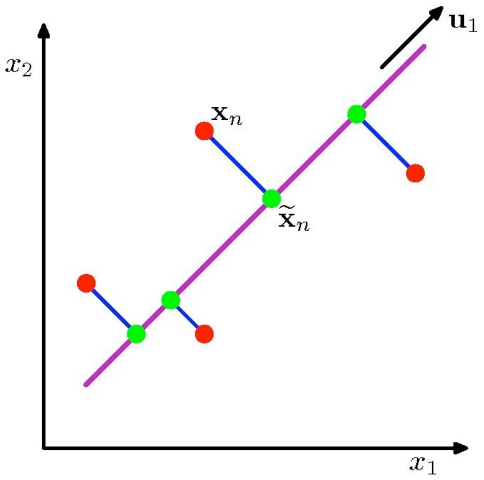
STA414/2104

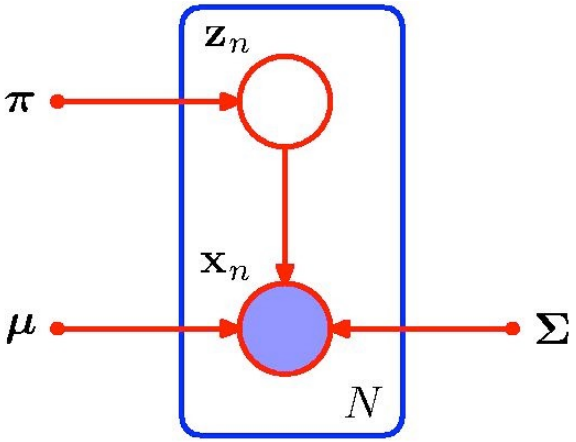
Statistical Methods for Machine Learning II

Murat A. Erdogdu

Department of Computer Science
Department of Statistical Sciences



Lecture 8



UNIVERSITY OF TORONTO

Announcements

- HW 3 is due next week. TA Ohs announced.
- Midterm results are released on Crowdmark.
- HW2 results are released on Crowdmark.

- Final exam will be on 4/20, at 9am EST.

Last time

- SVMs
- Decision trees
- Ensembles

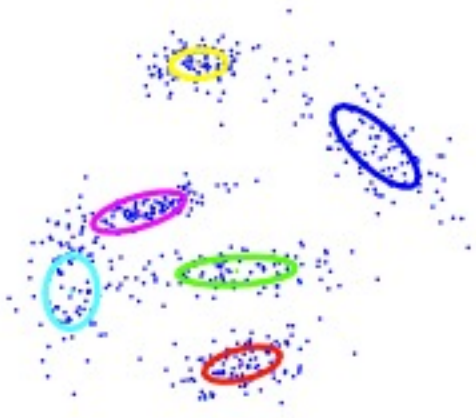
Today

- Unsupervised learning
- Mixture models
- K-means
- EM Algorithm

Unsupervised Learning

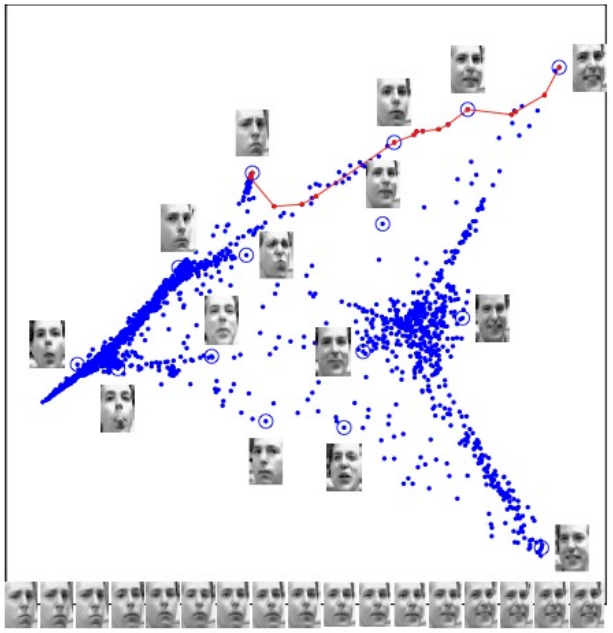
The goal is to construct statistical model that finds useful representation of data:

- Clustering
- Dimensionality reduction
- Modeling the data density
- Finding hidden causes (useful explanation) of the data



Unsupervised Learning can be used for:

- Structure discovery
- Anomaly detection / Outlier detection
- Data compression, Data visualization
- Used to aid classification/regression tasks (i.e. pre-processing)



Mixture Models

- We will look at the mixture models, including Gaussian mixtures.

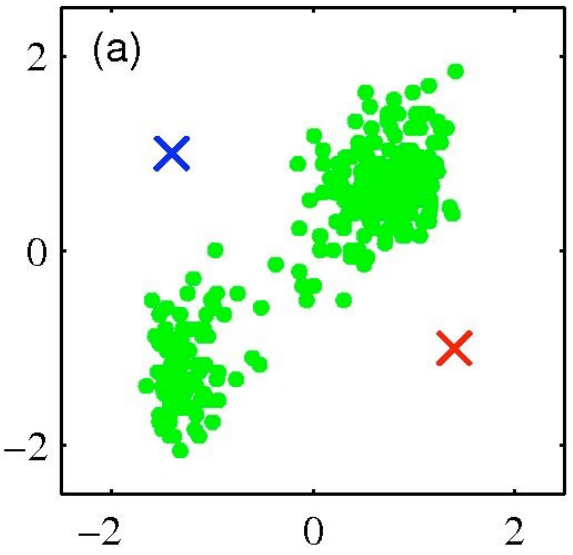
- The key idea is to introduce latent variables, which allows complicated distributions to be formed from simpler distributions.

- We will see that mixture models can be interpreted in terms of having discrete latent variables.

- Later, we will also see 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 $\{\mathbf{x_1}, ..., \mathbf{x}_N\}$ 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, $\boldsymbol{\mu}_k, k = 1, ..., K.$
- We can think of $\boldsymbol{\mu}_k,$ 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.

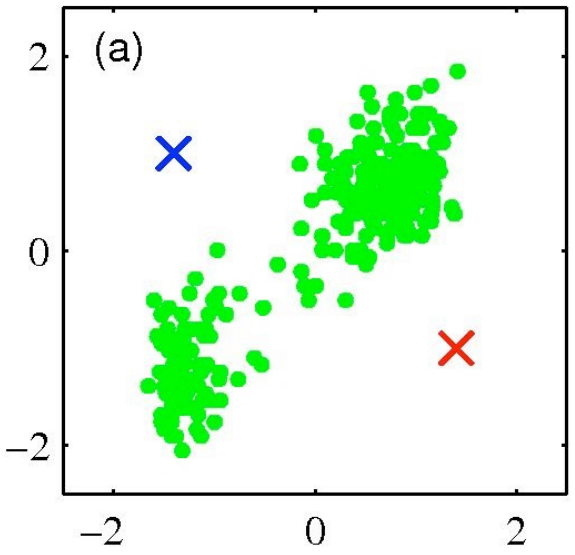
$$\mathbf{r}_n = [0, 0, 1, 0, 0]^T$$

• 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 is to 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:

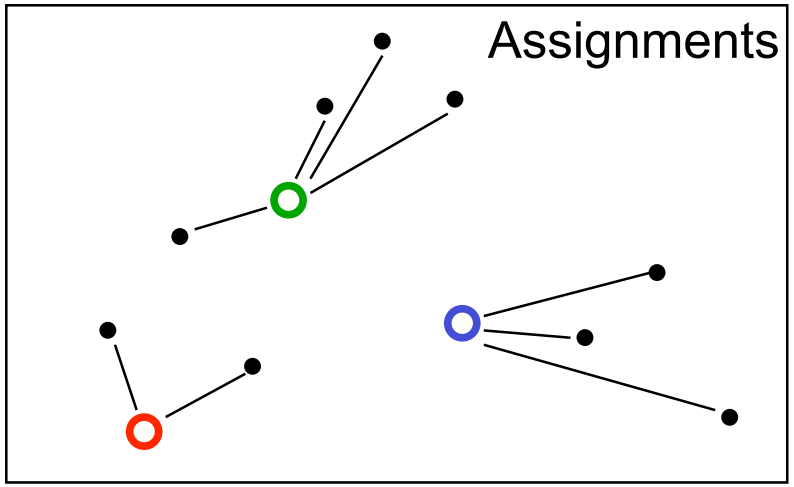
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2.$$

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

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

Hard assignments of points to clusters.

which simply says assign n^{th} data point x_n to its closest cluster center.



Iterative Algorithm

- Define iterative procedure to minimize:

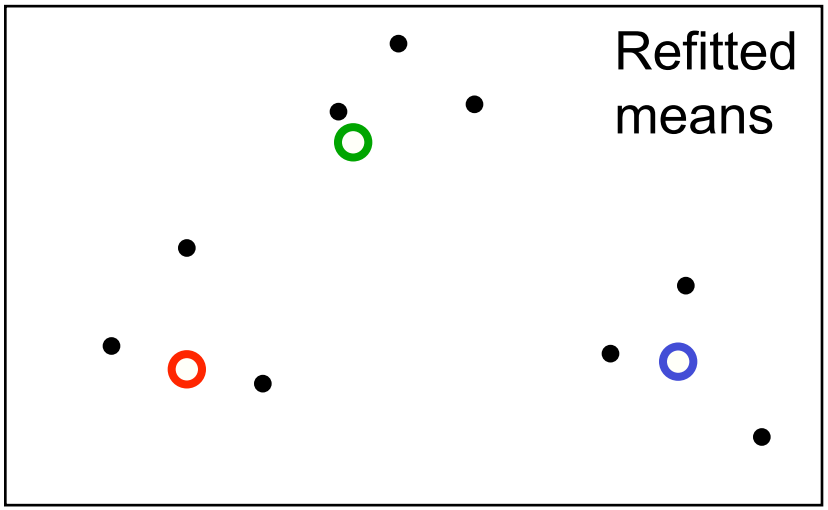
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2.$$

- 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.



Iterative Algorithm

- Define iterative procedure to minimize:

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

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

Hard assignments of points to clusters.

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

which simply says assign n^{th} data point x_n to its closest cluster center.

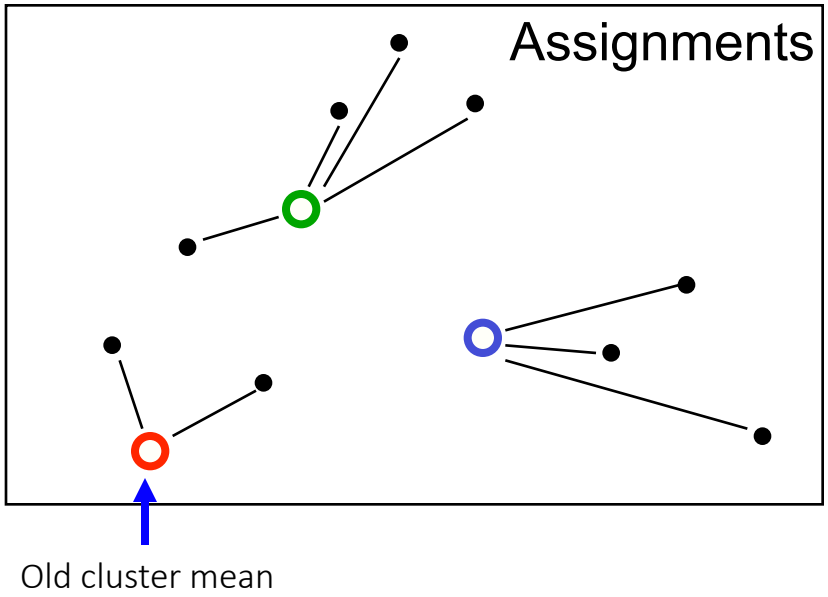
- Given r_{nk}, minimize J with respect to $\boldsymbol{\mu}_k$, (M-step):

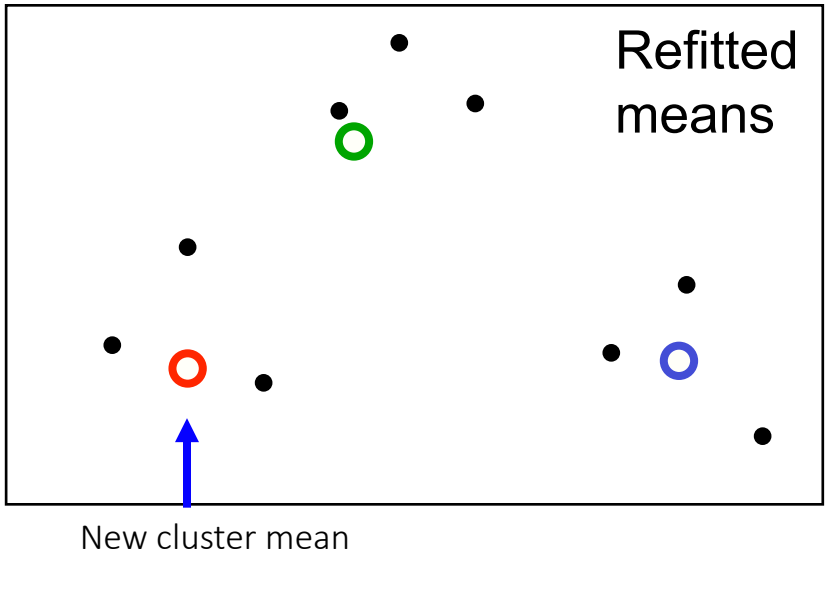
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}.$$

Number of points assigned to cluster k.

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

Iterative Algorithm

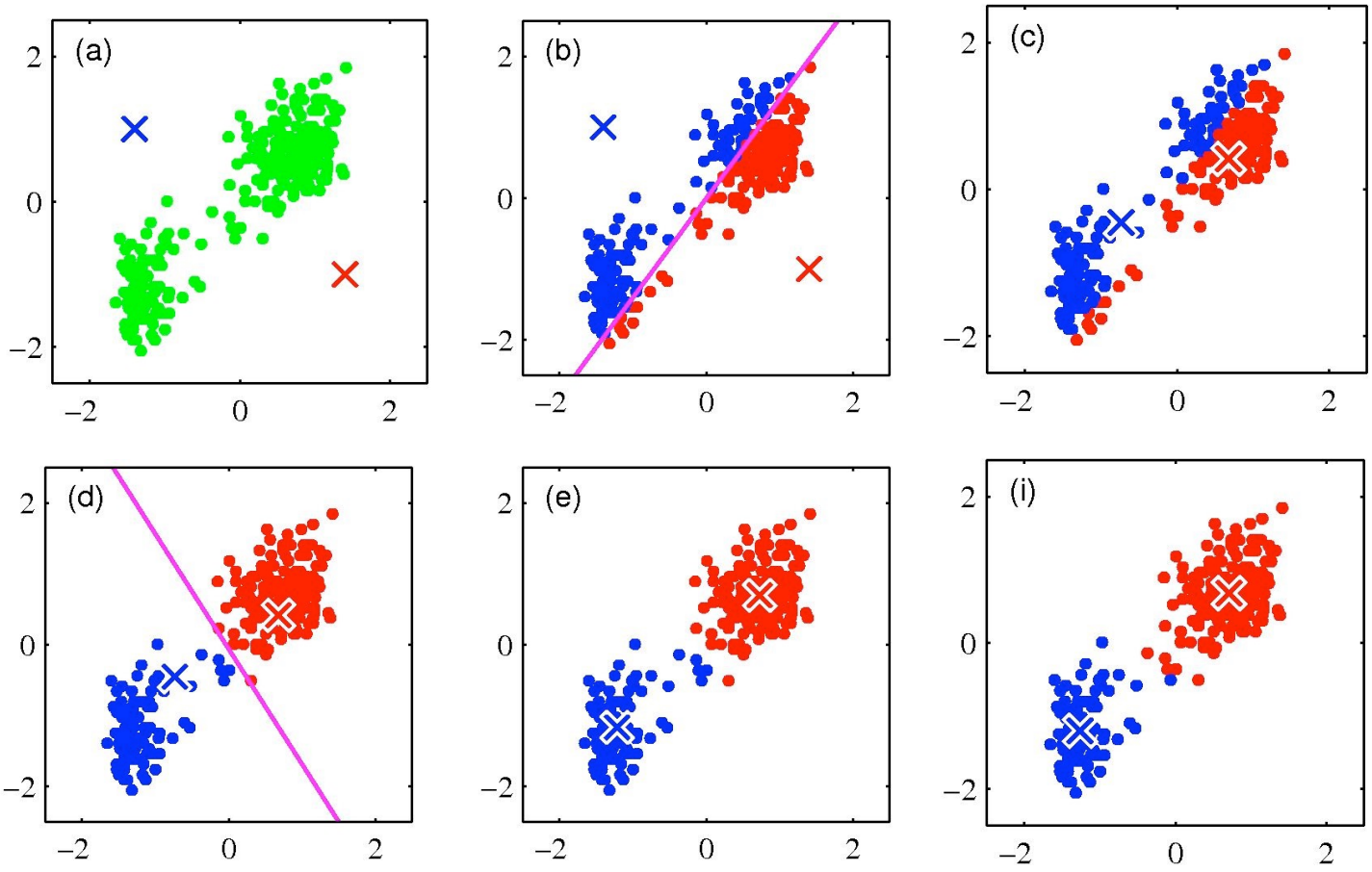




- Guaranteed convergence to local minimum (not global minimum).

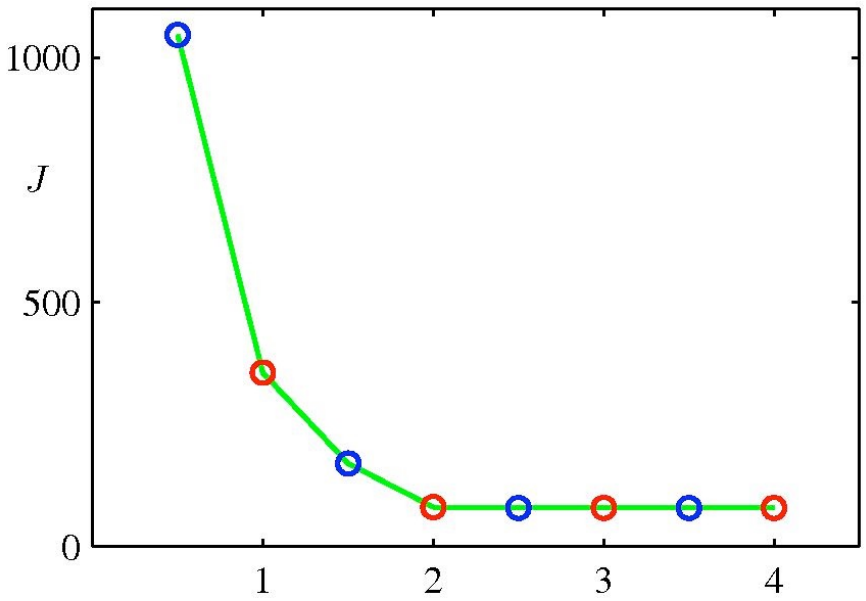
Example

- 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)



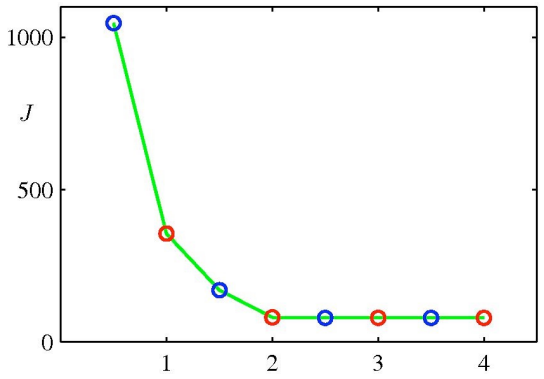
The algorithm has converged after 3 iterations.

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

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

K-Means Algorithm

- K-means algorithm reduces the cost at each iteration.
  - Whenever an assignment is changed, the sum of squared distances J of data points from their assigned centers is reduced.
  - Whenever a cluster center is moved, J is reduced.



- Test for convergence: If the assignments do not change in the assignment step, algorithm has converged (to at least a local minimum).
- This will always happen after a finite number of iterations, since the number of possible cluster assignments is finite

K-Means Algorithm

- The objective J is non-convex (so coordinate descent on J is not guaranteed to converge to the global minimum)
- There is no guarantee that k-means is not getting stuck at a bad local minima.
- We could try many random starting points.

A bad local optimum

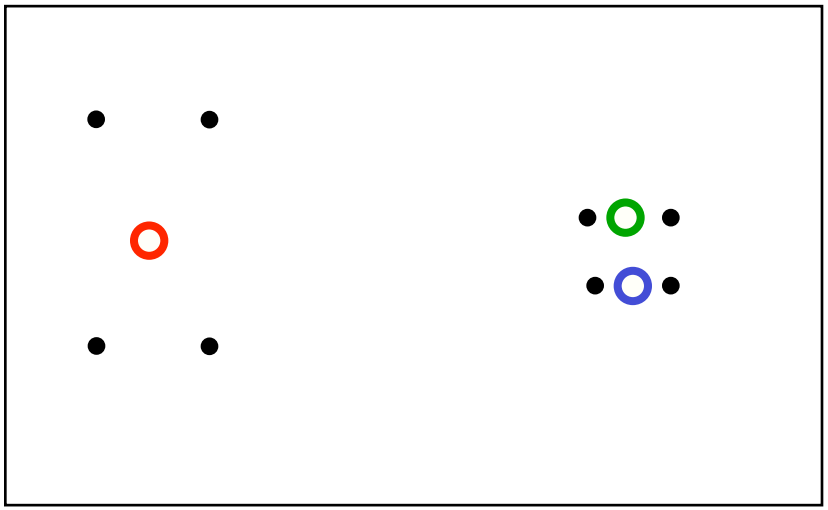


Image Segmentation

- 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.

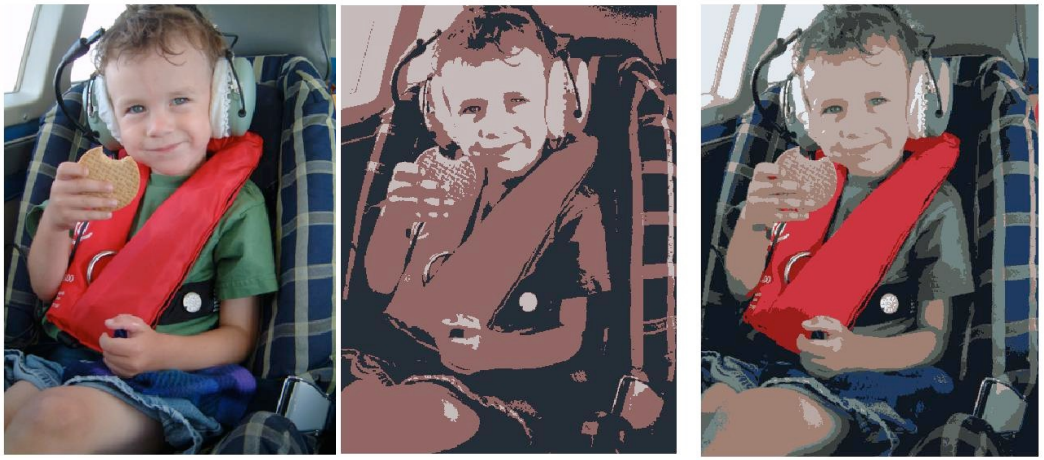
- For a given value of K, the algorithm represent an image using K colors.

- 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 $\boldsymbol{\mu}_k$,
- Provided K << N, we require significantly less data.

Original image K=3 K=10



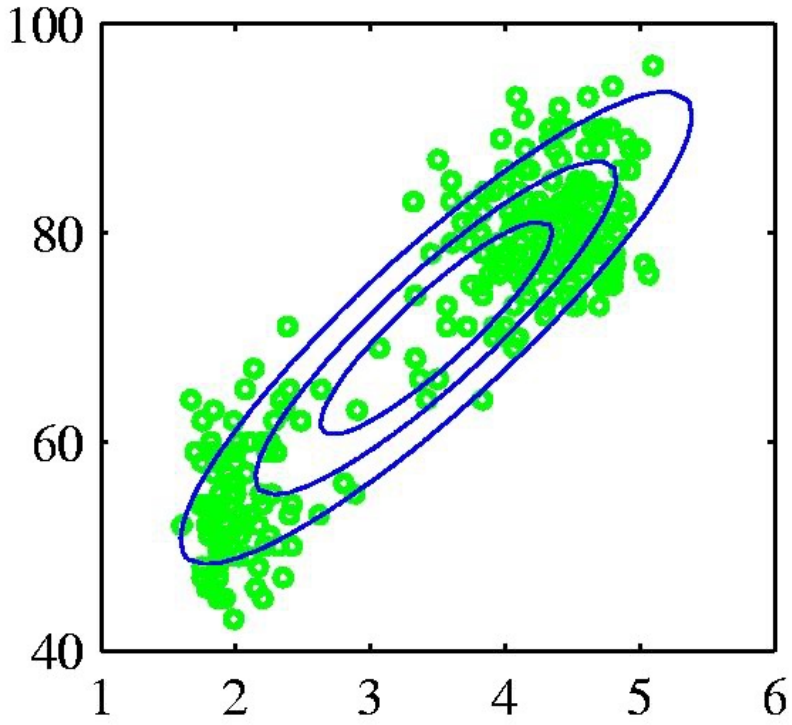
- The original image has 240 x 180 = 43,200 pixels.
- Each pixel contains {R,G,B} values, each of which requires 8 bits.

- Requires 43,200 x 24 = 1,036,800 bits to transmit directly.
- With K-means, we need to transmit K code-book vectors $\boldsymbol{\mu}_k$, -- 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 rations of 4.2%, 8.3%, and 16.7%.

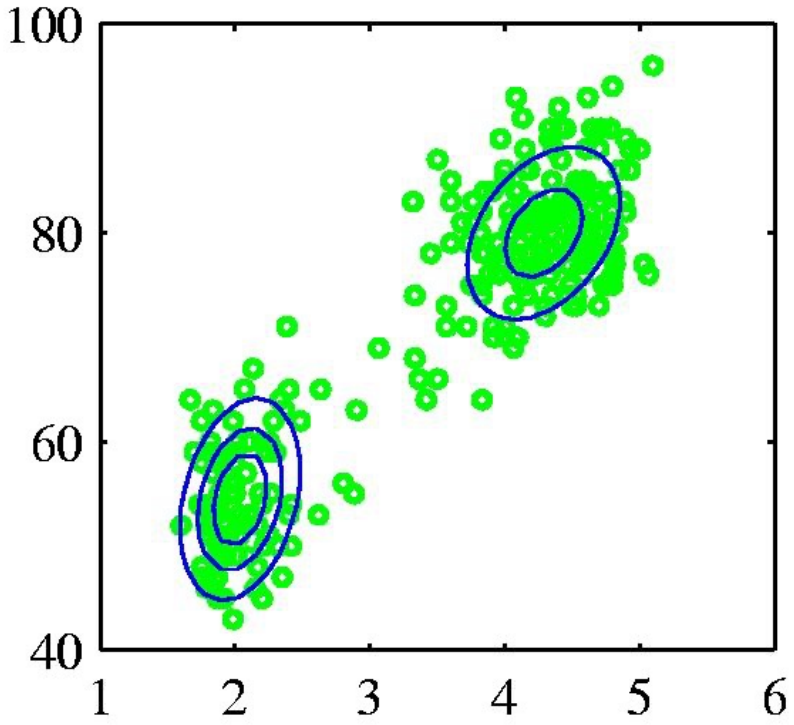
Mixture of Gaussians

- When modeling real-world data, Gaussian assumption may not be appropriate.

- Consider the following example: Old Faithful Dataset



Single Gaussian



Mixture of two Gaussians

Mixture of Gaussians

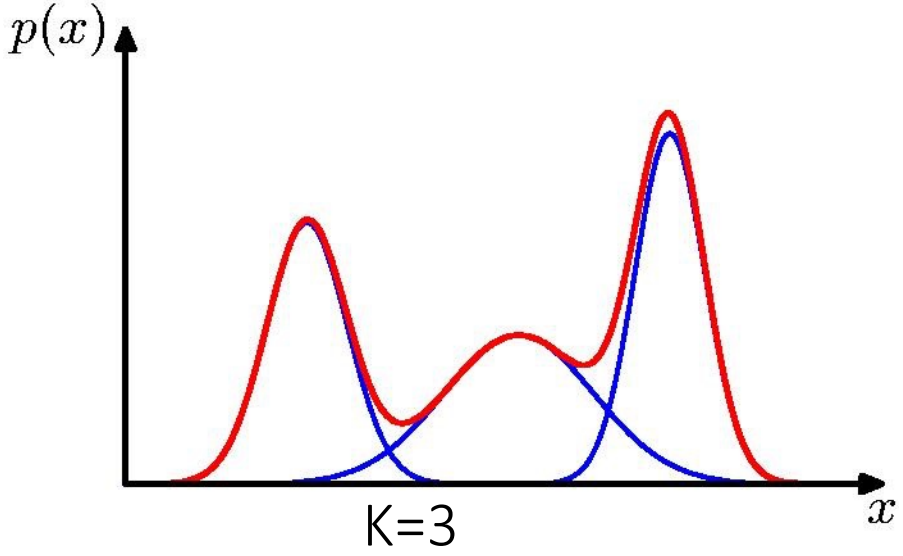
• We can combine simple models into a complex model by defining a superposition of K Gaussian densities of the form:

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

Component

Mixing coefficient

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

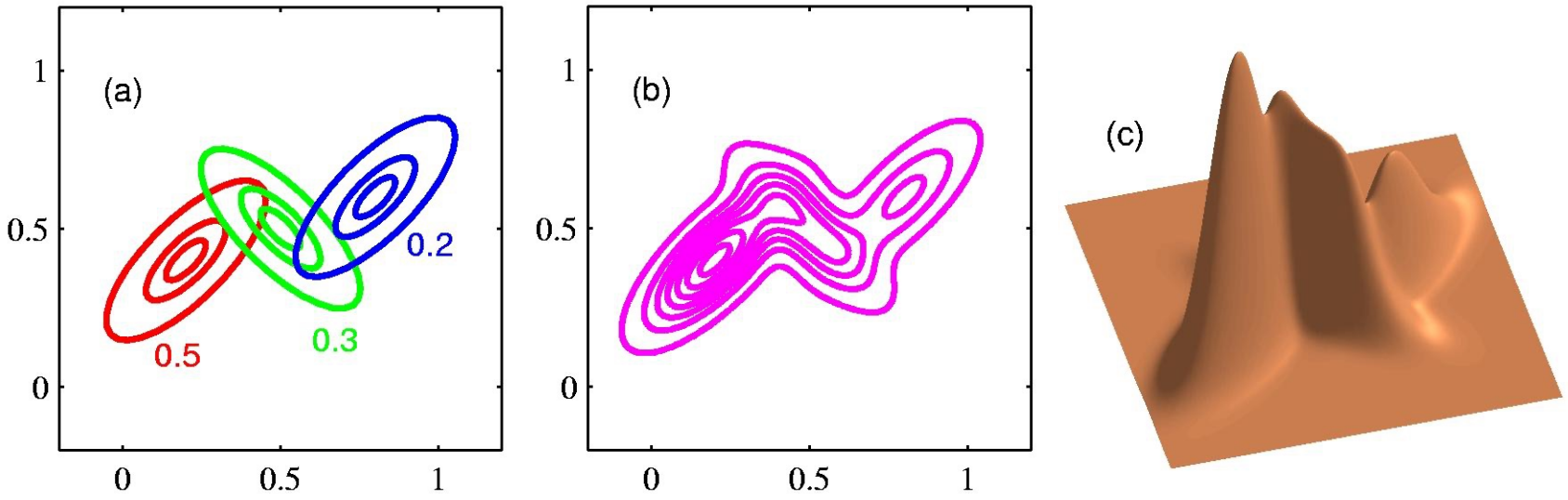


• Note that each Gaussian component has its own mean and covariance. The parameters π_k are called mixing coefficients.

• Mote generally, mixture models can comprise linear combinations of other distributions.

Mixture of Gaussians

- Illustration of a mixture of 3 Gaussians in a 2-dimensional space:



(a) Contours of constant density of each of the mixture components, along with the mixing coefficients

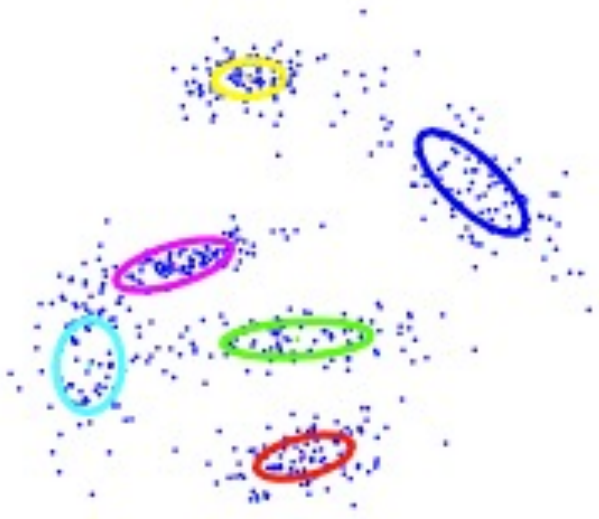
(b) Contours of marginal probability density $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

(c) A surface plot of the distribution p(x).

Mixture of Gaussians

- We will look at mixture of Gaussians in terms of discrete latent variables.
- The Gaussian mixture:

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



- K-dimensional binary random variable **z** having a 1-of-K representation is the latent variable:

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

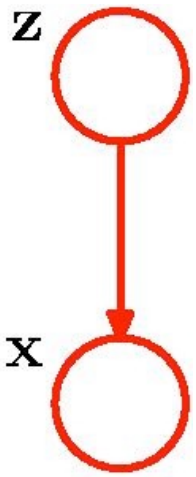
- We will specify the distribution over **z** in terms of mixing coefficients:

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

Mixture of Gaussians

- Because **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 **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 **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 $\mathbf{x}_1,\ldots,\mathbf{x}_N$, 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:

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

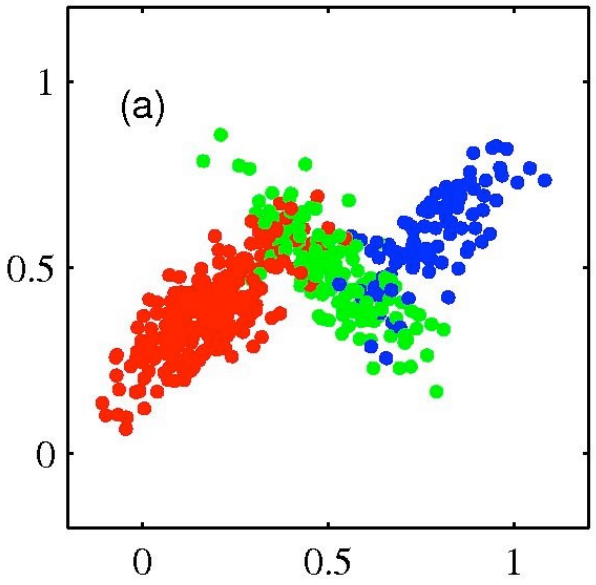
$$= \frac{\pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

responsibility that component k takes for explaining the data x

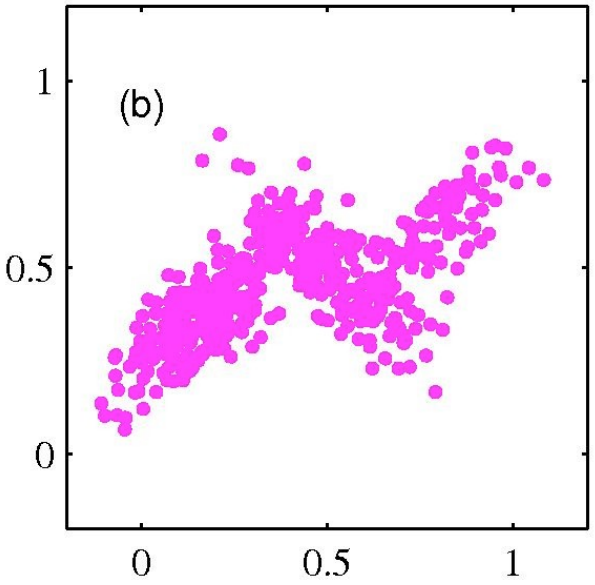
• We will view π_k as prior probability that z_k=1, and $\gamma(z_k)$ is the corresponding posterior once we have observed the data.

Example

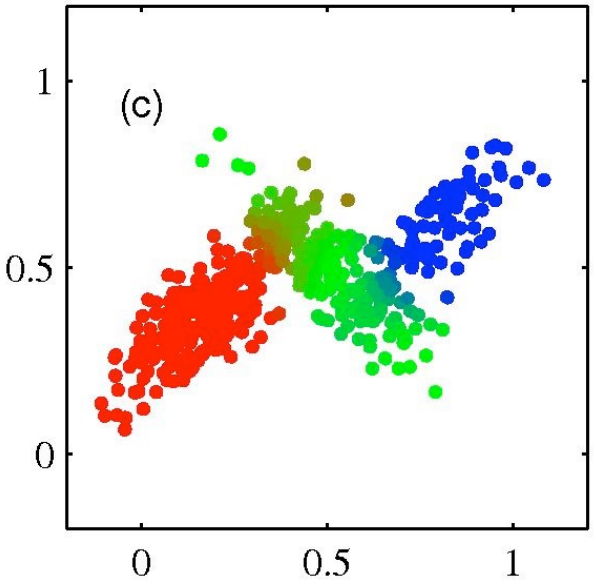
- 500 points drawn from a mixture of 3 Gaussians.



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



Samples from the marginal distribution p(x).



Same samples where colors represent the value of responsibilities.

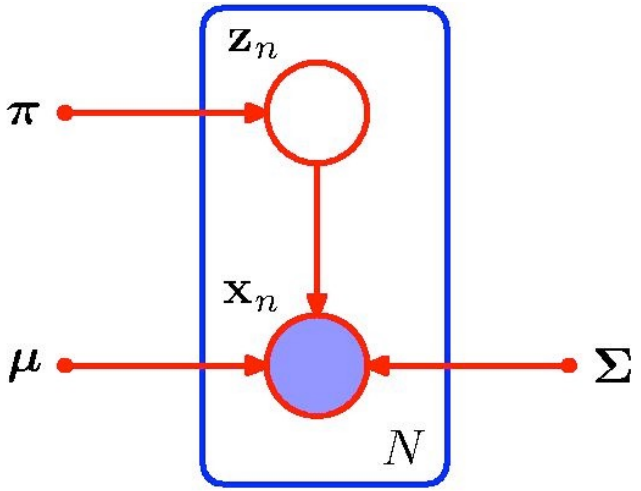
Maximum Likelihood

- Suppose we observe a dataset $\{x_1,\ldots,x_N\}$, and we model the data using mixture of Gaussians.
- We represent the dataset as an N by D matrix **X**.
- The corresponding latent variables will be represented and an N by K matrix **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 $\{x_n\}$, and corresponding latent variables $\{z_n\}$.

Maximum Likelihood

- The log-likelihood:

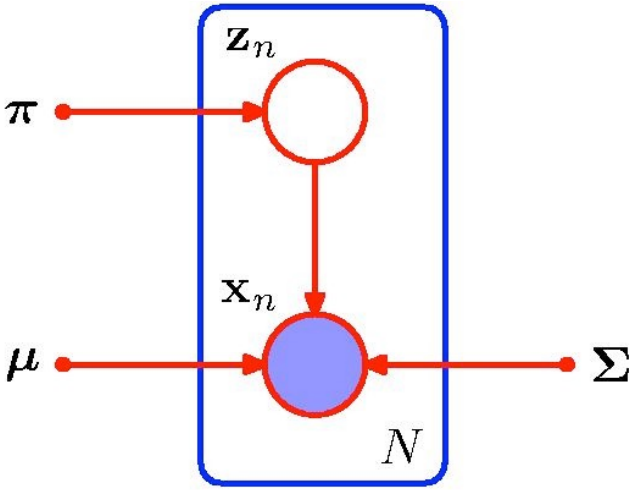
$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

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

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

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 x_n.

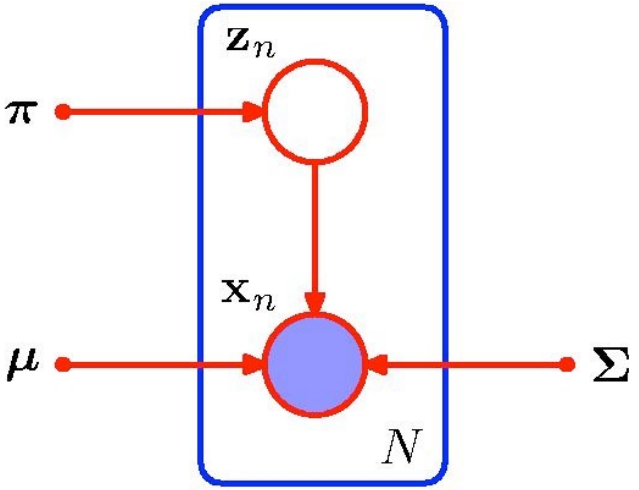
Maximum Likelihood

- 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 Σ_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 kth component is given by the average responsibility which that component takes for explaining the data.

Maximum Likelihood

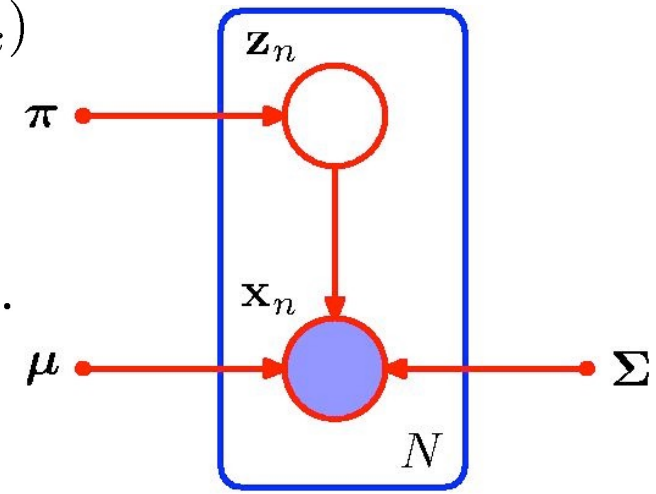
- The log-likelihood:

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

- 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:

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



- Iterative Solution:

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

  M-step: Update model parameters μ_k Σ_k π_k, for k=1,…,K.

EM algorithm

- Initialize the means μ_k, covariances Σ_k, and mixing proportions π_k
- E-step: Evaluate responsibilities using current parameter values:

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

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

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

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

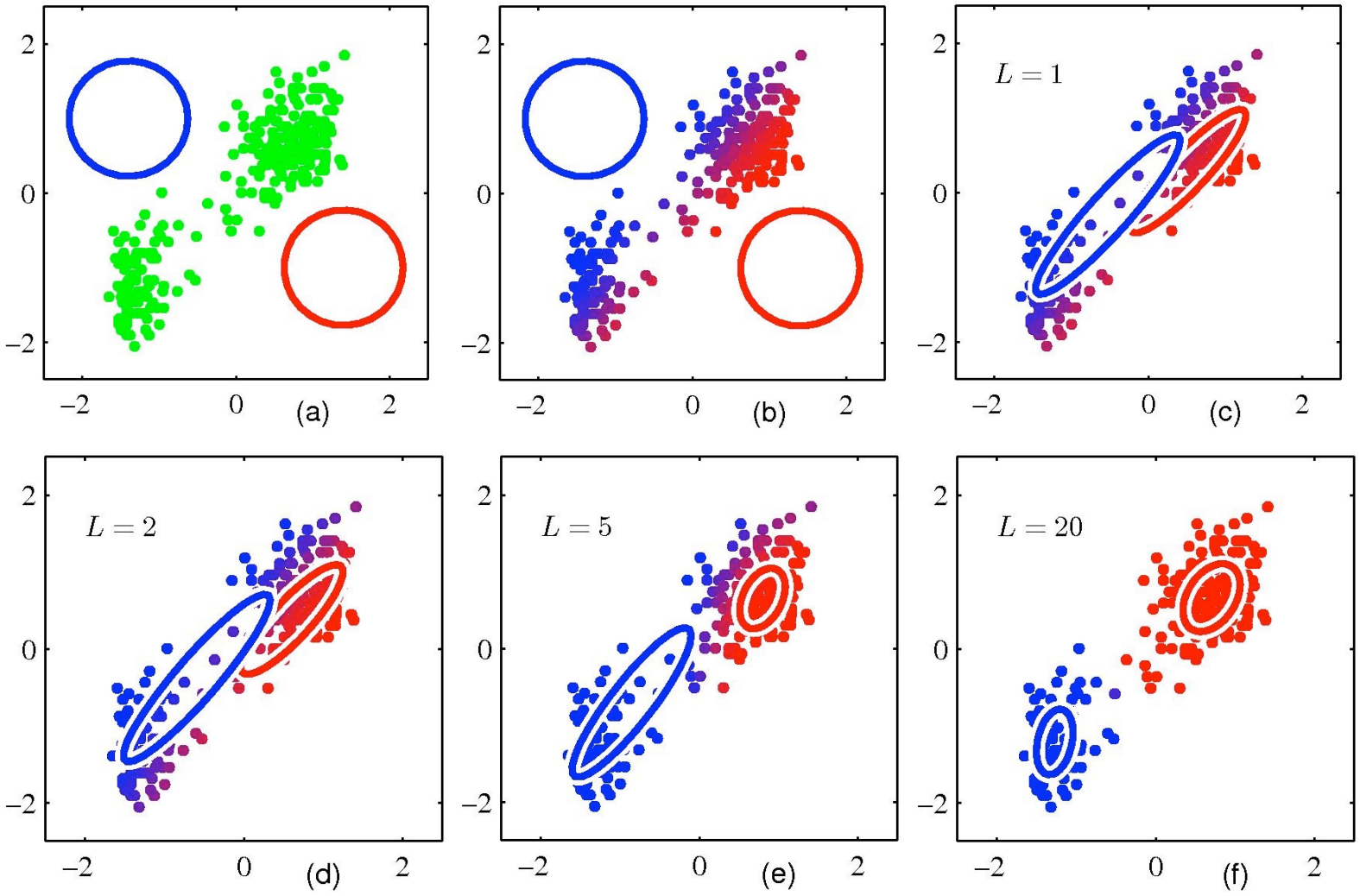
$$\pi_k^{new} = \frac{N_k}{N}.$$

$\boldsymbol{\mu}_k^{new}$

- Evaluate the log-likelihood and check for convergence.

Mixture of Gaussians: Example

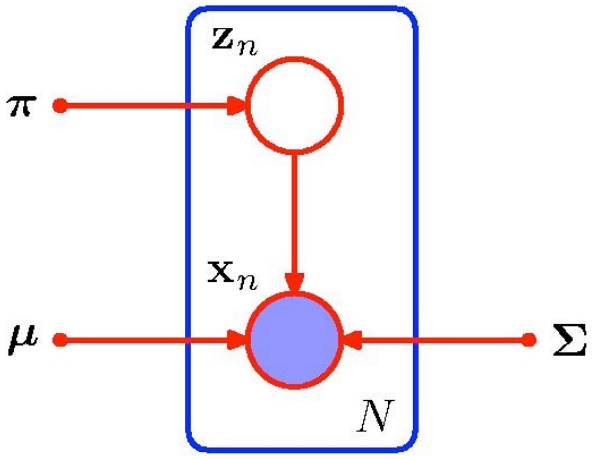
- Illustration of the EM algorithm (much slower convergence compared to K-means)



General EM formulation

- 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 **X**.
- Latent variables will be represented and an N by K matrix **Z**.
- The set of all model parameters is denoted by θ.
- The log-likelihood takes form:

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



- 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.

General EM formulation

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

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

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

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

Tractable

The General EM algorithm

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

where

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

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

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

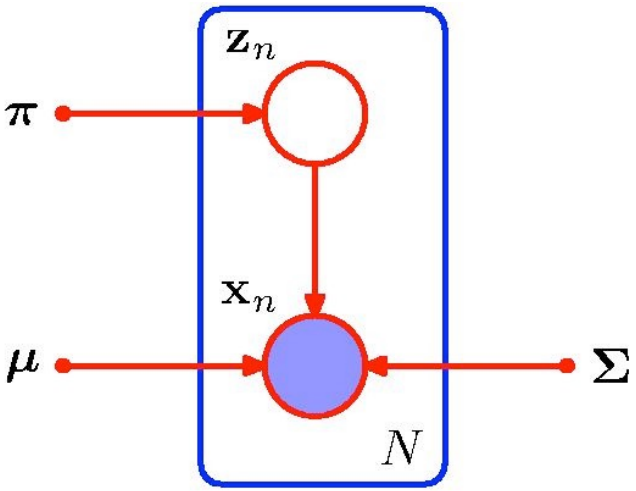
Otherwise:

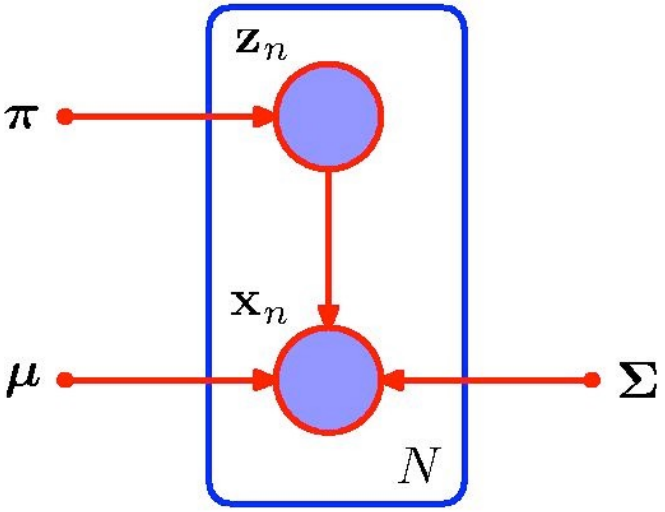
$$\theta^{new} \leftarrow \theta^{old}, \qquad \text{and iterate.}$$

Gaussian Mixtures Revisited

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

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





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

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

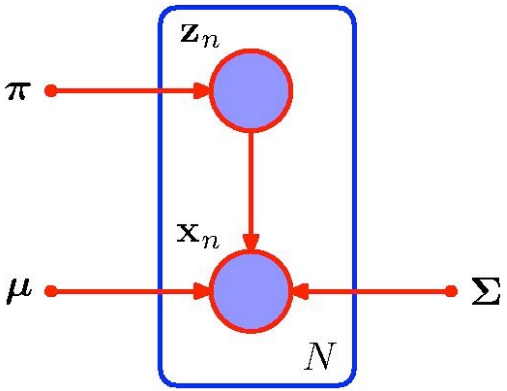
Maximizing Complete Data

- 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.



$\{\mathbf{X}, \mathbf{Z}\}$

-- complete dataset.

- Maximizing with respect to mixing proportions yields:

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

Posterior Over Latent Variables

- Remember:

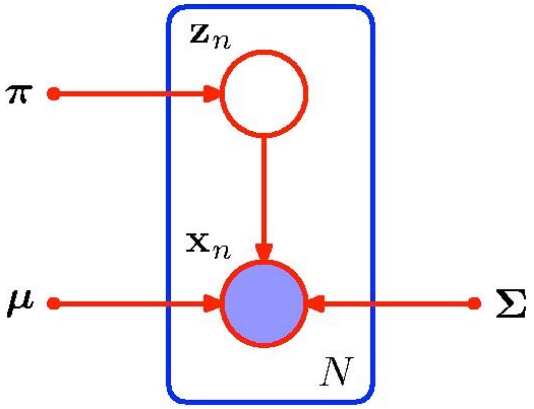
$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}, \quad p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}.$$

- The posterior over latent variables takes form:

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

- Note that the posterior factorizes over n points, so that under the posterior distribution $\{\mathbf{z}_n\}$ are independent.

$$p(z_{nk} = 1|\mathbf{X}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\pi_k \mathcal{N}(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$



Expected Complete Log-Likelihood

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

$$\mathbb{E}[z_{nk}] = \frac{\sum_{z_{nk}} z_{nk}[\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k)]^{z_{nk}}}{\sum_{z_{nj}} [\pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \mathbf{\Sigma}_j)]^{z_{nj}}}$$

Under posterior

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

- This represent the responsibility of component k for data point x_n.
- The complete-data log-likelihood:

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

- The expected complete data log-likelihood is:

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

Expected Complete Log-Likelihood

- The expected complete data log-likelihood is:

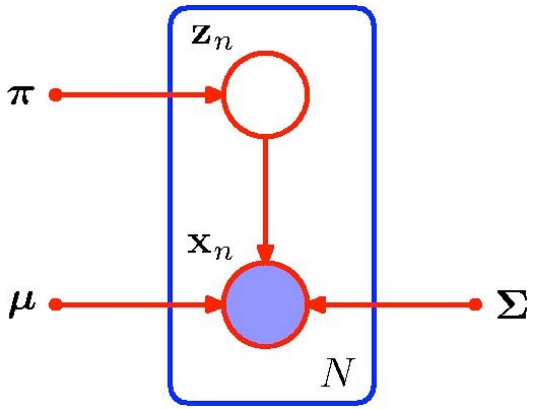
$$\mathbb{E}_{\mathbf{Z}}\left[\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})\right] = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left[\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right].$$

- Maximizing the respect to model parameters we obtain:

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

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

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



Relationship to K-Means

• Consider a Gaussian mixture model in which covariances are shared and are given by ϵI .

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

• Consider EM algorithm for a mixture of K Gaussians, in which we treat $\boldsymbol{\Sigma}_k$ 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 ϵ goes to 0.

• In the denominator, the term for which $||\mathbf{x}_n - \boldsymbol{\mu}_j||^2$ is smallest will go to zero most slowly. Hence $\gamma(z_{nk})$ goes to 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 ϵ 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 ϵ goes to 0, the expected complete log-likelihood becomes:

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

- Hence in the limit, maximizing the expected complete log-likelihood is equivalent to minimizing the distortion measure J for the K-means algorithm.
- This is why assignment step in k-Means is called E-step, and finding the cluster centers is called the M-step.

Continuous Latent Variable Models

- Often there are some unknown underlying causes of the data.
- So far we have looked at models with discrete latent variables, such as mixture of Gaussians.
- Sometimes, it is more appropriate to think in terms of continuous factors which control the data we observe.

- Motivation: for many datasets, data points lie close to a manifold of much lower dimensionality compared to that of the original data space.
- Training continuous latent variable models often called dimensionality reduction, since there are typically many fewer latent dimensions.
- Examples: Principal Components Analysis, Factor Analysis, Independent Components Analysis

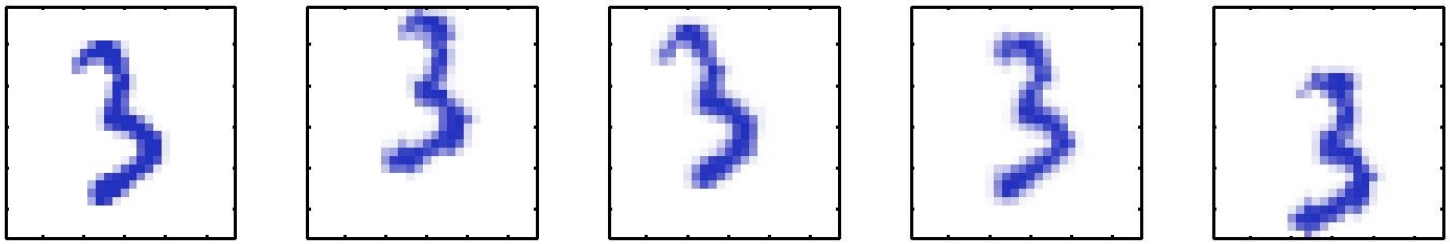
Intrinsic Latent Dimensions

- What are the intrinsic latent dimensions in these two datasets?

- How can we find these latent dimensions from this high-dimensional data.

Intrinsic Latent Dimensions

• In this dataset, there is only 3 degrees of freedom of variability, corresponding to vertical and horizontal translations, and the rotations.



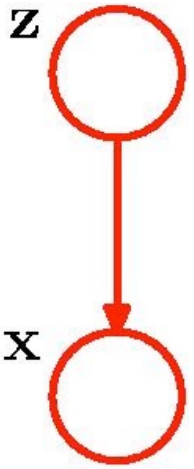
• Each image undergoes a random displacement and rotation within some larger image field.

• The resulting images have 100 x 100 = 10,000 pixels.

Generative View

• Each data example generated by first selecting a point from a distribution in the latent space, then generating a point from the conditional distribution in the input space

• Simplest latent variable models: Assume Gaussian distribution for both latent and observed variables.



• This leads to probabilistic formulation of the Principal Component Analysis.

• We will first look at standard PCA, and then consider its probabilistic formation.

• Advantages of probabilistic formulation: use of EM for parameter estimation, mixture of PCAs, Bayesian PCA.