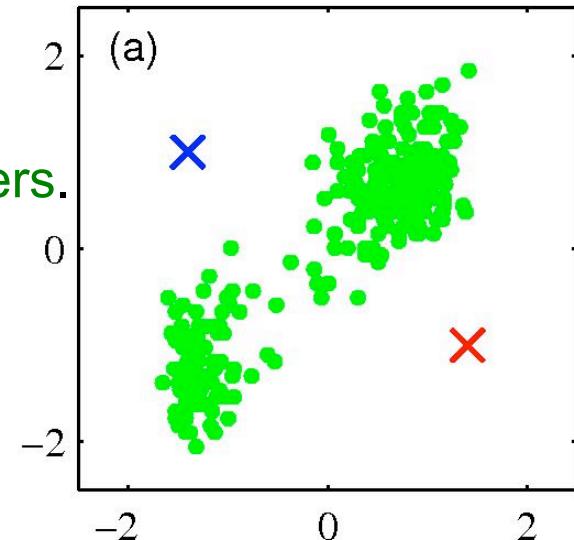


# Mixture Models

- We will look at the mixture models, including **Gaussian mixture** models and **mixture of Bernoulli**.
- The key idea is to introduce **latent variables**, which allows complicated distributions to be formed from simpler distributions.
- We will see that mixture models can be interpreted in terms of having **discrete latent variables** (in a directed graphical model).
- Later in class, we will also look at the continuous latent variables.

# K-Means Clustering

- Let us first look at the following problem: **Identify clusters**, or groups, of data points in a multidimensional space.
- We observe the dataset  $\{x_1, \dots, x_N\}$  consisting of  $N$  D-dimensional observations
- We would like to **partition the data into K clusters**, where  $K$  is given.
- We next introduce D-dimensional vectors, **prototypes**,  $\mu_k, k = 1, \dots, K$ .
- We can think of  $\mu_k$  as representing cluster centers.
- Our goal:
  - Find an **assignment of data points to clusters**.
  - Sum of squared distances of each data point to its closest prototype is at the **minimum**.

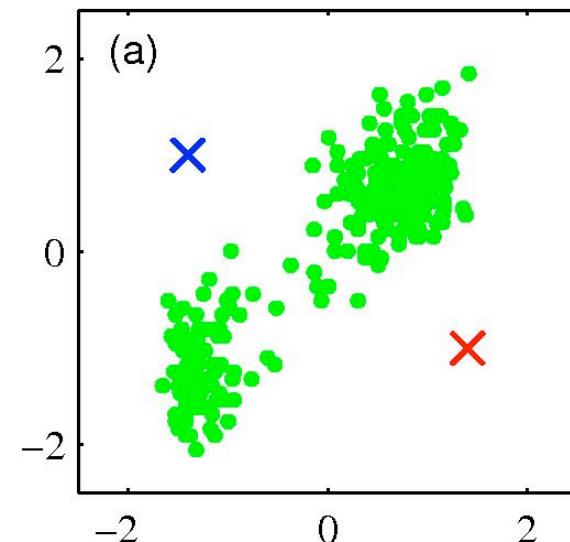


# K-Means Clustering

- For each data point  $\mathbf{x}_n$  we introduce a binary vector  $\mathbf{r}_n$  of length K (1-of-K encoding), which indicates which of the K clusters the data point  $\mathbf{x}_n$  is assigned to.
- Define objective (distortion measure):

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

- It represents the **sum of squares of the distances** of each data point to its assigned prototype  $\boldsymbol{\mu}_k$ .
- Our goal it find the values of  $r_{nk}$  and the cluster centers  $\boldsymbol{\mu}_k$  so as to minimize the objective J.



# Iterative Algorithm

- Define iterative procedure to minimize:

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

Hard assignments of points to clusters.

- Given  $\boldsymbol{\mu}_k$ , minimize J with respect to  $r_{nk}$  (**E-step**):

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



which simply says assign  $n^{\text{th}}$  data point  $\mathbf{x}_n$  to its closest cluster center.

- Given  $r_{nk}$ , minimize J with respect to  $\boldsymbol{\mu}_k$  (**M-step**):

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

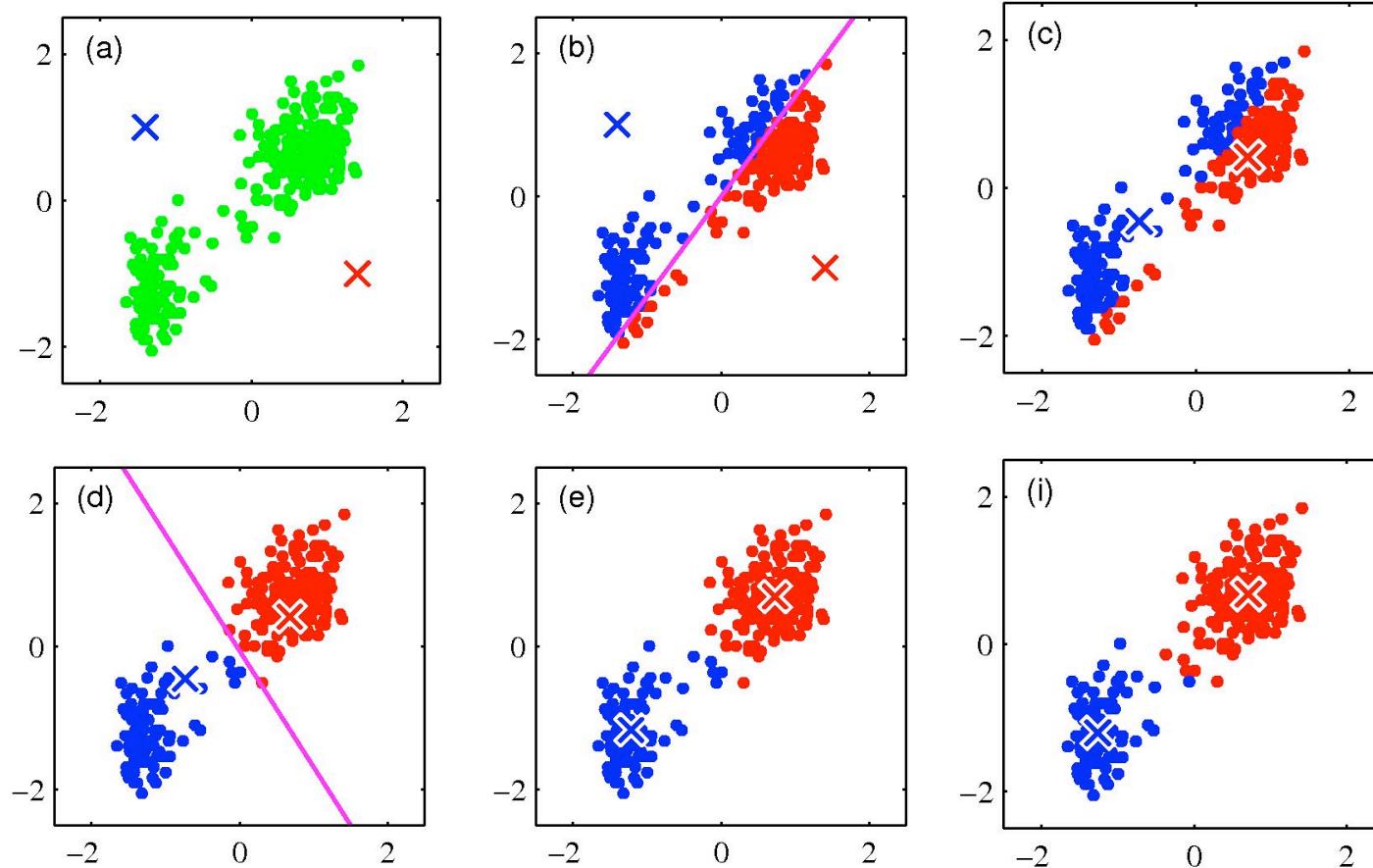
Number of points assigned to cluster k.

Set  $\boldsymbol{\mu}_k$  equal to the mean of all the data points assigned to cluster k.

- Guaranteed convergence to local minimum (not global minimum).

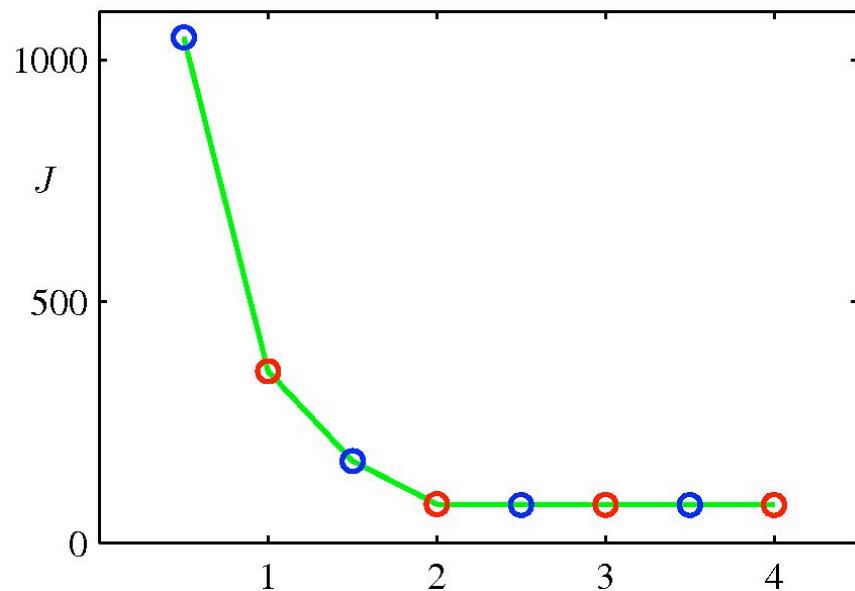
# Example

- Example of using K-means ( $K=2$ ) on Old Faithful dataset.



# Convergence

- Plot of the cost function after each E-step (blue points) and M-step (red points)



The algorithm has converged after 3 iterations.

- K-means can be generalized by introducing a **more general dissimilarity measure:**

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} K(\mathbf{x}_n, \boldsymbol{\mu}_k).$$

# Image Segmentation

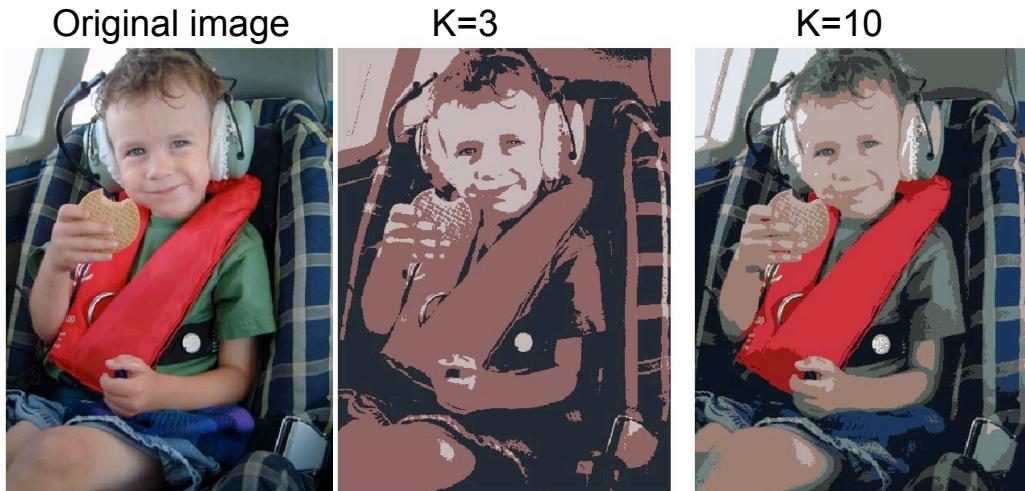
- Another application of K-means algorithm.
- Partition an image into regions corresponding, for example, to object parts.
- Each pixel in an image is a point in 3-D space, corresponding to R,G,B channels.



- For a given value of K, the algorithm represent an image using K colors.
- Another application is image compression.

# Image Compression

- For each data point, we store only the **identity k** of the assigned cluster.
- We also **store the values of the cluster centers**  $\mu_k$ .
- Provided  $K \ll N$ , we require significantly less data.



- The original image has  $240 \times 180 = 43,200$  pixels.
- Each pixel contains  $\{R,G,B\}$  values, each of which requires 8 bits.

- Requires  $43,200 \times 24 = 1,036,800$  bits to transmit directly.
- With K-means, we need to transmit K **code-book vectors**  $\mu_k$  --  $24K$  bits.
- For each pixel we need to transmit  $\log_2 K$  bits (as there are K vectors).
- **Compressed image** requires 43,248 ( $K=2$ ), 86,472 ( $K=3$ ), and 173,040 ( $K=10$ ) bits, which amounts to compression ratios of 4.2%, 8.3%, and 16.7%.

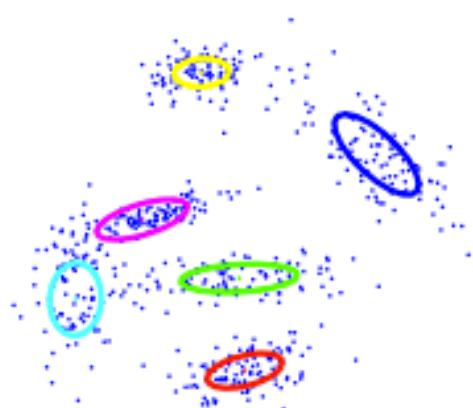
# Mixture of Gaussians

- We will look at mixture of Gaussians in terms of **discrete latent variables**.
- The Gaussian mixture can be written as a linear **superposition of Gaussians**:

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

- Introduce K-dimensional **binary random variable  $\mathbf{z}$**  having a 1-of-K representation:

$$z_k \in \{0, 1\}, \quad \sum_k z_k = 1.$$



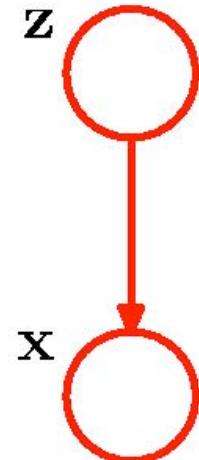
- We will specify the distribution over  $\mathbf{z}$  in terms of mixing coefficients:

$$p(z_k = 1) = \pi_k, \quad 0 \leq \pi_k \leq 1, \quad \sum_k \pi_k = 1.$$

# Mixture of Gaussians

- Because  $\mathbf{z}$  uses **1-of-K encoding**, we have:

$$p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}.$$



- We can now specify the conditional distribution:

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \text{ or } p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^K \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}.$$

- We have therefore specified the joint distribution:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}).$$

- The **marginal distribution** over  $\mathbf{x}$  is given by:

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

- The marginal distribution over  $\mathbf{x}$  is given by a **Gaussian mixture**.

# Mixture of Gaussians

- The marginal distribution:

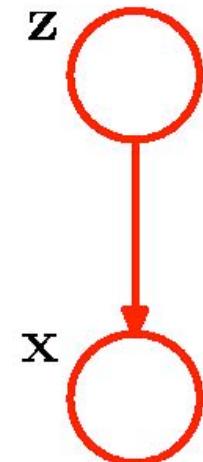
$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

- If we have several observations  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , it follows that for **every observed data point  $\mathbf{x}_n$** , there is a corresponding **latent variable  $\mathbf{z}_n$** .
- Let us look at the conditional  $p(\mathbf{z}|\mathbf{x})$ , responsibilities, which we will need for doing inference:

$$\gamma(z_k) = p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x}|z_j = 1)} =$$


  
 responsibility that component k takes for explaining the data x

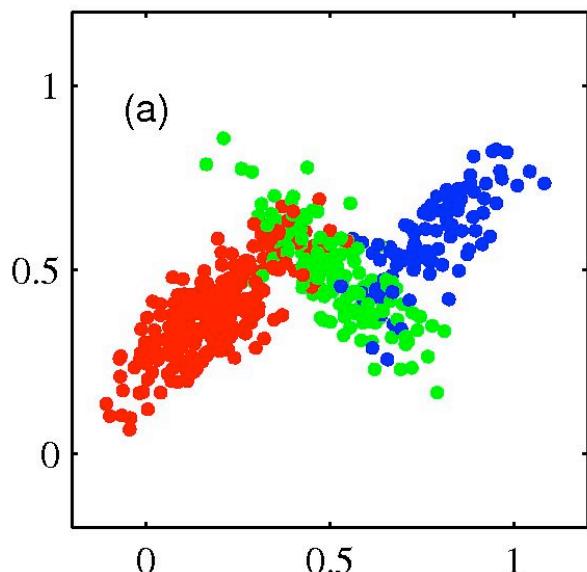
$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$



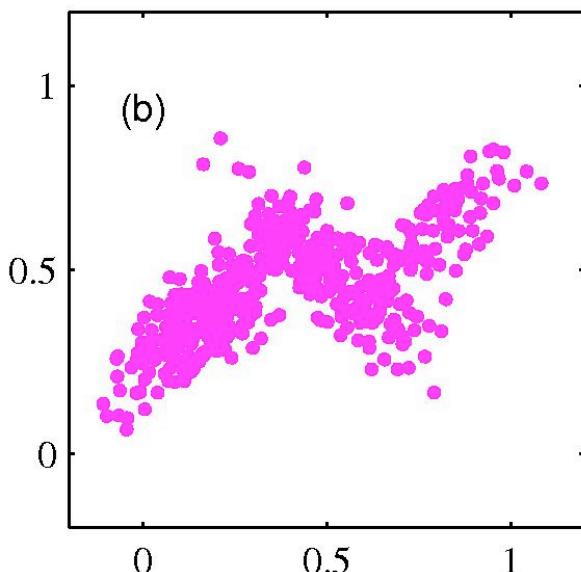
- We will view  $\pi_k$  as **prior probability** that  $z_k=1$ , and  $\gamma(z_k)$  is the **corresponding posterior** once we have observed the data.

# Example

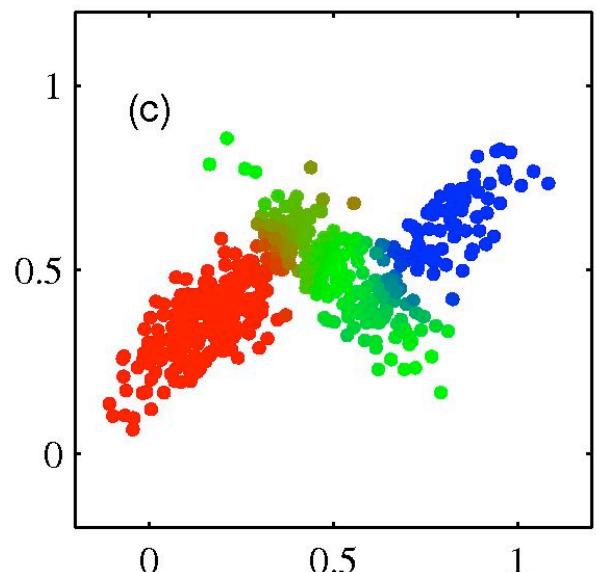
- 500 points drawn from a mixture of 3 Gaussians.



Samples from the **joint distribution**  $p(x,z)$ .



Samples from the **marginal distribution**  $p(x)$ .



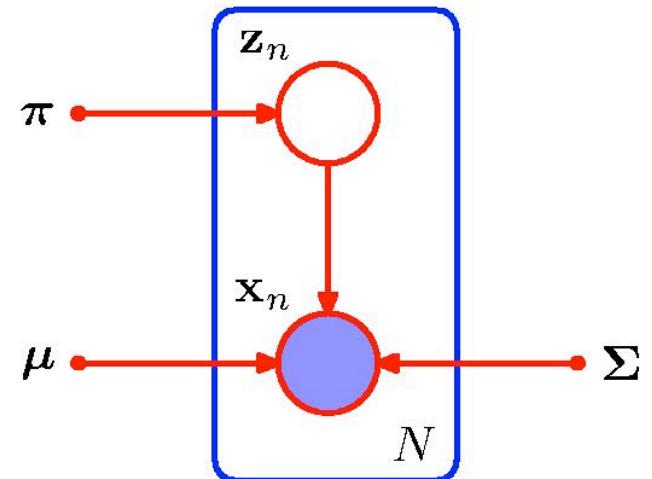
Same samples where colors represent the value of responsibilities.

# Maximum Likelihood

- Suppose we observe a dataset  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , and we model the data using mixture of Gaussians.
- We represent the dataset as an  $N$  by  $D$  matrix  $\mathbf{X}$ .
- The corresponding **latent variables** will be represented and an  $N$  by  $K$  matrix  $\mathbf{Z}$ .
- The log-likelihood takes form:

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

↑  
Model parameters



Graphical model for a Gaussian mixture model for a set of i.i.d. data point  $\{\mathbf{x}_n\}$ , and corresponding latent variables  $\{z_n\}$ .

# Maximum Likelihood

- The log-likelihood:

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

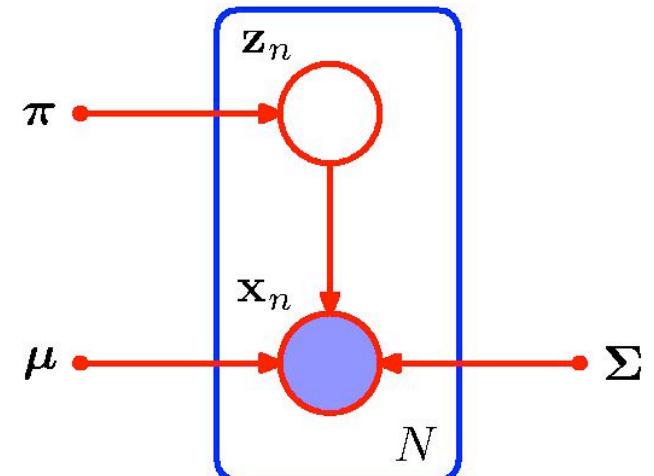
- Differentiating with respect to  $\boldsymbol{\mu}_k$  and setting to zero:

$$0 = \sum_n \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k).$$

$\gamma(z_{nk})$

Soft assignment

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_n \gamma(z_{nk}) \mathbf{x}_n, \quad N_k = \sum_n \gamma(z_{nk}).$$



- We can interpret  $N_k$  as effective number of points assigned to cluster  $k$ .
- The mean  $\boldsymbol{\mu}_k$  is given by the mean of all the data points weighted by the posterior  $\gamma(z_{nk})$  that component  $k$  was responsible for generating  $\mathbf{x}_n$ .

# Maximum Likelihood

- The log-likelihood:

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

- Differentiating with respect to  $\boldsymbol{\Sigma}_k$  and setting to zero:

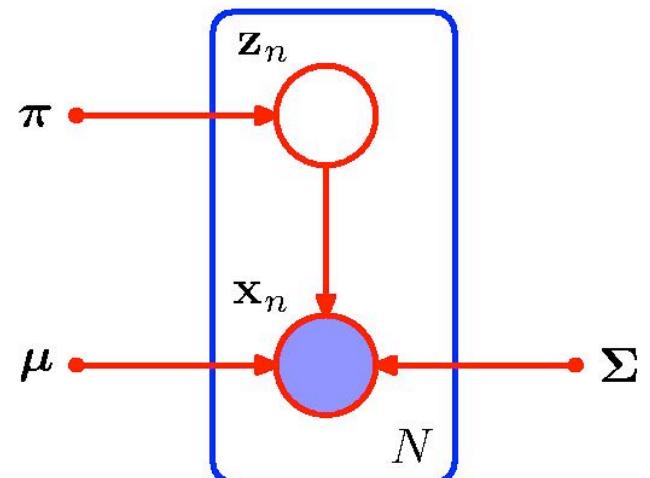
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T.$$

- Note that the data points are weighted by the posterior probabilities.

- Maximizing log-likelihood with respect to mixing proportions:

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

- Mixing proportion for the  $k^{\text{th}}$  component is given by the average responsibility which that component takes for explaining the data.



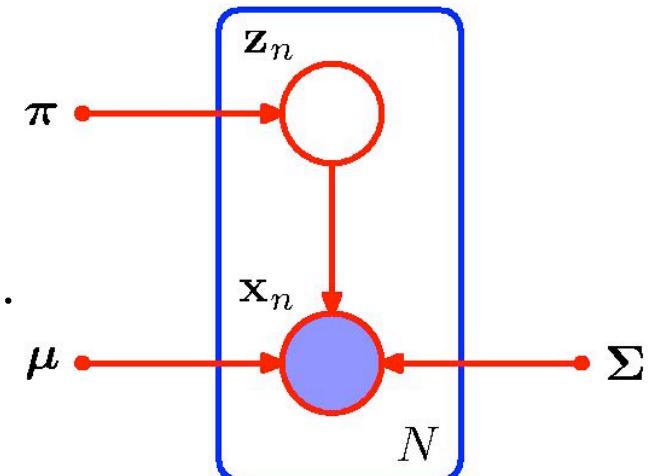
# Maximum Likelihood

- The log-likelihood:

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

- Note that the maximum likelihood **does not have a closed form solution**.
- Parameter updates depend on responsibilities  $\gamma(z_{nk})$ , which themselves depend on those parameters:

$$\gamma(z_{nk}) = p(z_{nk} = 1 | \mathbf{x}) = \frac{\pi_k N(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j N(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$



- Iterative Solution:

**E-step:** Update responsibilities  $\gamma(z_{nk})$ .

**M-step:** Update model parameters  $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ , for  $k=1, \dots, K$ .

# EM algorithm

- Initialize the means  $\mu_k$ , covariances  $\Sigma_k$ , and mixing proportions  $\pi_k$ .
- E-step: Evaluate responsibilities using current parameter values:

$$\gamma(z_{nk}) = p(z_{nk} = 1 | \mathbf{x}) = \frac{\pi_k N(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j N(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

- M-step: Re-estimate model parameters using the current responsibilities:

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_n \gamma(z_{nk}) \mathbf{x}_n, \quad N_k = \sum_n \gamma(z_{nk}),$$

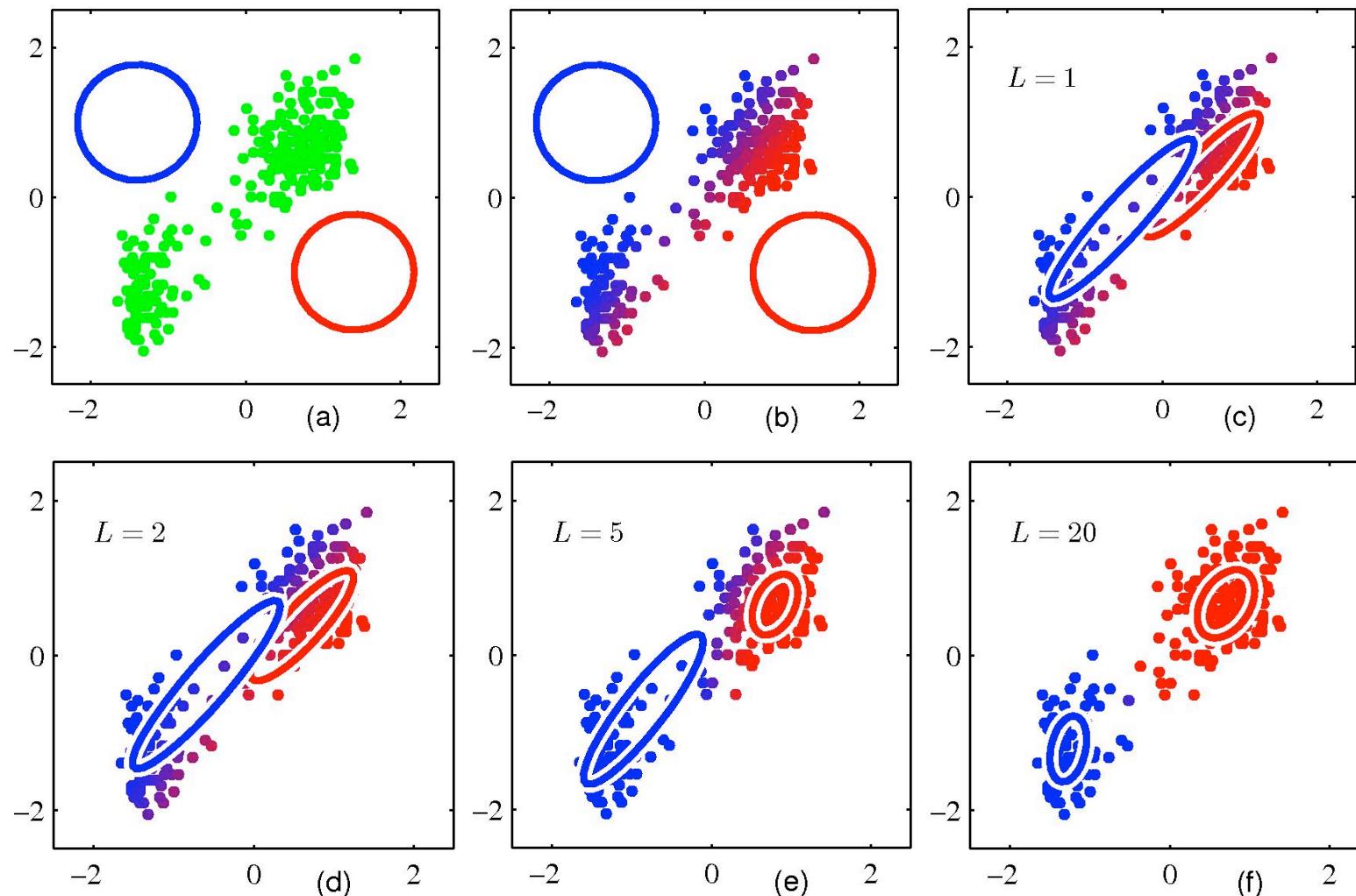
$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T,$$

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

- Evaluate the log-likelihood and check for convergence.

# Mixture of Gaussians: Example

- Illustration of the EM algorithm (much slower convergence compared to K-means)



# An Alternative View of EM

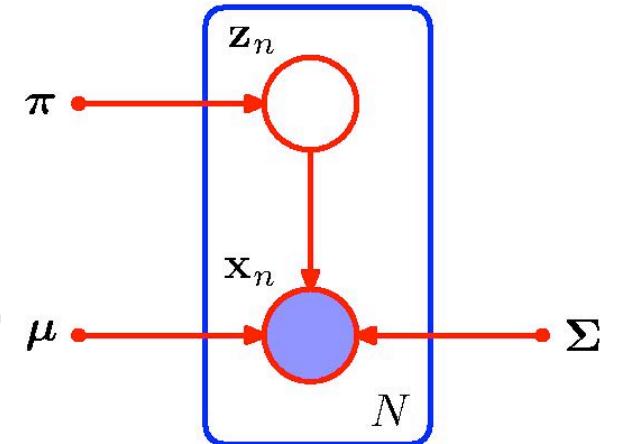
- The goal of EM is to **find maximum likelihood solutions** for models with latent variables.
- We represent the **observed dataset** as an N by D matrix  $\mathbf{X}$ .
- Latent variables will be represented and an N by K matrix  $\mathbf{Z}$ .
- The set of all **model parameters** is denoted by  $\theta$ .
- The log-likelihood takes form:

$$\ln p(\mathbf{X}|\theta) = \ln \left[ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) \right].$$

- Note: even **if the joint distribution belongs to exponential family**, the marginal typically does not!
- We will call:

$\{\mathbf{X}, \mathbf{Z}\}$  as **complete dataset**.

$\{\mathbf{X}\}$  as **incomplete dataset**.



# An Alternative View of EM

- In practice, we are **not given a complete dataset  $\{\mathbf{X}, \mathbf{Z}\}$** , but only incomplete dataset  $\{\mathbf{X}\}$ .
- Our knowledge about the latent variables is given only by the posterior distribution  $p(\mathbf{Z}|\mathbf{X}, \theta)$ .
- Because we cannot use the complete data log-likelihood, we can consider **expected complete-data log-likelihood**:

$$Q(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\theta).$$

 May seem ad-hoc.

- In the E-step, we use the current parameters  $\theta^{old}$  to compute **the posterior over the latent variables  $p(\mathbf{Z}|\mathbf{X}, \theta^{old})$** .
- We use this posterior to compute expected complete log-likelihood.
- In the M-step, we find the revised parameter estimate  $\theta^{new}$  by maximizing the **expected complete log-likelihood**:

$$\theta^{new} = \arg \max_{\theta} Q(\theta, \theta^{old}).$$

 Tractable

# The General EM algorithm

- Given a joint distribution  $p(\mathbf{Z}, \mathbf{X}|\theta)$  over observed and latent variables governed by parameters  $\theta$ , the goal is **to maximize the likelihood function**  $p(\mathbf{X}|\theta)$  with respect to  $\theta$ .
- Initialize parameters  $\theta^{old}$ .
- E-step:** Compute posterior over latent variables:  $p(\mathbf{Z}|\mathbf{X}, \theta^{old})$ .
- M-step:** Find the new estimate of parameters  $\theta^{new}$ :

$$\theta^{new} = \arg \max_{\theta} Q(\theta, \theta^{old}).$$

where

$$Q(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\theta).$$

- Check for convergence of either log-likelihood or the parameter values.

Otherwise:

$$\theta^{new} \leftarrow \theta^{old}, \quad \text{and iterate.}$$

- We will next show that **each step of EM algorithm maximizes the log-likelihood function.**

# Variational Bound

- Given a joint distribution  $p(\mathbf{Z}, \mathbf{X}|\theta)$  over observed and latent variables governed by parameters  $\theta$ , the goal is to **maximize the likelihood function**  $p(\mathbf{X}|\theta)$  with respect to  $\theta$ :

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta).$$

- We will assume that  $\mathbf{Z}$  is **discrete**, although derivations are identical if  $\mathbf{Z}$  contains continuous, or a combination of discrete and continuous variables.
- For any distribution  $q(\mathbf{Z})$  over latent variables we can derive the following **variational lower bound**:

$$\ln p(\mathbf{X}|\theta) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) = \ln \sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}$$

Jensen's inequality 

$$\geq \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})} = \mathcal{L}(q, \theta).$$

# Variational Bound

- Variational lower-bound:

$$\begin{aligned}\ln p(\mathbf{X}|\theta) &= \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) = \ln \sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})} \\ &\geq \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})} \\ &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln p(\mathbf{X}, \mathbf{Z}|\theta) + \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{1}{q(\mathbf{Z})} \\ &= \mathbb{E}_{q(\mathbf{Z})} [\ln p(\mathbf{X}, \mathbf{Z}|\theta)] + \mathcal{H}(q(\mathbf{Z})) = \mathcal{L}(q, \theta).\end{aligned}$$



Expected complete  
log-likelihood



Entropy functional. Variational lower-  
bound

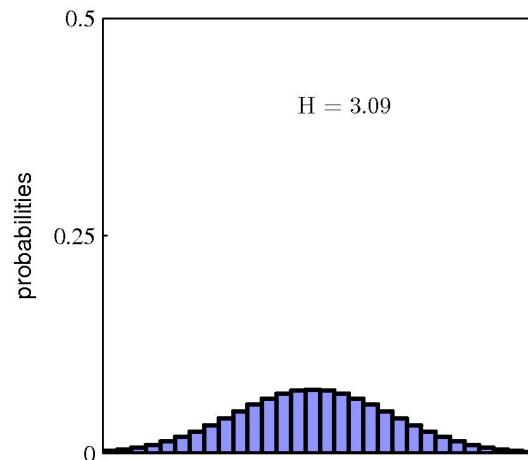
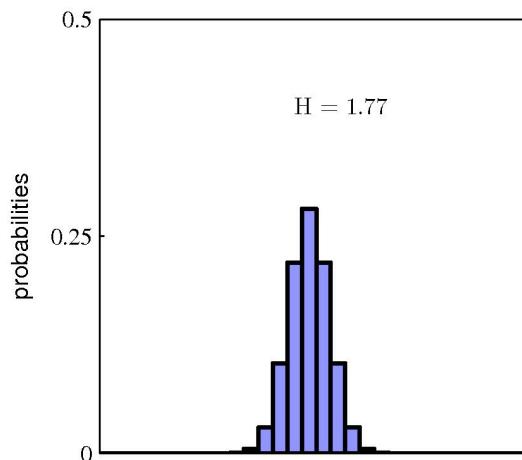


# Entropy

- For a discrete random variable  $X$ , where  $P(X=x_i) = p(x_i)$ , the entropy of a random variable is:

$$\mathcal{H}(p) = - \sum_i p(x_i) \log p(x_i).$$

- Distributions that are sharply picked around a few values will have a relatively low entropy, whereas those that are spread more evenly across many values will have higher entropy



- Histograms of two probability distributions over 30 bins.
  - The largest entropy will arise from a uniform distribution  $H = -\ln(1/30) = 3.40$ .

- For a density defined over continuous random variable, the differential entropy is given by:  $\mathcal{H}(p) = - \int p(x) \log p(x) dx.$

# Variational Bound

- We saw:

$$\ln p(\mathbf{X}|\theta) \geq \mathbb{E}_{q(\mathbf{Z})} [\ln p(\mathbf{X}, \mathbf{Z}|\theta)] + \mathcal{H}(q(\mathbf{Z})) = \mathcal{L}(q, \theta).$$

- We also note that the following decomposition also holds:

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q, \theta) + \text{KL}(q||p),$$

where

$$\mathcal{L}(q, \theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})},$$

Variational lower-bound

$$\text{KL}(q||p) = - \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})}.$$

Kullback-Leibler (KL) divergence.

Also known as  
Relative Entropy.

- KL divergence is **not symmetric**.
- $\text{KL}(q||p) \geq 0$  with equality iff  $p(\mathbf{x}) = q(\mathbf{x})$ .
- Intuitively, it measures the “**distance**” between the two distributions.

# Variational Bound

- Let us derive that:

$$\log p(\mathbf{X}|\theta) = \mathcal{L}(q, \theta) + \text{KL}(q||p),$$

- We can write:

$$\ln p(\mathbf{X}, \mathbf{Z}|\theta) = \ln p(\mathbf{Z}|\mathbf{X}, \theta) + \ln p(\mathbf{X}|\theta),$$

and plugging into the definition of  $\mathcal{L}(q, \theta)$ , gives the desired result.

- Note that **variational bound becomes tight iff  $q(\mathbf{Z}) = p(\mathbf{Z} | \mathbf{X}, \theta)$ .**
- In other words the distribution  $q(\mathbf{Z})$  is **equal to the true posterior** distribution over the latent variables, so that  $\text{KL}(q||p) = 0$ .
- As  $\text{KL}(q||p) \geq 0$ , it immediately follows that:

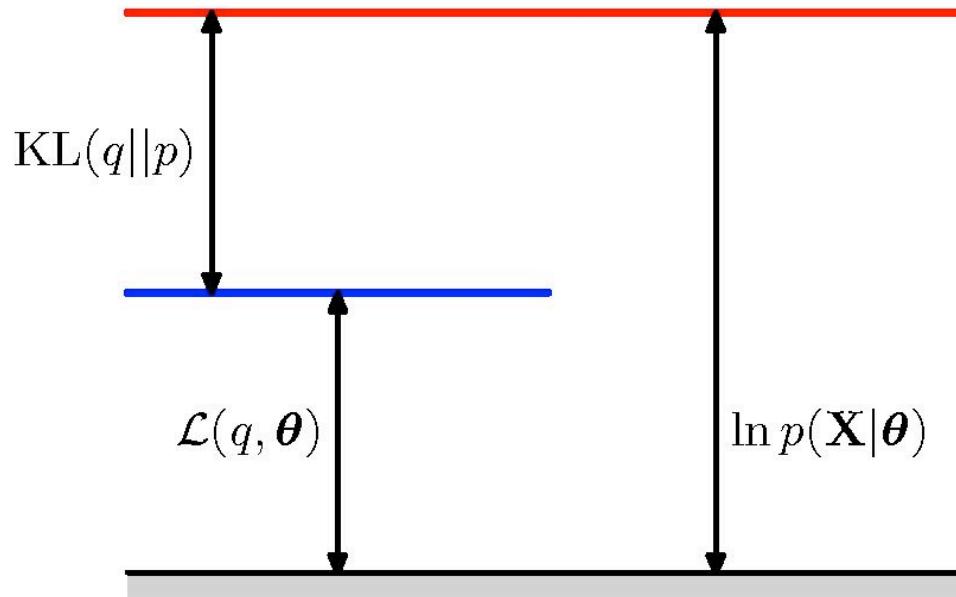
$$\ln p(\mathbf{X}|\theta) \geq \mathcal{L}(q, \theta),$$

which also showed using **Jensen's inequality**.

# Decomposition

- Illustration of the decomposition which holds for any distribution  $q(\mathbf{Z})$ .

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q, \theta) + \text{KL}(q||p),$$



# Alternative View of EM

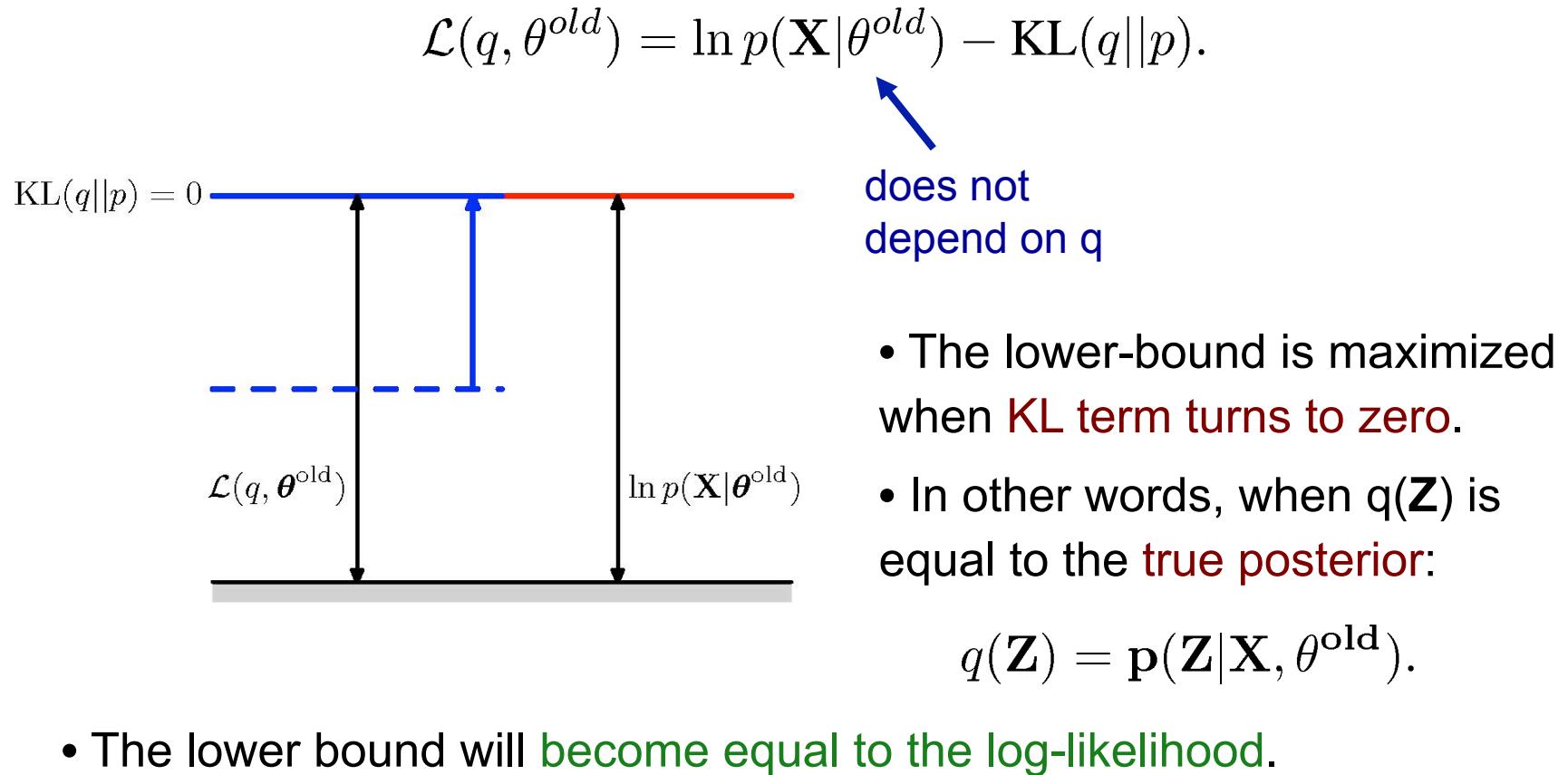
- We can use our decomposition to define the EM algorithm and show that it maximizes the log-likelihood function.

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q, \theta) + \text{KL}(q||p),$$

- Summary:
  - In the **E-step**, the lower bound  $\mathcal{L}(q, \theta)$  is maximized with respect to distribution  $q$  while holding parameters  $\theta$  fixed.
  - In the **M-step**, the lower bound  $\mathcal{L}(q, \theta)$  is maximized with respect to parameters  $\theta$  while holding the distribution  $q$  fixed.
- These steps will increase the corresponding log-likelihood.

# E-step

- Suppose that the current value of the parameter vector is  $\theta^{old}$ .
- In the E-step, we maximize the lower with respect to  $q$  while holding parameters  $\theta^{old}$  fixed.

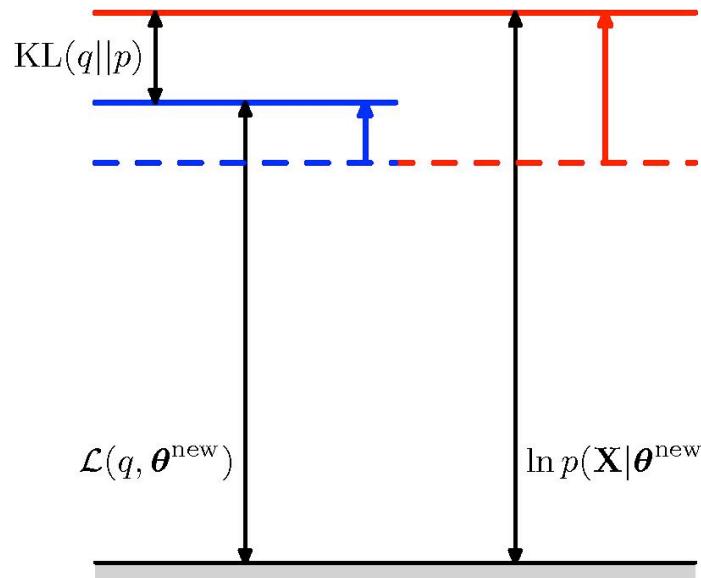


# M-step

- In the M-step, the lower bound is maximized with respect to parameters  $\theta$  while holding the distribution  $q$  fixed.

$\downarrow$  does not depend on  $\theta$ .

$$\mathcal{L}(q, \theta) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\theta) + \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \ln \frac{1}{p(\mathbf{Z}|\mathbf{X}, \theta^{old})}.$$



$$\mathcal{L}(q, \theta) = Q(\theta, \theta^{old}) + \text{const.}$$

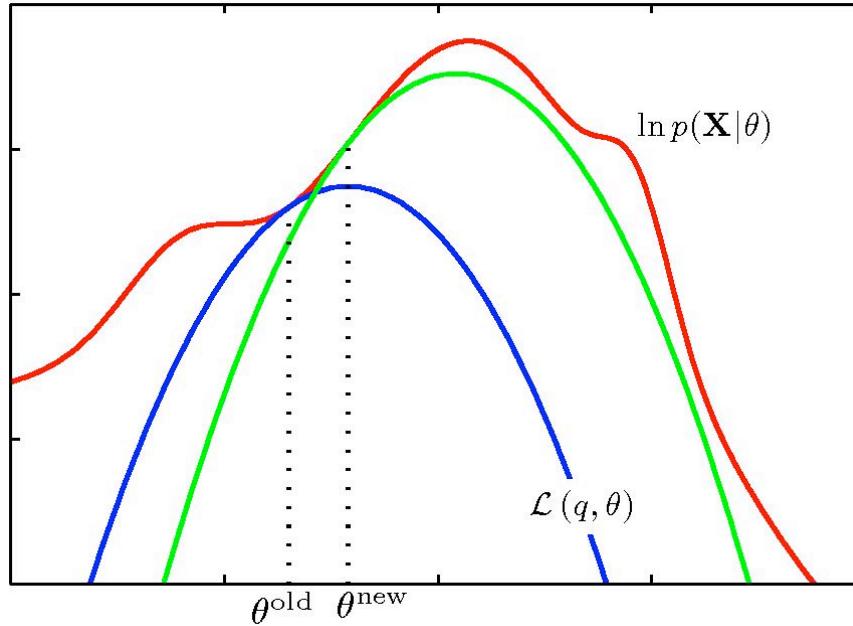
- Hence the M-step amounts to maximizing the expected complete log-likelihood.

$$\theta^{new} = \arg \max_{\theta} Q(\theta, \theta^{old}).$$

- Because KL divergence is non-negative, this causes the log-likelihood  $\log p(\mathbf{X} | \theta)$  to increase by at least as much as the lower bound does.

# Bound Optimization

- The EM algorithm belongs to the general class of bound optimization methods:



- At each step, we compute:
  - E-step: **a lower bound on the log-likelihood** function for the current parameter values. The bound is concave with unique global optimum.
  - M-step: **maximize the lower-bound** to obtain the new parameter values.

# Extensions

- For some complex problems, it maybe the case that either E-step or M-step, or both **remain intractable**.
- This leads to two possible extensions.
- The **Generalized EM** deals with intractability of the M-step.
- Instead of maximizing the lower-bound in the M-step, we instead seek to **change parameters so as to increase its value** (e.g. using nonlinear optimization, conjugate gradient, etc.).
- We can also **generalize the E-step** by performing a partial, rather than complete, optimization of the lower-bound with respect to  $q$ .
- For example, we can use an **incremental form of EM**, in which at each EM step only one data point is processed at a time.
- In the E-step, instead of recomputing the responsibilities for all the data points, we just **re-evaluate the responsibilities for one data point**, and proceed with the M-step.

# Maximizing the Posterior

- We can also use EM to **maximize the posterior**  $p(\theta | \mathbf{X})$  for models in which we have introduced the prior  $p(\theta)$ .
- To see this, note that:

$$\ln p(\theta|\mathbf{X}) = \ln p(\mathbf{X}|\theta) + \ln p(\theta) - \ln p(\mathbf{X}).$$

- Decomposing the log-likelihood into **lower-bound** and **KL** terms, we have:

$$\ln p(\mathbf{X}|\theta) = \mathcal{L}(q, \theta) + \text{KL}(q||p),$$

- Hence

$$\ln p(\theta|\mathbf{X}) = \mathcal{L}(q, \theta) + \text{KL}(q||p) + \ln p(\theta) - \ln p(\mathbf{X}).$$

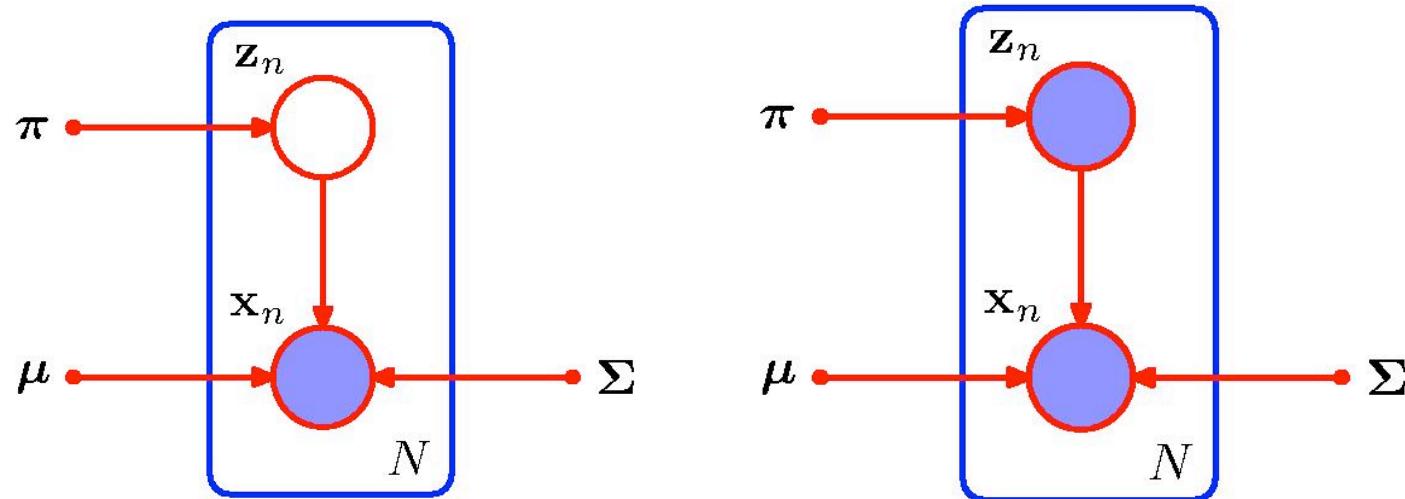
where  $\ln p(\mathbf{X})$  is a constant.

- Optimizing with respect to  $q$  gives rise to the **same E-step** as for the standard EM algorithm.
- The M-step equations are **modified through introduction of the prior** term, which typically amounts to only a small modification to the standard ML M-step equations.

# Gaussian Mixtures Revisited

- We now consider the application of the latent variable view of EM the case of **Gaussian mixture model**.
- Recall:

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



$\{\mathbf{X}\}$  -- incomplete dataset.     $\{\mathbf{X}, \mathbf{Z}\}$  -- complete dataset.

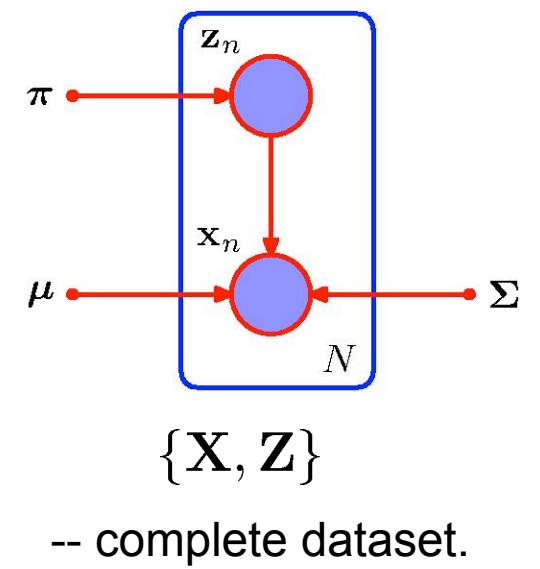
# Maximizing Complete Data

- Consider the problem of maximizing the likelihood for the complete data:

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^N \prod_{k=1}^K \left[ \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]^{z_{nk}}.$$

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^K \left[ \sum_{n=1}^N z_{nk} \ln \pi_k + z_{nk} \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right].$$

Sum of K independent contributions, one for each mixture component.



- Maximizing with respect to mixing proportions

yields:

$$\pi_k = \frac{1}{N} \sum_{n=1}^N z_{nk}.$$

- And similarly for the means and covariances.

# Posterior Over Latent Variables

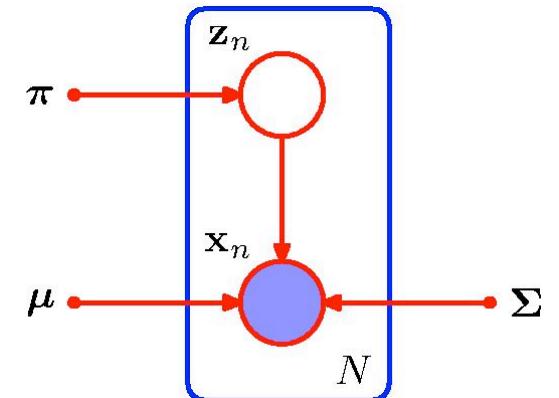
- Remember:

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^K \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}, \quad p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k}.$$

- The posterior over latent variables takes form:

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \prod_{n=1}^N \prod_{k=1}^K \left[ \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]^{z_k}.$$

- Note that the posterior factorizes over  $n$  points, so that under the posterior distribution  $\{\mathbf{z}_n\}$  are independent.
- This can be verified by inspection of directed graph and making use of the d-separation property.



# Expected Complete Log-Likelihood

- The expected value of indicator variable  $z_{nk}$  under the posterior distribution is:

$$\begin{aligned}\mathbb{E}[z_{nk}] &= \frac{\sum_{\mathbf{z}_n} z_{nk} \prod_j [\pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)]^{z_{nj}}}{\sum_{\mathbf{z}_n} \prod_j [\pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)]^{z_{nj}}} \\ &= \frac{\pi_k N(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j N(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \gamma(z_{nk}).\end{aligned}$$

- This represent **the responsibility** of component k for data point  $\mathbf{x}_n$ .
- The **complete-data log-likelihood**:

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \left[ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right].$$

- The **expected complete data log-likelihood** is:

$$\mathbb{E}_{\mathbf{Z}} [\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] = \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \left[ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right].$$

# Expected Complete Log-Likelihood

- The expected complete data log-likelihood is:

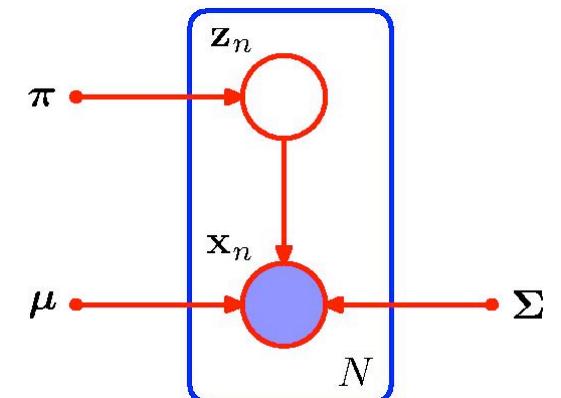
$$\mathbb{E}_{\mathbf{Z}} [\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] = \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \left[ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right].$$

- Maximizing the respect to model parameters we obtain:

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_n \gamma(z_{nk}) \mathbf{x}_n, \quad N_k = \sum_n \gamma(z_{nk}),$$

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

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



# Relationship to K-Means

- Consider a Gaussian mixture model in which covariances are shared and are given by  $\epsilon I$ .

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

- Consider EM algorithm for a mixture of K Gaussians, in which we treat  $\epsilon$  as a fixed constant. The posterior responsibilities take form:

$$\gamma(z_{nk}) = \frac{\pi_k \exp(-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon)}{\sum_{j=1}^K \pi_j \exp(-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon)}.$$

- Consider the limit  $\epsilon \rightarrow 0$ .
- In the denominator, the term for which  $\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2$  is smallest will go to zero most slowly. Hence  $\gamma(z_{nk}) \rightarrow r_{nk}$ , where

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

# Relationship to K-Means

- Consider EM algorithm for a mixture of K Gaussians, in which we treat  $\epsilon$  as a fixed constant. The **posterior responsibilities** take form:

$$\gamma(z_{nk}) = \frac{\pi_k \exp(-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon)}{\sum_{j=1}^K \pi_j \exp(-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon)}.$$

- Finally, in the limit  $\epsilon \rightarrow 0$ , the **expected complete log-likelihood** becomes:

$$\mathbb{E}_{\mathbf{Z}} [\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] \rightarrow -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 + \text{const.}$$

- Hence in the limit, maximizing the **expected complete log-likelihood** is equivalent to minimizing the distortion measure  $J$  for the K-means algorithm.

# Bernoulli Distribution

- So far we focused on distributions over continuous variables.
- We will now look at **mixture of discrete binary variables** described by **Bernoulli distributions**.
- Consider a set of binary random variables  $x_i$ ,  $i=1,\dots,D$ , each of which is governed by a Bernoulli distribution with  $\mu_i$ .

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{i=1}^D \mu_i^{x_i} (1 - \mu_i)^{1-x_i}.$$

- The **mean** and **covariance** of this distribution are:

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}, \quad \text{cov}[\mathbf{x}] = \text{diag}(\mu_i(1 - \mu_i)).$$

# Mixture of Bernoulli Distributions

- Consider a **finite mixture of Bernoulli distributions**:

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

$$p(\mathbf{x}|\boldsymbol{\mu}_k) = \prod_{i=1}^D \mu_{ki}^{x_i} (1 - \mu_{ki})^{1-x_i}.$$

- The **mean** and **covariance** of this mixture distribution are:

$$\mathbb{E}[\mathbf{x}] = \sum_{k=1}^K \pi_k \boldsymbol{\mu}_k, \quad \text{cov}[\mathbf{x}] = \sum_{k=1}^K \pi_k (\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T) - \mathbb{E}[\mathbf{x}] \mathbb{E}[\mathbf{x}]^T,$$

where  $\boldsymbol{\Sigma}_k = \text{diag}(\mu_{ki}(1 - \mu_{ki}))$ .

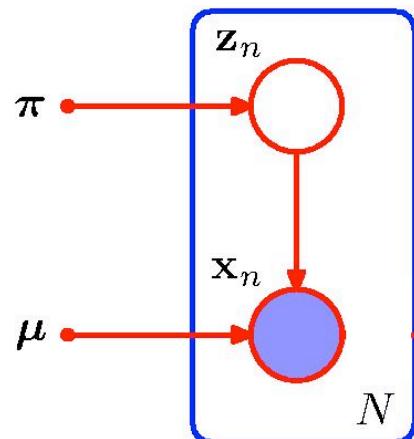
- The **covariance matrix is no longer diagonal**, so the mixture distribution can capture correlations between the variables, unlike a single Bernoulli distribution.

# Maximum Likelihood

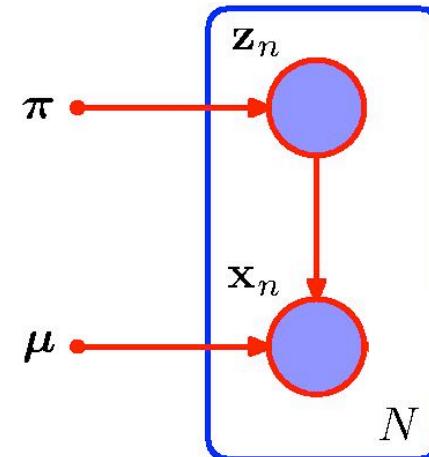
- Given a dataset  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , the log-likelihood takes form:

$$\ln p(\mathbf{X}|\pi, \mu) = \sum_{n=1}^N \ln \left[ \sum_{k=1}^K \pi_k p(\mathbf{x}| \mu_k) \right].$$

- Again, we see the sum inside the log, so the **maximum likelihood solution no longer has a closed form solution**.
- We will now derive EM for maximizing this likelihood function.



$\{\mathbf{X}\}$  -- incomplete dataset.



$\{\mathbf{X}, \mathbf{Z}\}$  -- complete dataset.

# Complete Log-Likelihood

- By introducing **latent discrete random variables**, we have:

$$p(\mathbf{z}|\boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{z_k}, \quad p(\mathbf{x}|\mathbf{z}, \boldsymbol{\mu}) = \prod_{k=1}^K p(\mathbf{x}|\boldsymbol{\mu}_k)^{z_k}.$$

- We can write down the **complete log-likelihood**

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\pi}, \boldsymbol{\mu}) = \sum_{i=1}^N \sum_{k=1}^K z_{nk} \left[ \ln \pi_k + \sum_{i=1}^D [x_{ni} \ln \mu_{ki} + (1 - x_{ni}) \ln(1 - \mu_{ki})] \right].$$

- The **expected complete-data log-likelihood**:

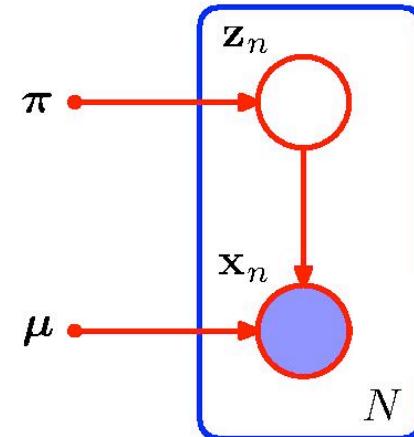
$$\mathbb{E}_{\mathbf{Z}} \left[ \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\pi}, \boldsymbol{\mu}) \right] = \sum_{i=1}^N \sum_{k=1}^K \gamma(z_{nk}) \left[ \ln \pi_k + \sum_{i=1}^D [x_{ni} \ln \mu_{ki} + (1 - x_{ni}) \ln(1 - \mu_{ki})] \right],$$

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

# E-step

- Similar to the mixture of Gaussians, in the E-step, we evaluate responsibilities using Bayes' rule:

$$\begin{aligned}\mathbb{E}[z_{nk}] &= \frac{\sum_{\mathbf{z}_n} z_{nk} \prod_k [\pi_{k'} p(\mathbf{x}_n | \boldsymbol{\mu}_{k'})]^{z_{nk'}}}{\sum_{\mathbf{z}_n} \prod_j [\pi_j p(\mathbf{x}_n | \boldsymbol{\mu}_j)]^{z_{nj}}} \\ &= \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_n | \boldsymbol{\mu}_j)} = \gamma(z_{nk}).\end{aligned}$$



# M-step

- The expected complete-data log-likelihood:

$$\mathbb{E}_{\mathbf{Z}} \left[ \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\pi}, \boldsymbol{\mu}) \right] = \sum_{i=1}^N \sum_{k=1}^K \gamma(z_{nk}) \left[ \ln \pi_k + \sum_{i=1}^D [x_{ni} \ln \mu_{ki} + (1-x_{ni}) \ln (1-\mu_{ki})] \right],$$

- Maximizing the expected complete-data log-likelihood:

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

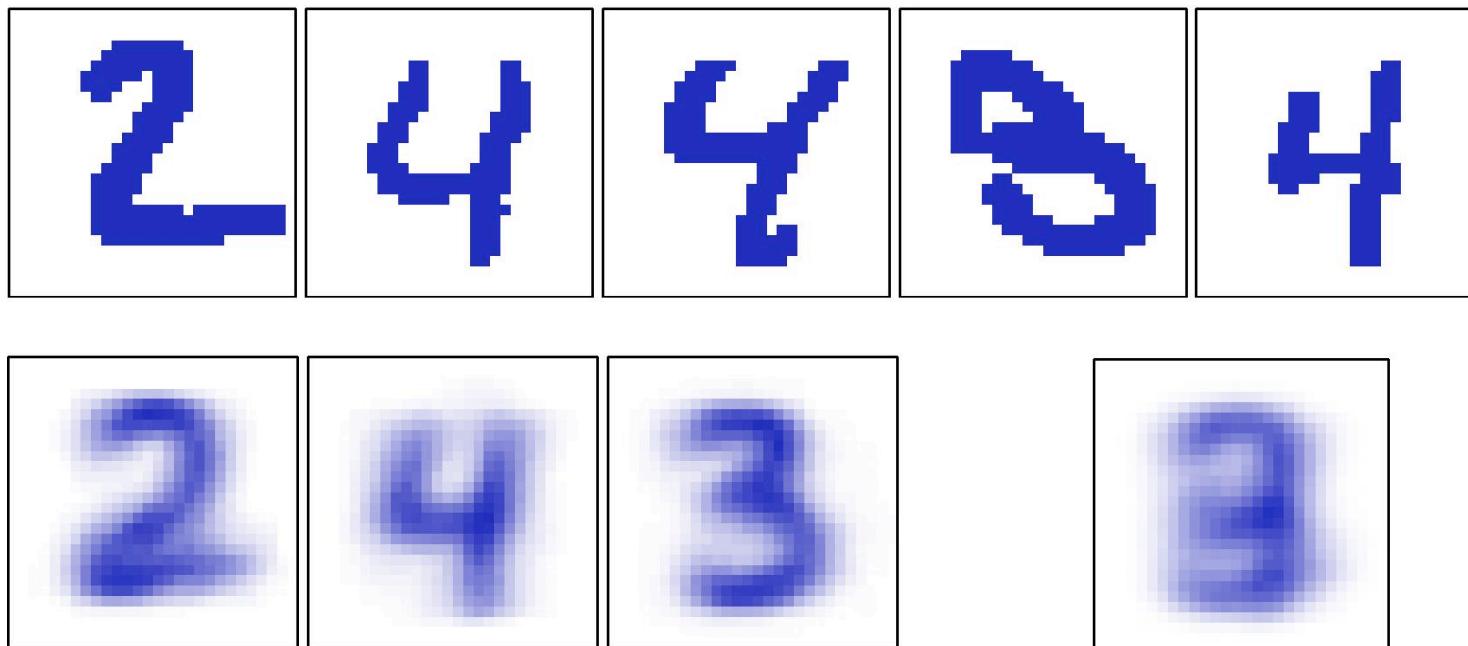
where  $N_k$  is the **effective number of data points** associated with component  $k$ .

- Note that the mean of component  $k$  is equal to the weighted mean of the data, with weights given by the responsibilities that component  $k$  takes for explaining the data points.

# Example

- Illustration of the Bernoulli mixture model

Training data



Learned  $\mu_k$  for the first three components.

A single multinomial Bernoulli distribution fit to the full data.