = 1 \mid \mathbf{x}_n) \quad z_{nk} \in \{0, 1\}$$
$$\sum_k z_{nk} = 1$$



# Expectation-Maximization for GMM

- First, we define the **responsibilities**:

$$\begin{aligned}\gamma(z_{nk}) &= p(z_{nk} = 1 \mid \mathbf{x}_n) \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j)}\end{aligned}$$





# Expectation-Maximization for GMM

- First, we define the **responsibilities**:

$$\begin{aligned}\gamma(z_{nk}) &= p(z_{nk} = 1 \mid \mathbf{x}_n) \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j)}\end{aligned}$$

- Next, we derive the log-likelihood wrt. to  $\boldsymbol{\mu}_k$ :

$$\frac{\partial \log p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \Sigma)}{\partial \boldsymbol{\mu}_k} \stackrel{!}{=} \mathbf{0}$$



# Expectation-Maximization for GMM

- First, we define the **responsibilities**:

$$\begin{aligned}\gamma(z_{nk}) &= p(z_{nk} = 1 \mid \mathbf{x}_n) \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j)}\end{aligned}$$

- Next, we derive the log-likelihood wrt. to  $\boldsymbol{\mu}_k$ :

$$\frac{\partial \log p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \Sigma)}{\partial \boldsymbol{\mu}_k} \stackrel{!}{=} \mathbf{0}$$

and we obtain:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^N \gamma(z_{nk})}$$



# Expectation-Maximization for GMM

- We can do the same for the covariances:

$$\frac{\partial \log p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k} \stackrel{!}{=} \mathbf{0}$$

and we obtain:

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^N \gamma(z_{nk})}$$

- Finally, we derive wrt. the mixing coefficients  $\pi_k$  :

$$\frac{\partial \log p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \pi_k} \stackrel{!}{=} \mathbf{0} \quad \text{where:} \quad \sum_{k=1}^K \pi_k = 1$$



# Expectation-Maximization for GMM

- We can do the same for the covariances:

$$\frac{\partial \log p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k} \stackrel{!}{=} \mathbf{0}$$

and we obtain:

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^N \gamma(z_{nk})}$$

- Finally, we derive wrt. the mixing coefficients  $\pi_k$  :

$$\frac{\partial \log p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \pi_k} \stackrel{!}{=} \mathbf{0} \quad \text{where:} \quad \sum_{k=1}^K \pi_k = 1$$

and the result is:

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



# Algorithm Summary

1. Initialize means  $\mu_k$  covariance matrices  $\Sigma_k$  and mixing coefficients  $\pi_k$
2. Compute the initial log-likelihood  $\log p(X \mid \pi, \mu, \Sigma)$
3. **E-Step.** Compute the responsibilities:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n \mid \mu_j, \Sigma_j)}$$

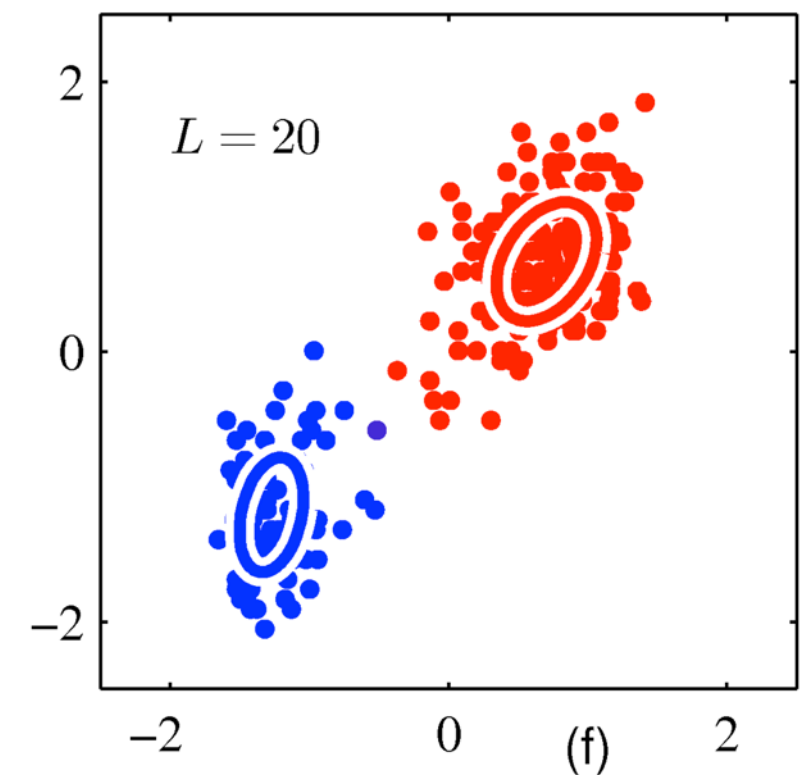
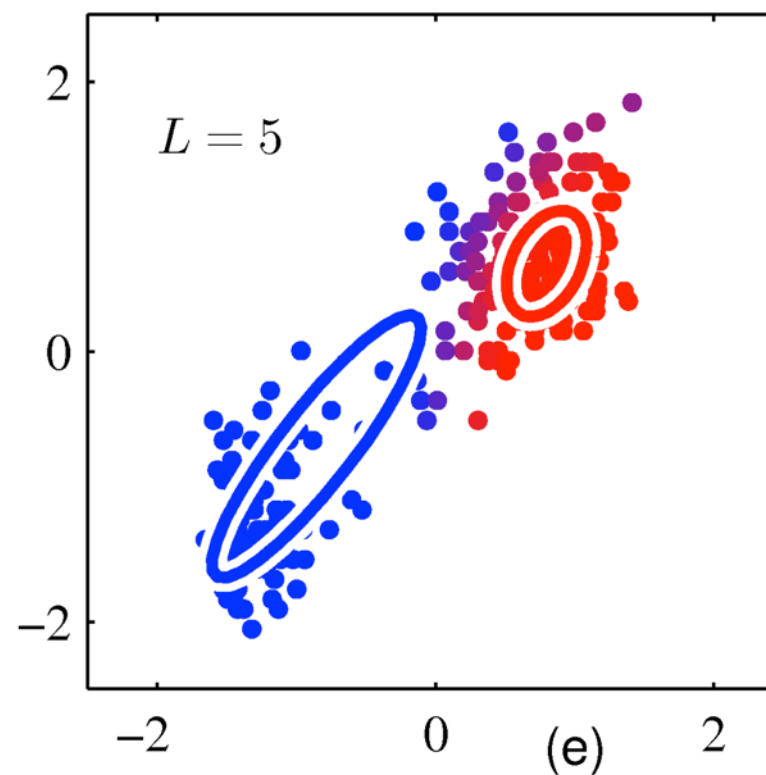
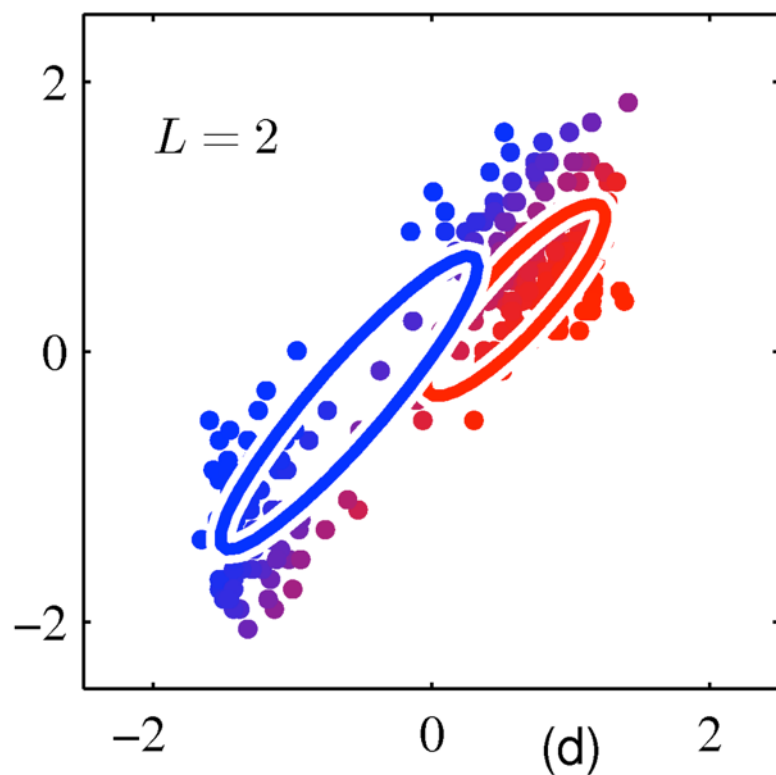
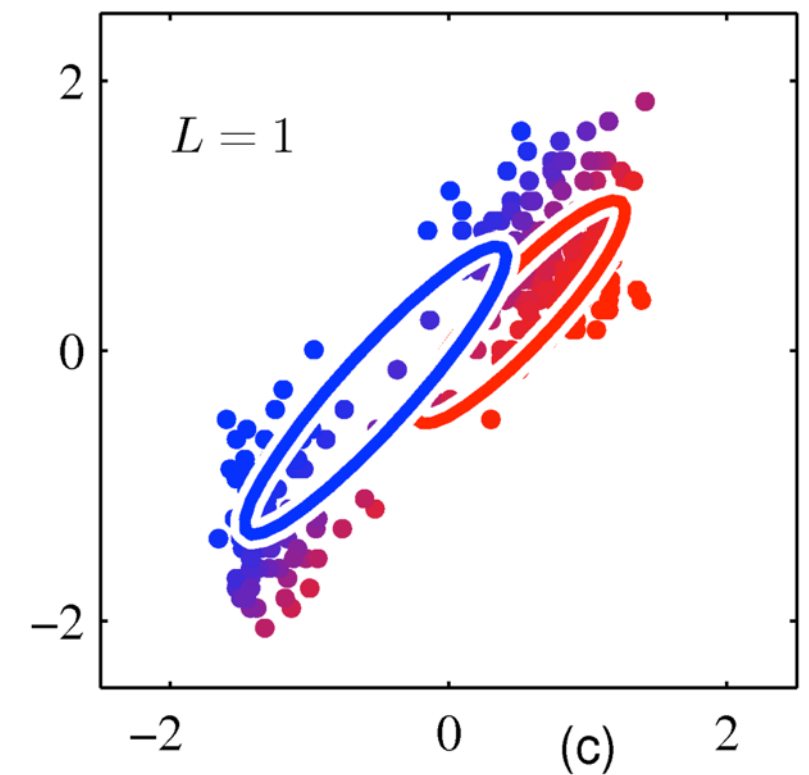
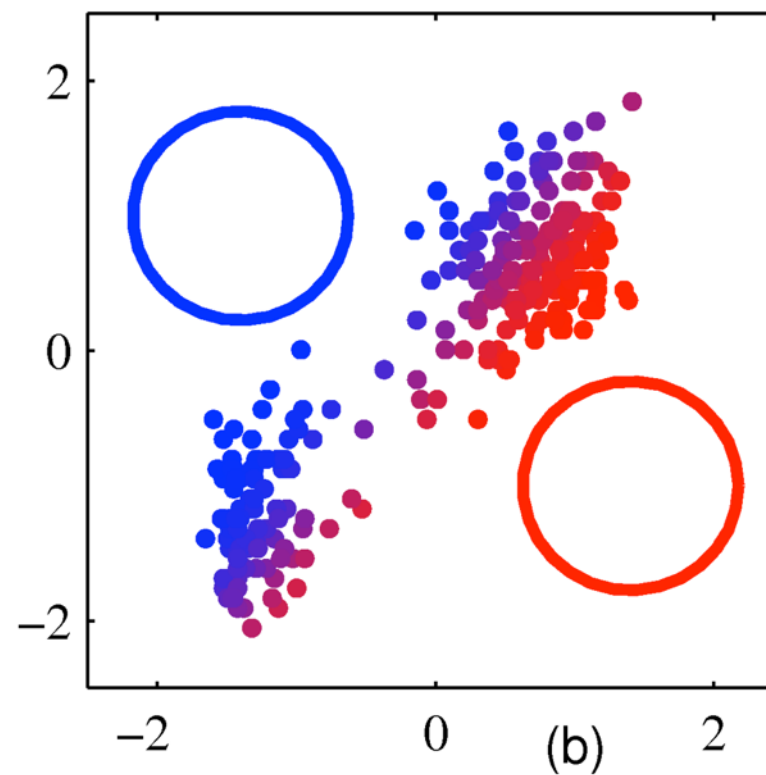
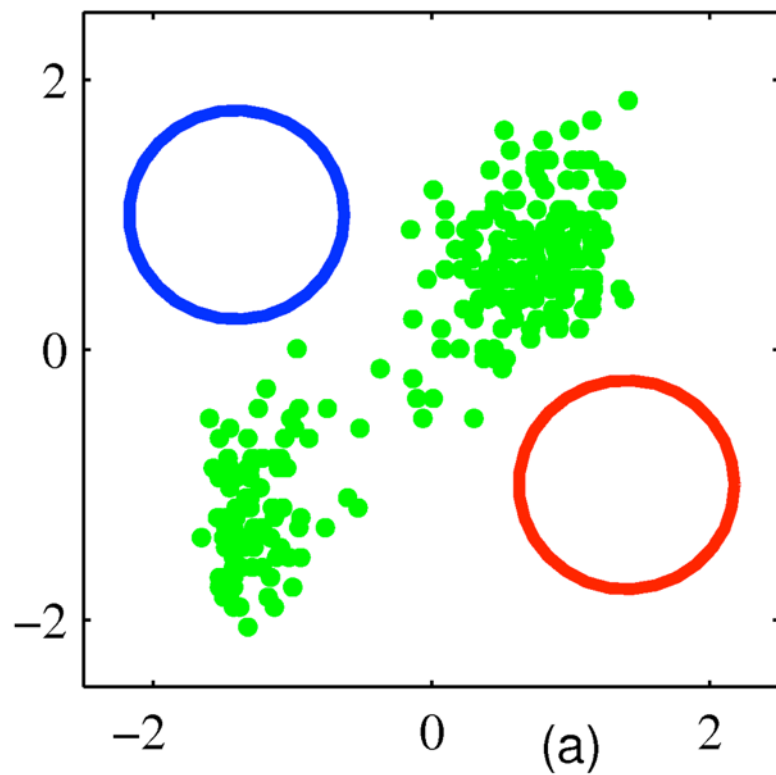
4. **M-Step.** Update the parameters:

$$\mu_k^{\text{new}} = \frac{\sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^N \gamma(z_{nk})} \quad \Sigma_k^{\text{new}} = \frac{\sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \mu_k^{\text{new}})(\mathbf{x}_n - \mu_k^{\text{new}})^T}{\sum_{n=1}^N \gamma(z_{nk})} \quad \pi_k^{\text{new}} = \frac{1}{N} \sum_{n=1}^N \gamma(z_{nk})$$

5. Compute log-likelihood; if not converged go to 3.



# The Same Example Again



# Observations

- Compared to K-means, points can now belong to both clusters (**soft assignment**)
- In addition to the cluster center, a covariance is estimated by EM
- Initialization is the same as used for K-means
- Number of iterations needed for EM is much higher
- Also: each cycle requires much more computation
- Therefore: start with K-means and run EM on the result of K-means (covariances can be initialized to the sample covariances of K-means)
- EM only finds a **local** maximum of the likelihood!



# A More General View of EM

- Assume for a moment that we observe  $X$  and the binary latent variables  $Z$ . The likelihood is then:

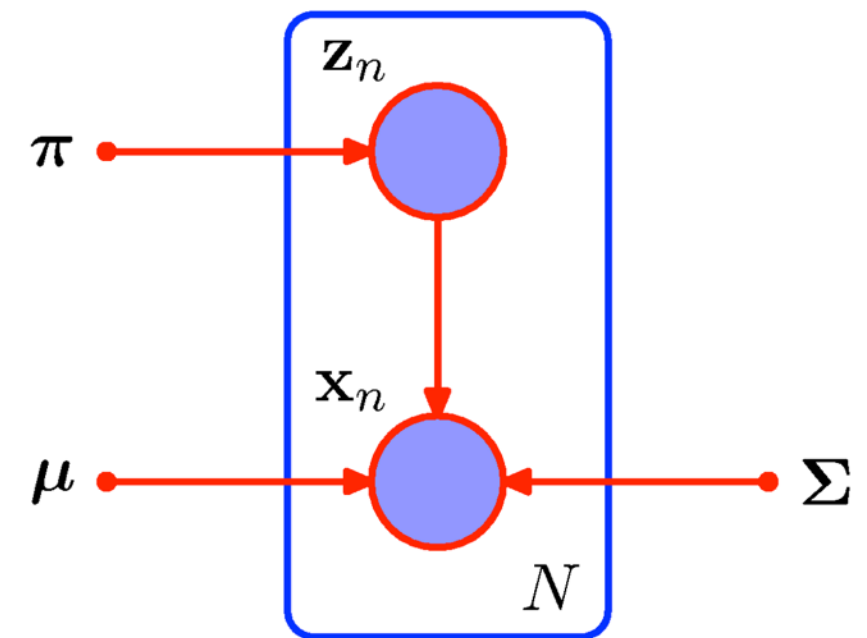
$$p(X, Z \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^N p(\mathbf{z}_n \mid \boldsymbol{\pi}) p(\mathbf{x}_n \mid \mathbf{z}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

**Remember:**

$$z_{nk} \in \{0, 1\}, \quad \sum_{k=1}^K z_{nk} = 1$$

where  $p(\mathbf{z}_n \mid \boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{z_{nk}}$  and

$$p(\mathbf{x}_n \mid \mathbf{z}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{k=1}^K \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$



which leads to the log-formulation:

$$\log p(X, Z \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} (\log \pi_k + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))$$





# The Complete-Data Log-Likelihood

$$\log p(X, Z \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} (\log \pi_k + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))$$

- This is called the **complete-data log-likelihood**
- Advantage: solving for the parameters  $(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$  is much simpler, as the log is inside the sum!
- We could switch the sums and then for every mixture component  $k$  only look at the points that are associated with that component.
- This leads to simple closed-form solutions for the parameters
- However: the latent variables  $Z$  are not observed!



# The Main Idea of EM

- Instead of maximizing the joint log-likelihood, we maximize its **expectation** under the latent variable distribution:

$$\mathbb{E}_Z[\log p(X, Z \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \Sigma)] = \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}_Z[z_{nk}] (\log \pi_k + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k))$$

where the latent variable distribution per point is:

$$\begin{aligned} p(\mathbf{z}_n \mid \mathbf{x}_n, \boldsymbol{\theta}) &= \frac{p(\mathbf{x}_n \mid \mathbf{z}_n, \boldsymbol{\theta}) p(\mathbf{z}_n \mid \boldsymbol{\theta})}{p(\mathbf{x}_n \mid \boldsymbol{\theta})} & \boldsymbol{\theta} &= (\boldsymbol{\pi}, \boldsymbol{\mu}, \Sigma) \\ &= \frac{\prod_{l=1}^K (\pi_l \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_l, \Sigma_l))^{z_{nl}}}{\sum_{j=1}^K (\pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \Sigma_j))^{z_{nj}}} \end{aligned}$$



# The Main Idea of EM

The expected value of the latent variables is:

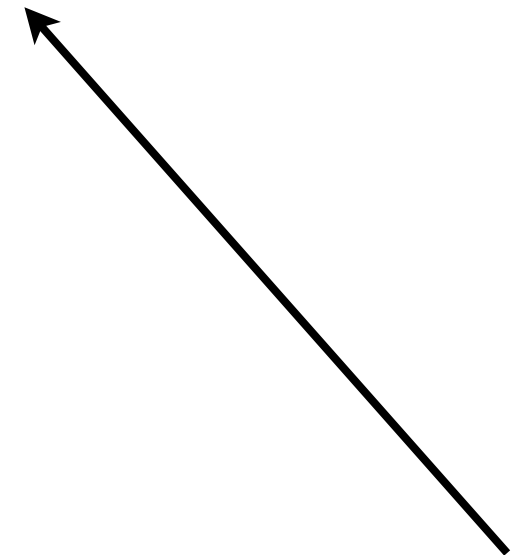
$$\mathbb{E}[z_{nk}] = \gamma(z_{nk})$$

plugging in we obtain:

$$\mathbb{E}_Z[\log p(X, Z \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \Sigma)] = \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) (\log \pi_k + \log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k))$$

We compute this iteratively:

1. Initialize  $i = 0$ ,  $(\pi_k^i, \boldsymbol{\mu}_k^i, \Sigma_k^i)$
2. Compute  $\mathbb{E}[z_{nk}] = \gamma(z_{nk})$
3. Find parameters  $(\pi_k^{i+1}, \boldsymbol{\mu}_k^{i+1}, \Sigma_k^{i+1})$  that maximize this
4. Increase  $i$ ; if not converged, goto 2.



# The Theory Behind EM

- We have seen that EM maximizes the **expected complete-data log-likelihood**, but:
- Actually, we need to maximize the log-marginal

$$\log p(X \mid \theta) = \log \sum_Z p(X, Z \mid \theta)$$

- It turns out that the log-marginal is maximized **implicitly!**



# The Theory Behind EM

- We have seen that EM maximizes the **expected complete-data log-likelihood**, but:
- Actually, we need to maximize the log-marginal

$$\log p(X \mid \boldsymbol{\theta}) = \log \sum_Z p(X, Z \mid \boldsymbol{\theta})$$

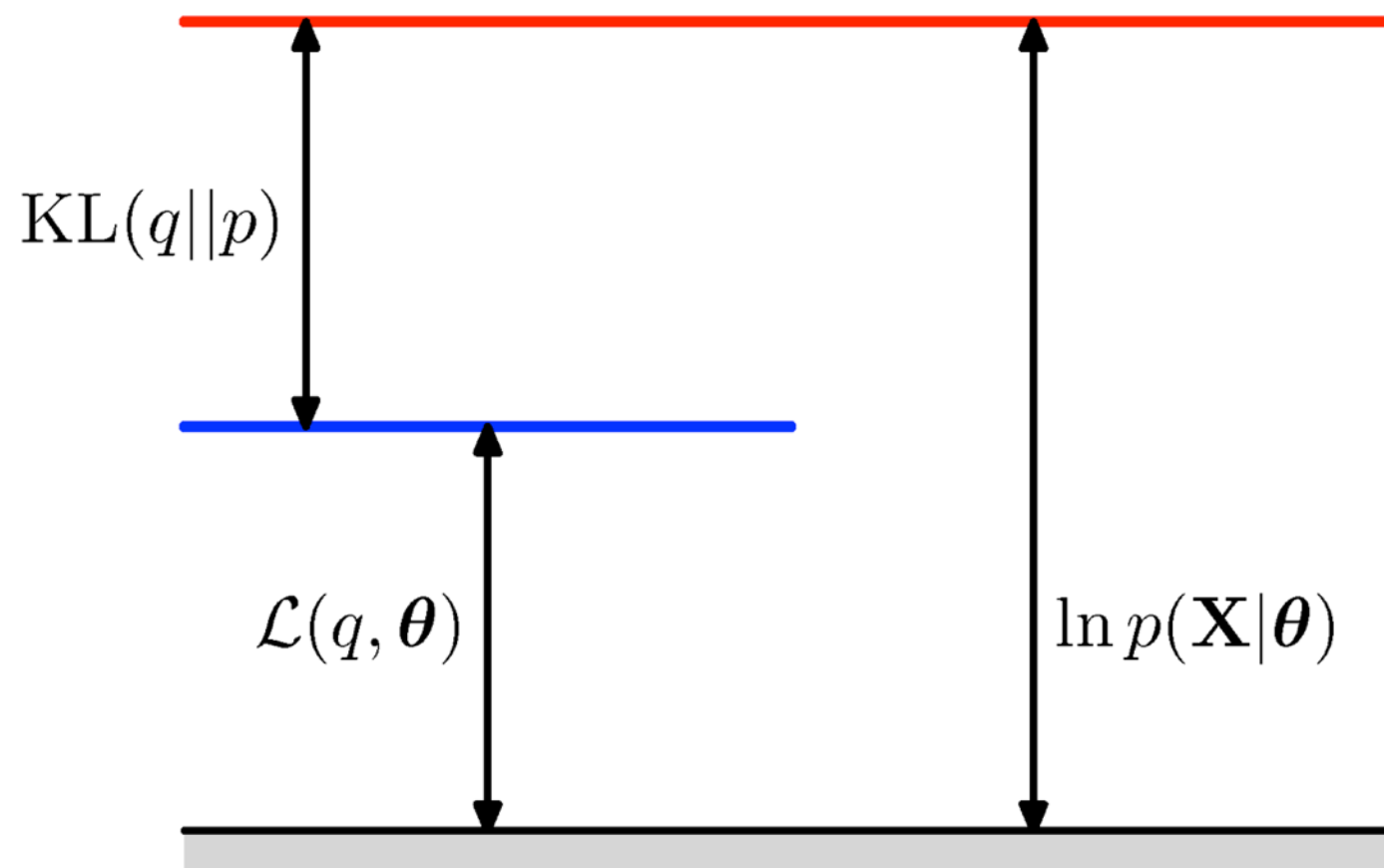
- It turns out that the log-marginal is maximized **implicitly!**

$$\log p(X \mid \boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \text{KL}(q \parallel p)$$

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_Z q(Z) \log \frac{p(X, Z \mid \boldsymbol{\theta})}{q(Z)} \quad \text{KL}(q \parallel p) = - \sum_Z q(Z) \log \frac{p(Z \mid X, \boldsymbol{\theta})}{q(Z)}$$



# Visualization

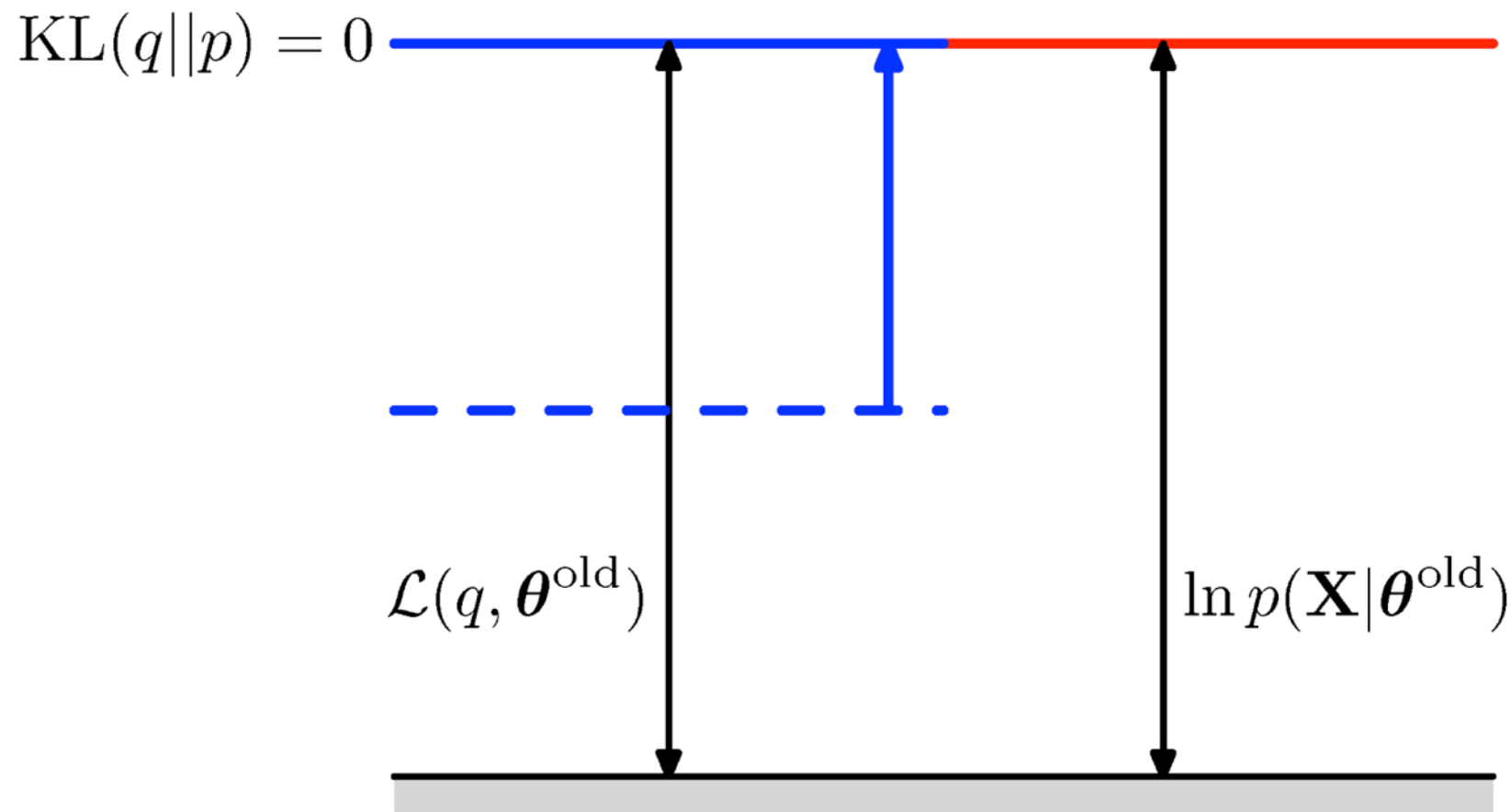


- The KL-divergence is positive or 0
- Thus, the log-likelihood is at least as large as  $\mathcal{L}$  or:
- $\mathcal{L}$  is a lower bound of the log-likelihood:

$$\log p(X \mid \theta) \geq \mathcal{L}(q, \theta)$$



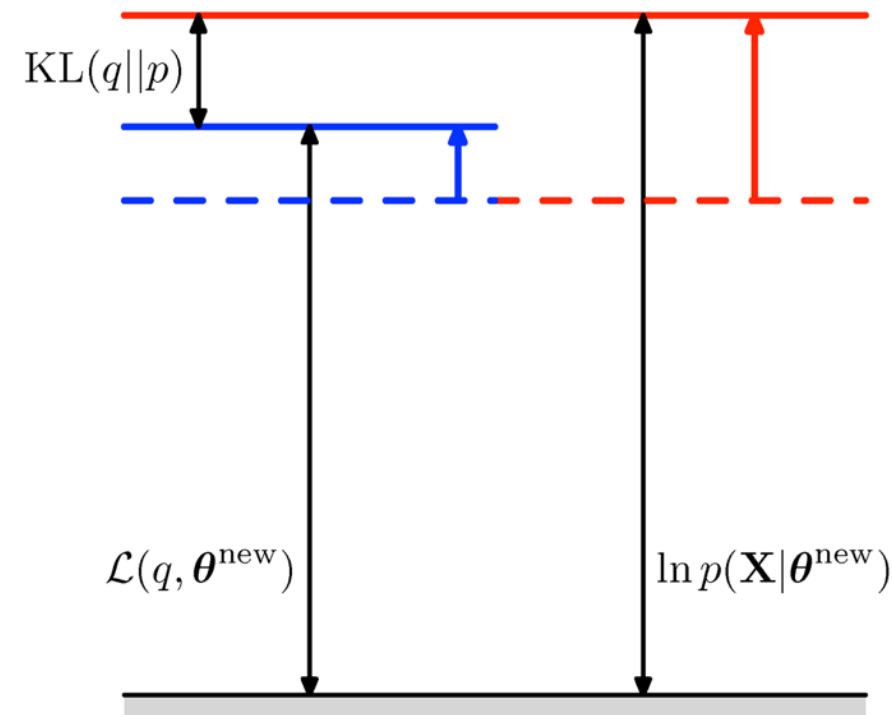
# What Happens in the E-Step?



- The log-likelihood is independent of  $q$
- Thus:  $L$  is maximized iff  $KL$  is minimal
- This is the case iff  $q(Z) = p(Z | X, \theta)$



# What Happens in the M-Step?



- In the M-step we keep  $q$  fixed and find new  $\theta$

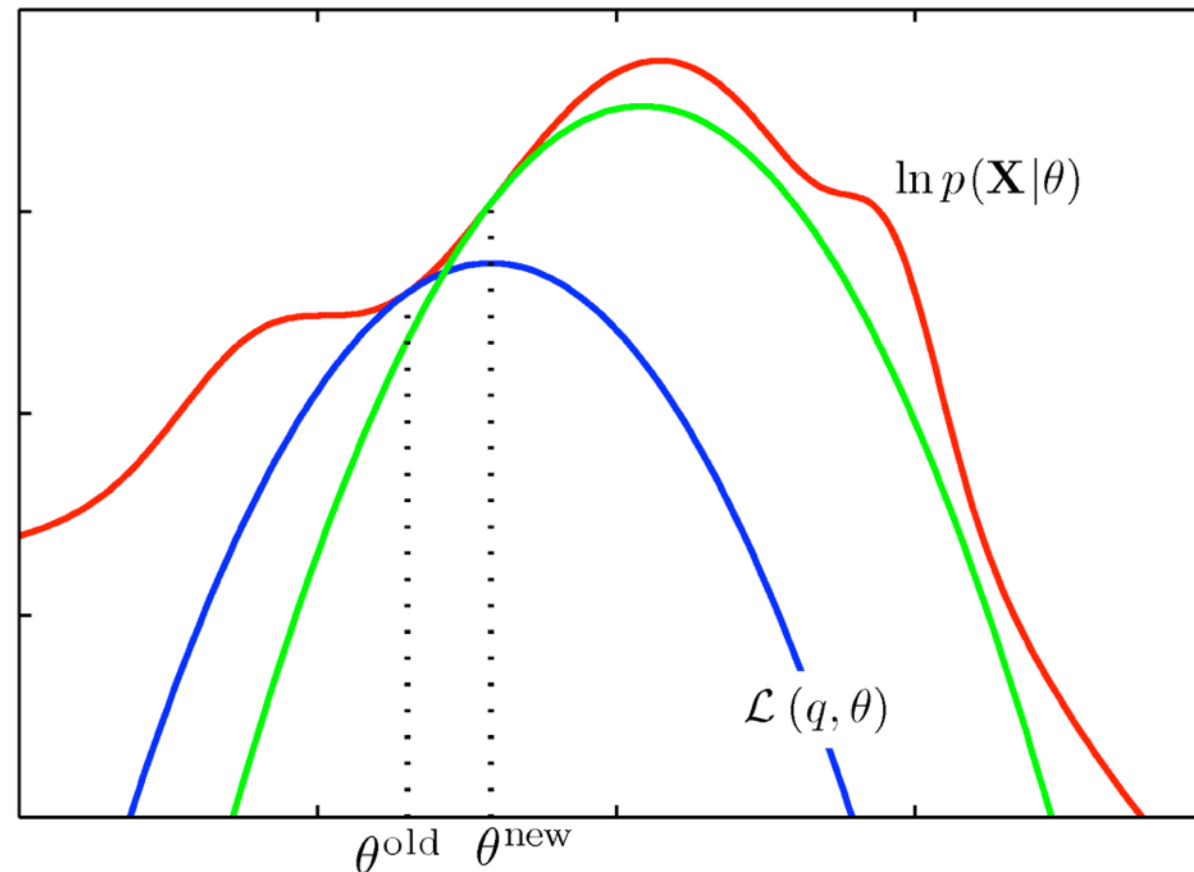
$$\mathcal{L}(q, \theta) = \sum_Z p(Z | X, \theta^{\text{old}}) \log p(X, Z | \theta) - \sum_Z q(Z) \log q(Z)$$

- We maximize the first term, the second is indep.
- This implicitly makes KL non-zero
- The log-likelihood is maximized even more!





# Visualization in Parameter-Space



- In the E-step we compute the concave lower bound for given old parameters  $\theta^{\text{old}}$  (blue curve)
- In the M-step, we maximize this lower bound and obtain new parameters  $\theta^{\text{new}}$
- This is repeated (green curve) until convergence



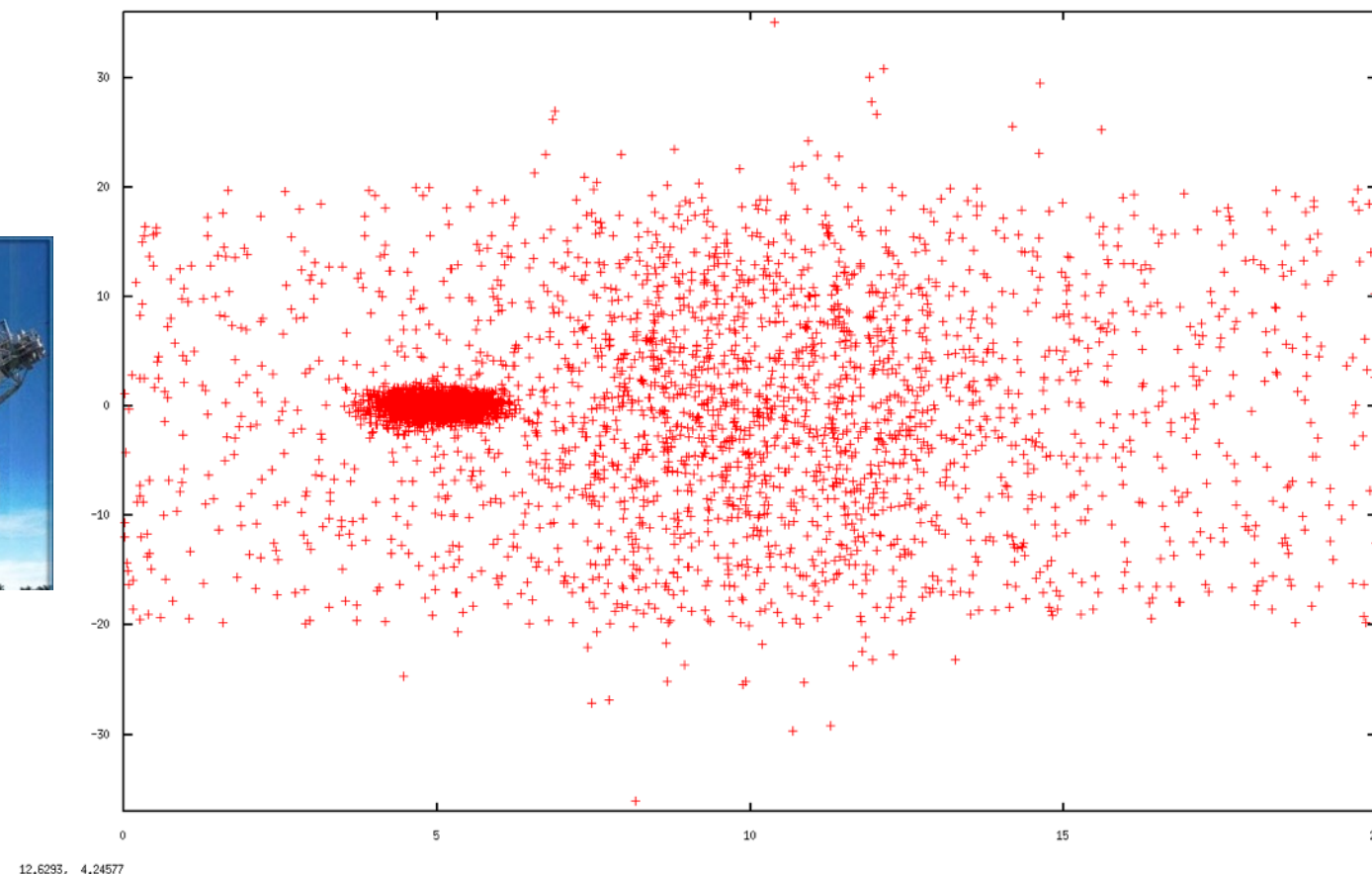
# Variants of EM

- Instead of maximizing the log-likelihood, we can use EM to maximize a **posterior** when a prior is given (MAP instead of MLE)  $\Rightarrow$  less overfitting
- In Generalized EM, the M-step only increases the lower bound instead of maximization (useful if standard M-step is intractable)
- Similarly, the E-step can be generalized in that the optimization wrt.  $q$  is not complete
- Furthermore, there are incremental versions of EM, where data points are given sequentially and the parameters are updated after each data point.



# Example 1: Learn a Sensor Model

- A Radar range finder on a metallic target will returns 3 types of measurement:
  - The distance to target
  - The distance to the wall behind the target
  - A completely random value

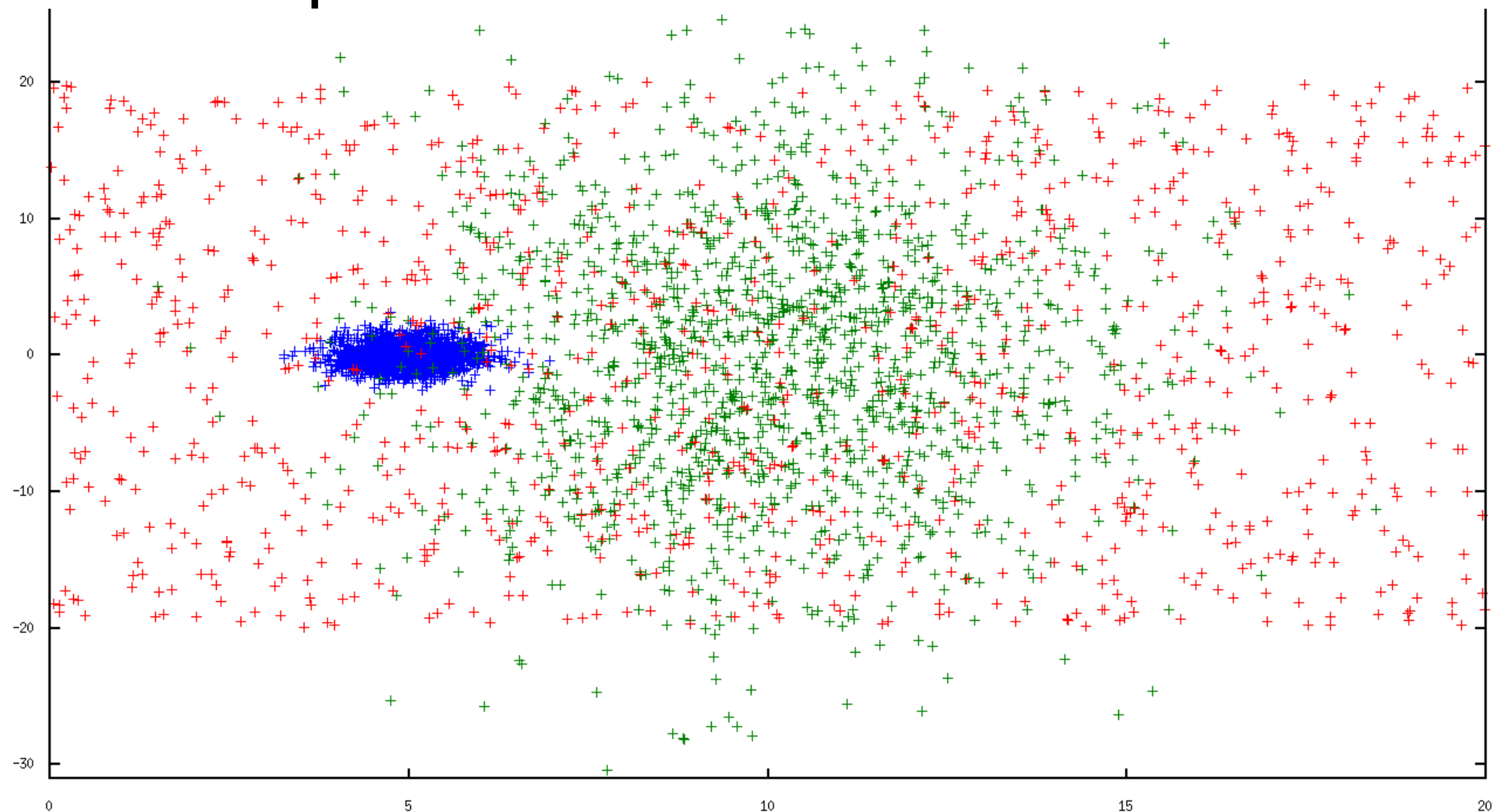


12.6293, 4.24577



# Example 1: Learn a Sensor Model

- Which point corresponds to from which model?
- What are the different model parameters?
- Solution: Expectation-Maximization

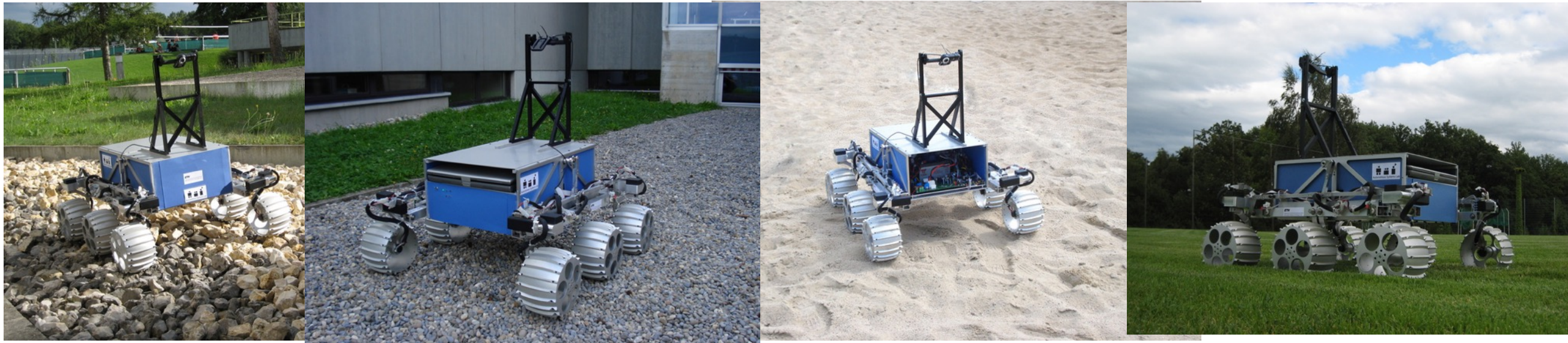


10,1361, 3,56237

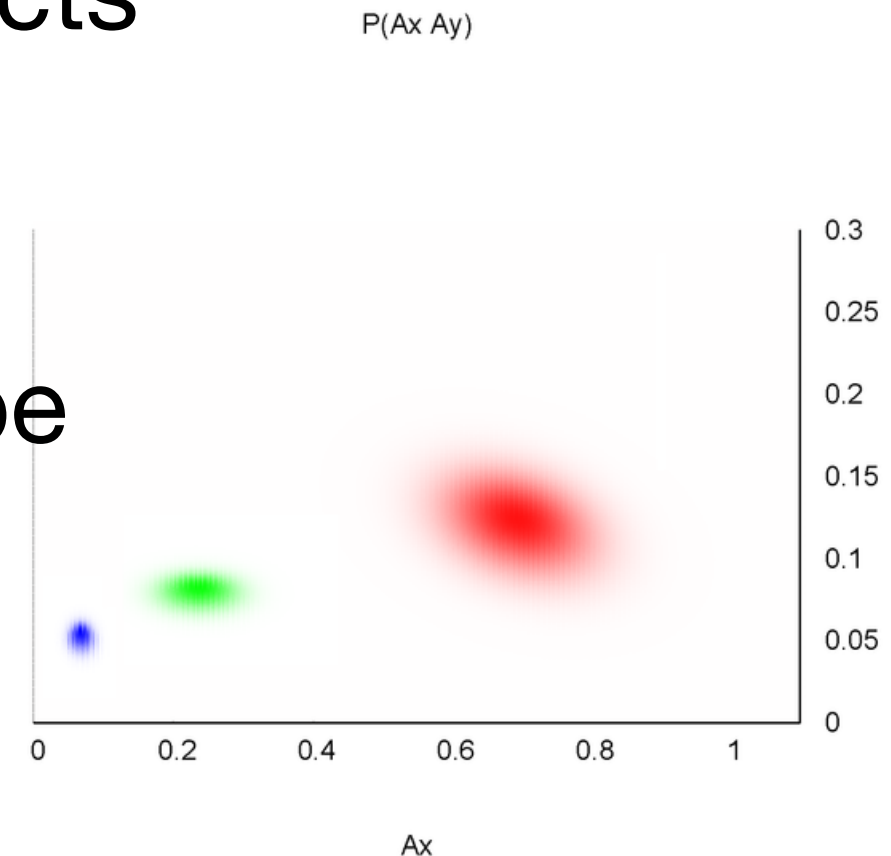




# Example 2: Environment Classification



- From each image, the robot extracts features:  $\Rightarrow$  points in  $nD$  space
- K-means only finds the cluster centers, not their extent and shape
- The centers and covariances can be obtained with EM



# Example 3: Plane Fitting in 3D

- Has been done in this paper
- Given a set of 3D points, fit planes into the data
- Idea: Model parameters  $\theta$  are normal vectors and distance to origin for a set of planes
- Gaussian noise model:  $p(z \mid \theta) = \mathcal{N}(d(\mathbf{z}, \theta) \mid 0, \sigma)$ 

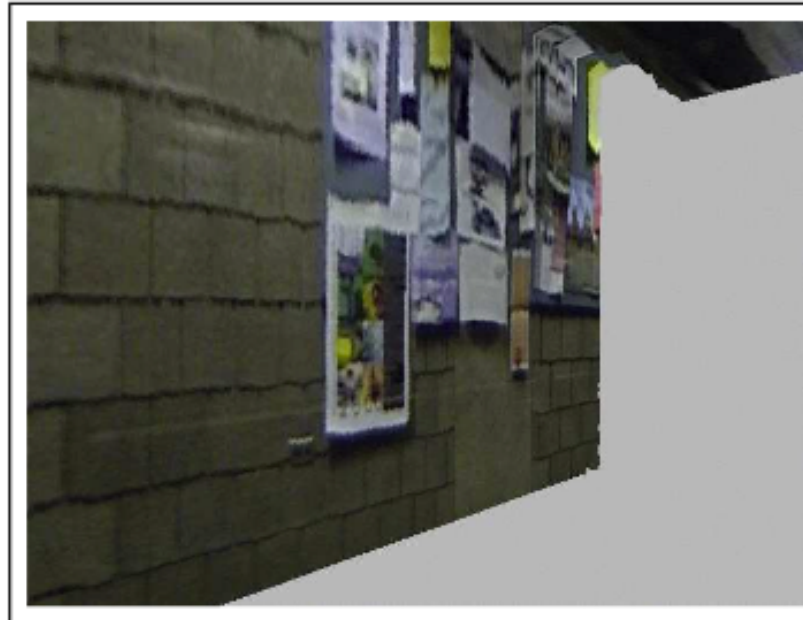
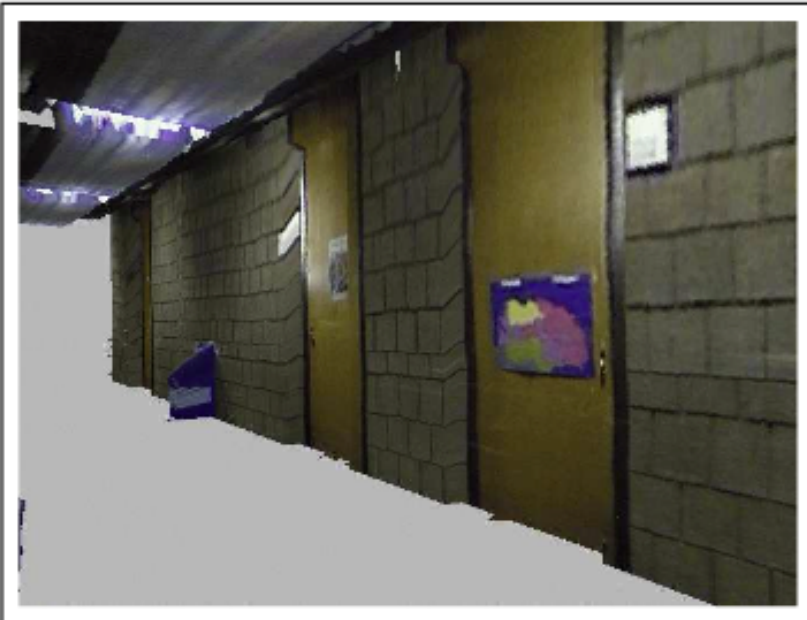
$\uparrow$   
point-to-plane  
distance

$\uparrow$   
noise  
variance
- Introduce latent correspondence variables  $C_{ij}$  and maximize the expected log-lik.:
$$\mathbb{E}[\log p(Z, C \mid \theta)]$$
- Maximization can be done in closed form





# Example 3: Plane Fitting in 3D



# Summary

- K-means is an iterative method for clustering
- Mixture models can be formalized using latent (unobserved) variables
- A very common example are Gaussian mixture models (GMMs)
- To estimate the parameters of a GMM we can use expectation-maximization (EM)
- In general EM can be interpreted as maximizing a lower bound to the complete-data loglikelihood
- EM is guaranteed to converge, but it may run into local maxima

