NUS National University of Singapore | School of Computing

CS5340
Uncertainty Modeling in AI

Lecture 7:
Mixture Models and the EM Algorithm

Assoc. Prof. Lee Gim Hee

AY 2022/23

Semester 1

Course Schedule

Week	Date	Topic	Remarks
1	10 Aug	Introduction to probabilistic reasoning	**Assignment 0:** Python Numpy Tutorial (Ungraded)
2	17 Aug	Bayesian networks (Directed graphical models)	
3	24 Aug	Markov random Fields (Undirected graphical models)	
4	31 Aug	Variable elimination and belief propagation	**Assignment 1:** Belief propagation and maximal probability (15%)
5	07 Sep	Factor graph and the junction tree algorithm	
6	14 Sep	Parameter learning with complete data	**Assignment 1:** Due **Assignment 2:** Junction tree and parameter learning (15%)
-	21 Sep	Recess week	**No lecture**
7	28 Sep	Mixture models and the EM algorithm	**Assignment 2:** Due
8	05 Oct	Hidden Markov Models (HMM)	**Assignment 3:** Hidden Markov model (15%)
9	12 Oct	Monte Carlo inference (Sampling)	
*	15 Oct	Variational inference	Makeup Lecture (LT15) Time: 9.30am – 12.30pm (Saturday)
10	19 Oct	Variational Auto-Encoder and Mixture Density Networks	**Assignment 3:** Due **Assignment 4:** MCMC Sampling (15%)
11	26 Oct	No Lecture	I will be traveling
12	02 Nov	Graph-cut and alpha expansion	**Assignment 4:** Due
13	09 Nov	-	

Acknowledgements

- A lot of slides and content of this lecture are adopted from:

1. "Pattern Recognition and Machine Learning", Christopher Bishop, Chapter 8.
2. "Machine Learning – A Probabilistic Perspective", Kevin Murphy, Chapter 11.
3. "An Introduction to Probabilistic Graphical Models", Michael I. Jordan, Chapters 10 and 11.
   http://people.eecs.berkeley.edu/~jordan/prelims/chapter10.pdf
   http://people.eecs.berkeley.edu/~jordan/prelims/chapter11.pdf
4. "Probabilistic Graphical Models", Daphne Koller and Nir Friedman, chapter 19.2.2.
5. "Computer Vision: Models, Learning and Inference", Simon Prince, Chapters 7.1-7.4 and 7.8.

Learning Outcomes

- Students should be able to:

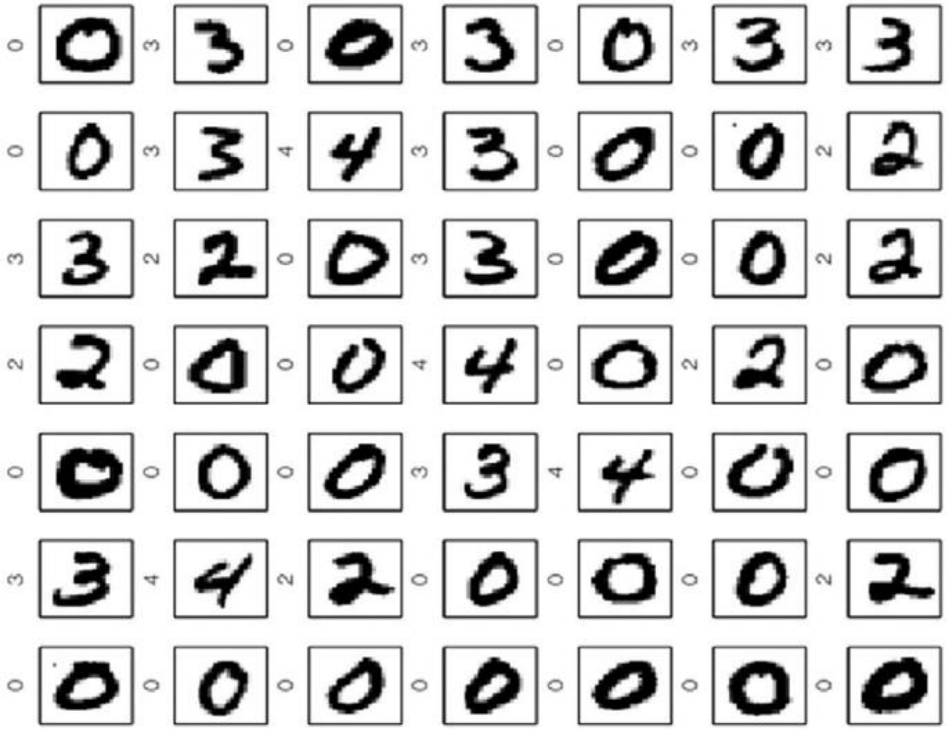
1. Use the non-probabilistic k-mean algorithm to solve the clustering problem.

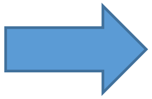
2. Describe the Gaussian-mixture model.

3. Apply the Expectation-Maximization algorithm for estimation of both the unknown parameters and latent variables.

Motivation

- Suppose that we are given many images of handwritten digits, and we can extract a 2D representation of each image $\mathrm{x}_n \in \mathbb{R}^2$.





$\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \ldots, \mathrm{x}_N$

MNIST Dataset: http://yann.lecun.com/exdb/mnist/

Motivation

- We can observe clusters from the 2D plot of $\mathrm{x}_1, \ldots, \mathrm{x}_N$, where $\mathrm{x}_n \in \mathbb{R}^2$.

$\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \ldots, \mathrm{x}_N$

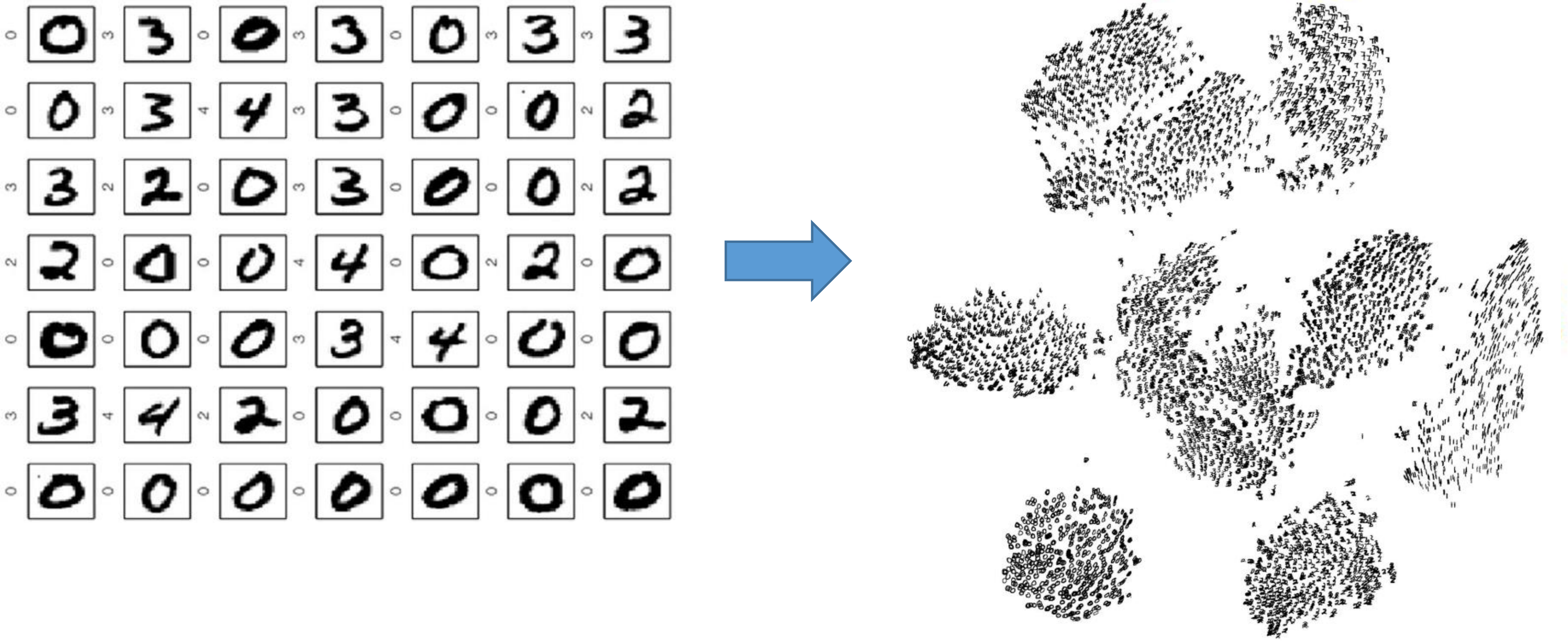


Image source: https://stab-iitb.org/electronics-club/blog/2016/05/deep-learning-based-image-classification/

Motivation

- We can observe clusters from the 2D plot of $\mathrm{x}_1, \dots, \mathrm{x}_N$, where $\mathrm{x}_n \in \mathbb{R}^2$.

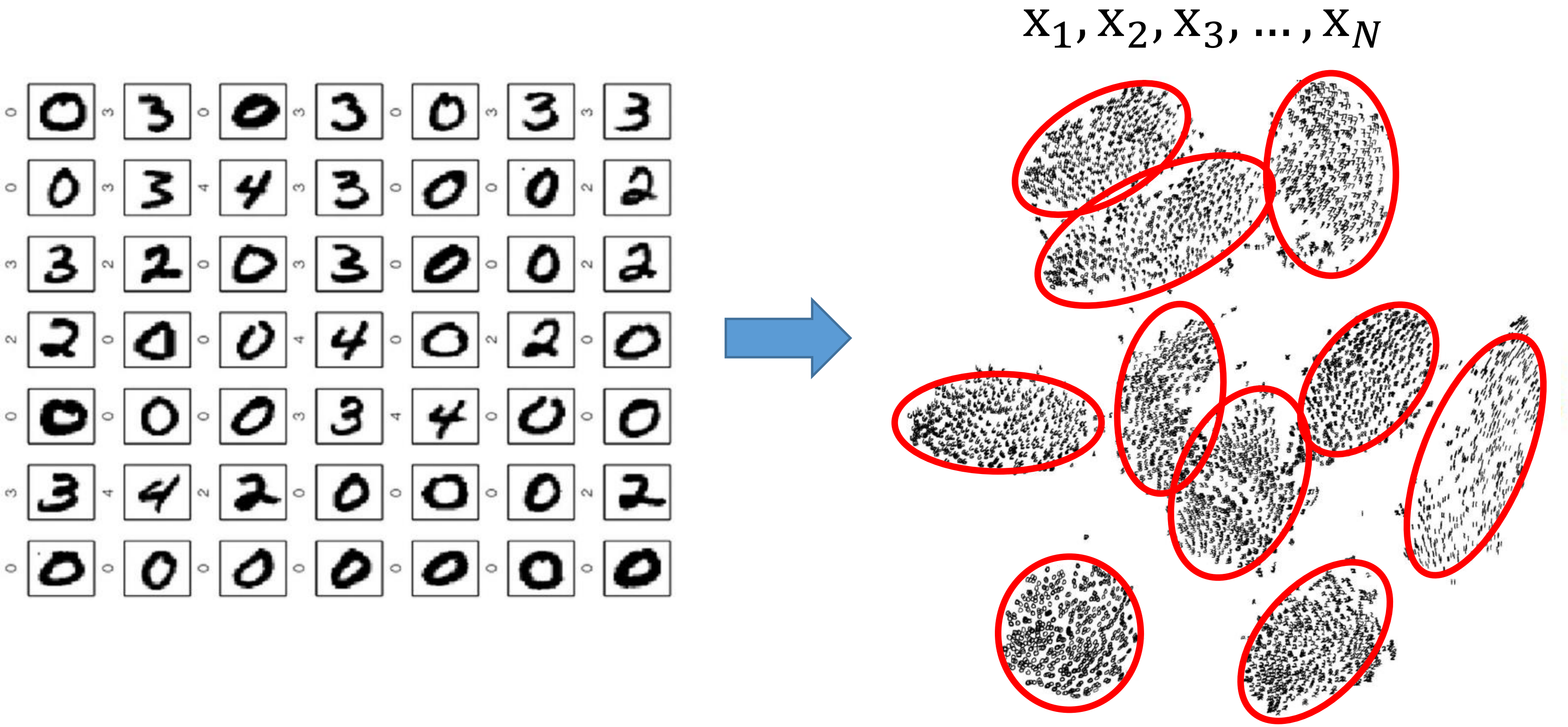
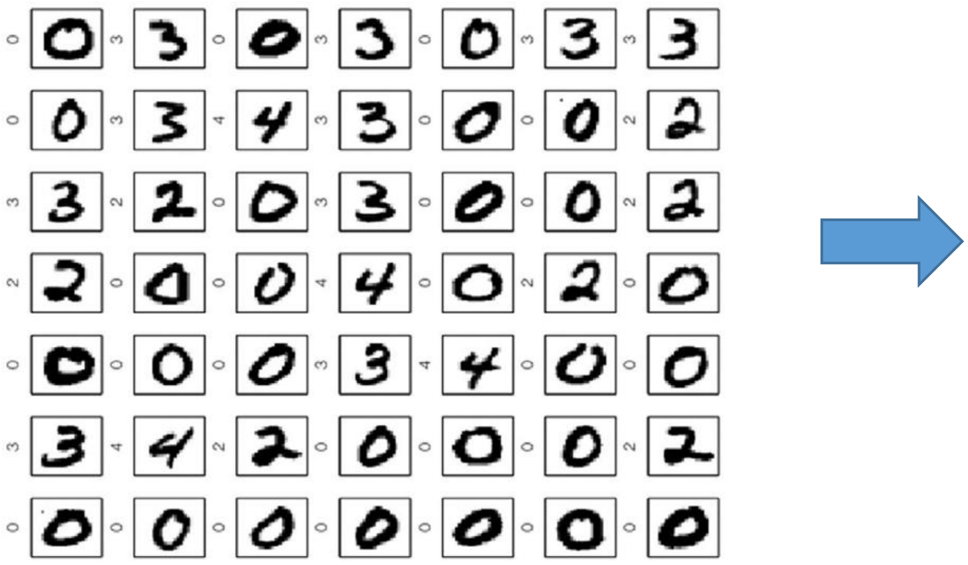


Image source: https://stab-iitb.org/electronics-club/blog/2016/05/deep-learning-based-image-classification/

Motivation

- It turns out that the data points x_n with the same digit tend to be associated with the same cluster!
- This suggests that clusters can be used to represent complex distributions.



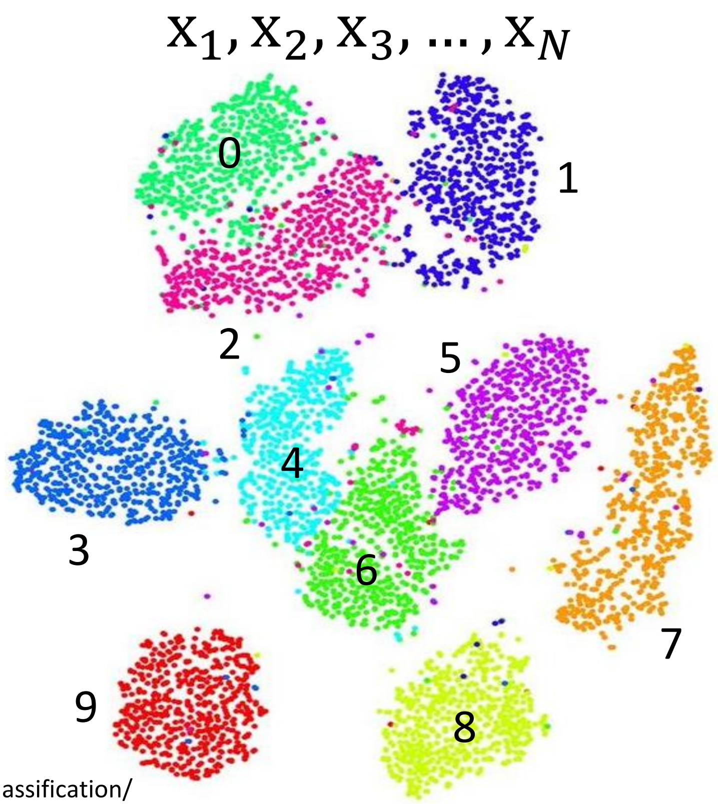

$\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \dots, \mathrm{x}_N$
0
1
2
5
3
4
6
7
9
8

Image source: https://stab-iitb.org/electronics-club/blog/2016/05/deep-learning-based-image-classification/

Motivation

Chicken-and-Egg Problem:

1. How to find unknown parameters of the clusters?
2. Which cluster does each data belong to, i.e. data association?

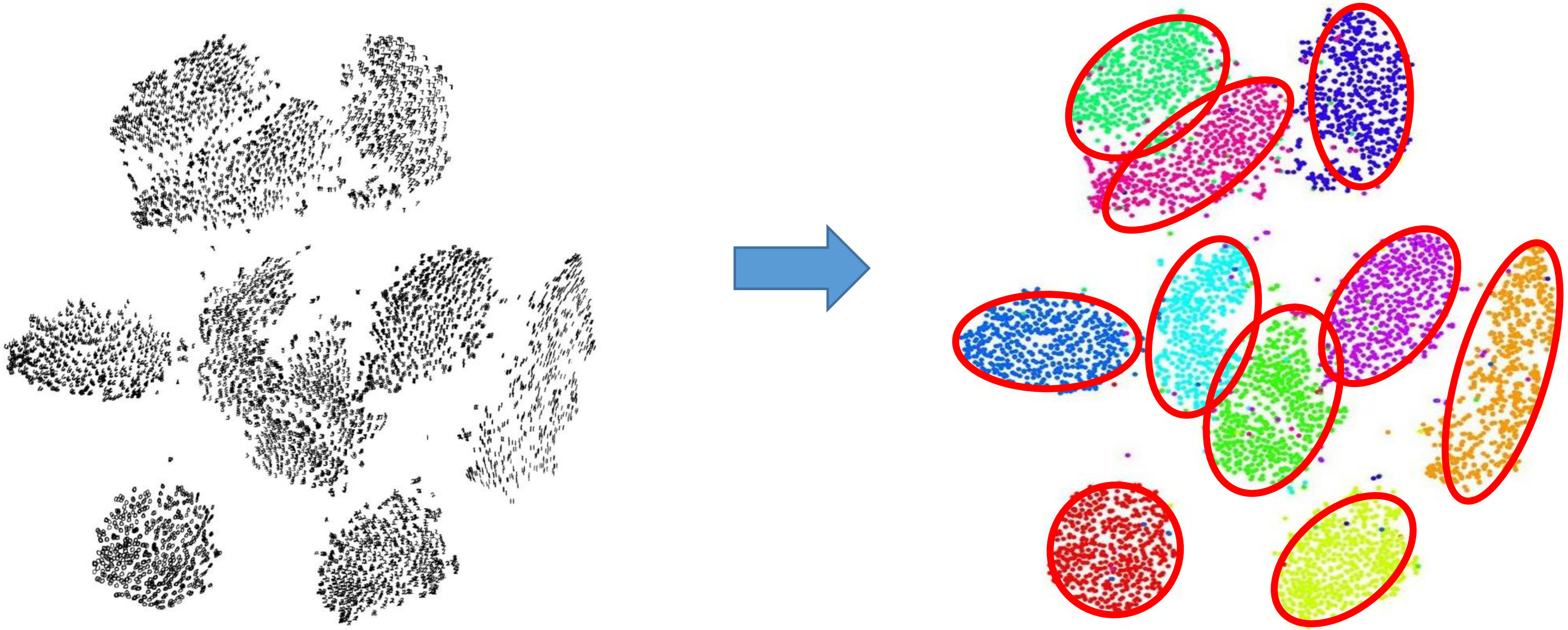


Image source: https://stab-iitb.org/electronics-club/blog/2016/05/deep-learning-based-image-classification/

Non-Probabilistic Approach: K-Means

Given:

1. Data set $\{\mathrm{x}_1, \ldots \mathrm{x}_N\}$ of observations
2. Number of clusters K

Find:

1. The K cluster centers $\{\mu_1, \ldots, \mu_K\}$, i.e. unknown parameters (assume each cluster is a circle)
2. Assignment of each point x_n to a cluster center k, i.e. data association

Non-Probabilistic Approach: K-Means

1-of-K coding:

- For each data point x_n, we introduce a corresponding set of binary indicator variables:

$$r_{nk} \in \{0,1\} \;\; \forall k = 1, \ldots K, \qquad \text{s.t.} \quad \underbrace{\textstyle\sum_k r_{nk} = 1}_{\text{1-of-}K\text{ constraint}}.$$

- This binary indicator variable describes which of the K clusters the data point x_n is assigned to.

- 1-of-K constraint ensures that each data point x_n gets assigned to only ONE cluster k.

Non-Probabilistic Approach: K-Means

- Formally, the **goal** of k-means is to find values for $\{r_{nk}\}$ and $\{\mu_k\}$ to minimize the "distortion measure":

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2, \qquad \text{s.t.} \quad \sum_k r_{nk} = 1.$$

- Represents the sum-of-squares of the distances of each point x_n to its assigned vector μ_k.

Non-Probabilistic Approach: K-Means

K-Means (a.k.a. Lloyd) Algorithm:

1. Initialization: Randomly choose some initial values for $\{\mu_k\}$.
2. Assignment step: Minimize J w.r.t. $\{r_{nk}\}$, while keeping $\{\mu_k\}$ fixed.
3. Update step: Minimize J w.r.t. $\{\mu_k\}$, while keeping $\{r_{nk}\}$ fixed.

Iterate until convergence

Non-Probabilistic Approach: K-Means

Assignment Step:

- We can optimize each r_n separately since they are independent:

$$\underset{r_n}{\operatorname{argmin}} \sum_k r_{nk} \|\mathrm{x}_n - \mu_k\|^2 , \qquad \text{s.t.} \quad \sum_k r_{nk} = 1.$$

- By inspection, the minimum occurs when we assign x_n to the current closest cluster center:

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

Non-Probabilistic Approach: K-Means

Update Step:

- We can optimize each μ_k separately since they are independent:

$$\underset{\mu_k}{\operatorname{argmin}} \underbrace{\sum\nolimits_n r_{nk} \| \mathrm{x}_n - \mu_k \|^2}_{L}$$

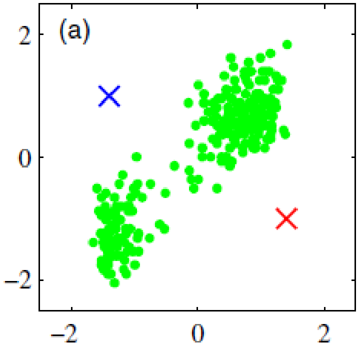
- Differentiation of L w.r.t. μ_k, and equate to zero gives:

$$2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \Rightarrow \boxed{\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}.}$$

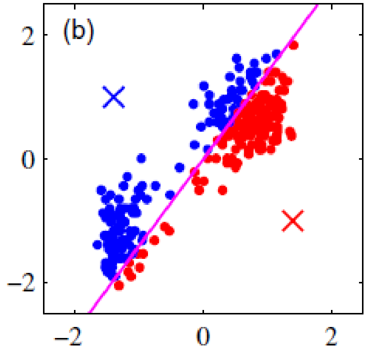
Mean of all points assigned to cluster k, hence "k-means"!

Non-Probabilistic Approach: K-Means

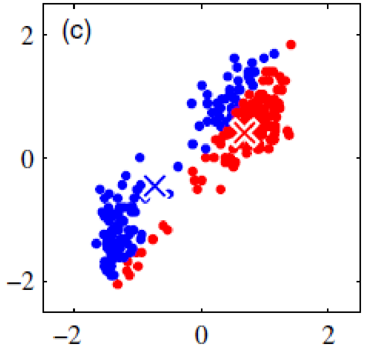
Initialization ($K = 2$):
Choose μ_k



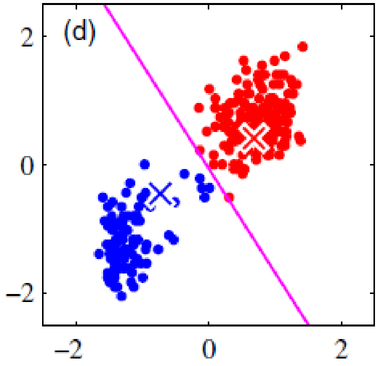
Iteration (1):
Assignment of r_{nk}



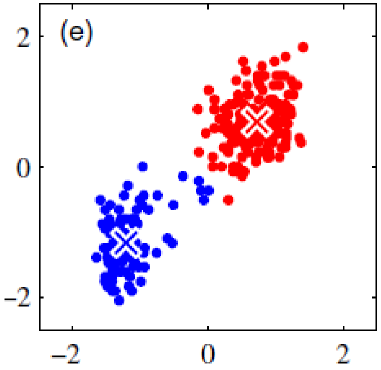
Iteration (1):
Update of μ_k



Iteration (2):
Assignment of r_{nk}



Iteration (2):
Update of μ_k



Iteration (3):
Assignment of r_{nk}

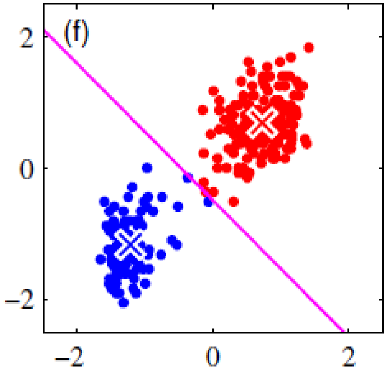
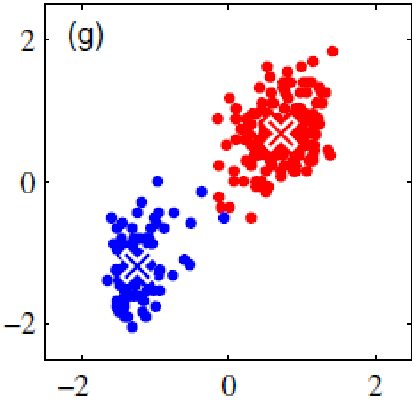


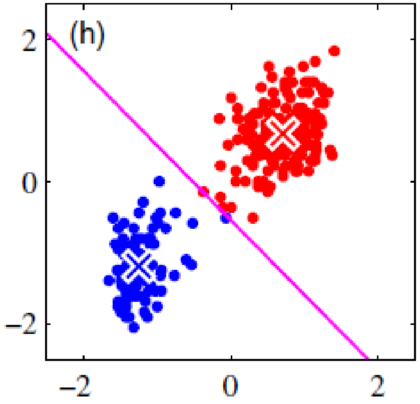
Image source: “Pattern recognition and machine learning”, Christopher Bishop

Non-Probabilistic Approach: K-Means

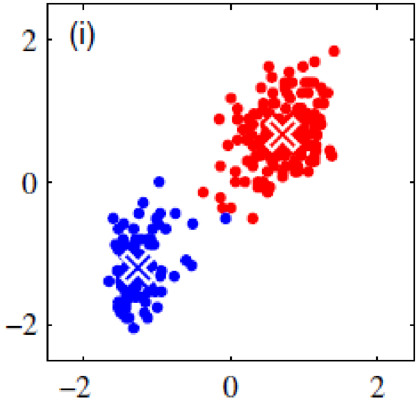
Iteration (3):
Update of μ_k

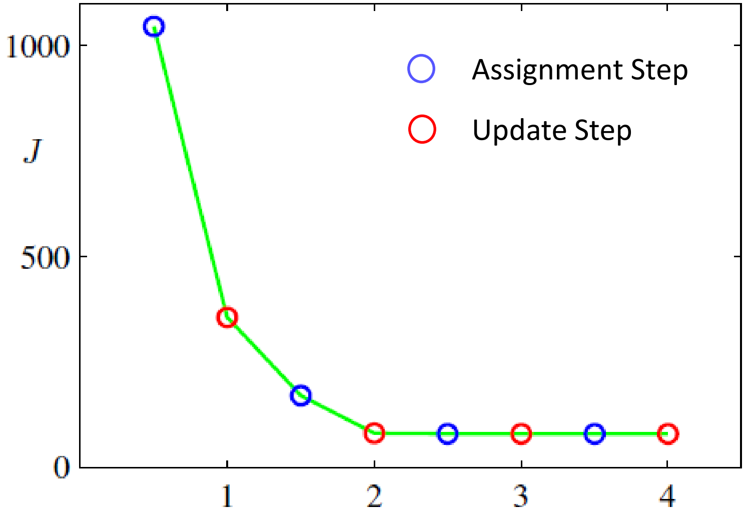


Iteration (4):
Assignment of r_{nk}



Iteration (4):
Update of μ_k





- Plot of the cost function *J* after each iteration.
- The algorithm has converged after the third update step.

Image source: “Pattern recognition and machine learning”, Christopher Bishop

Application: Image Segmentation

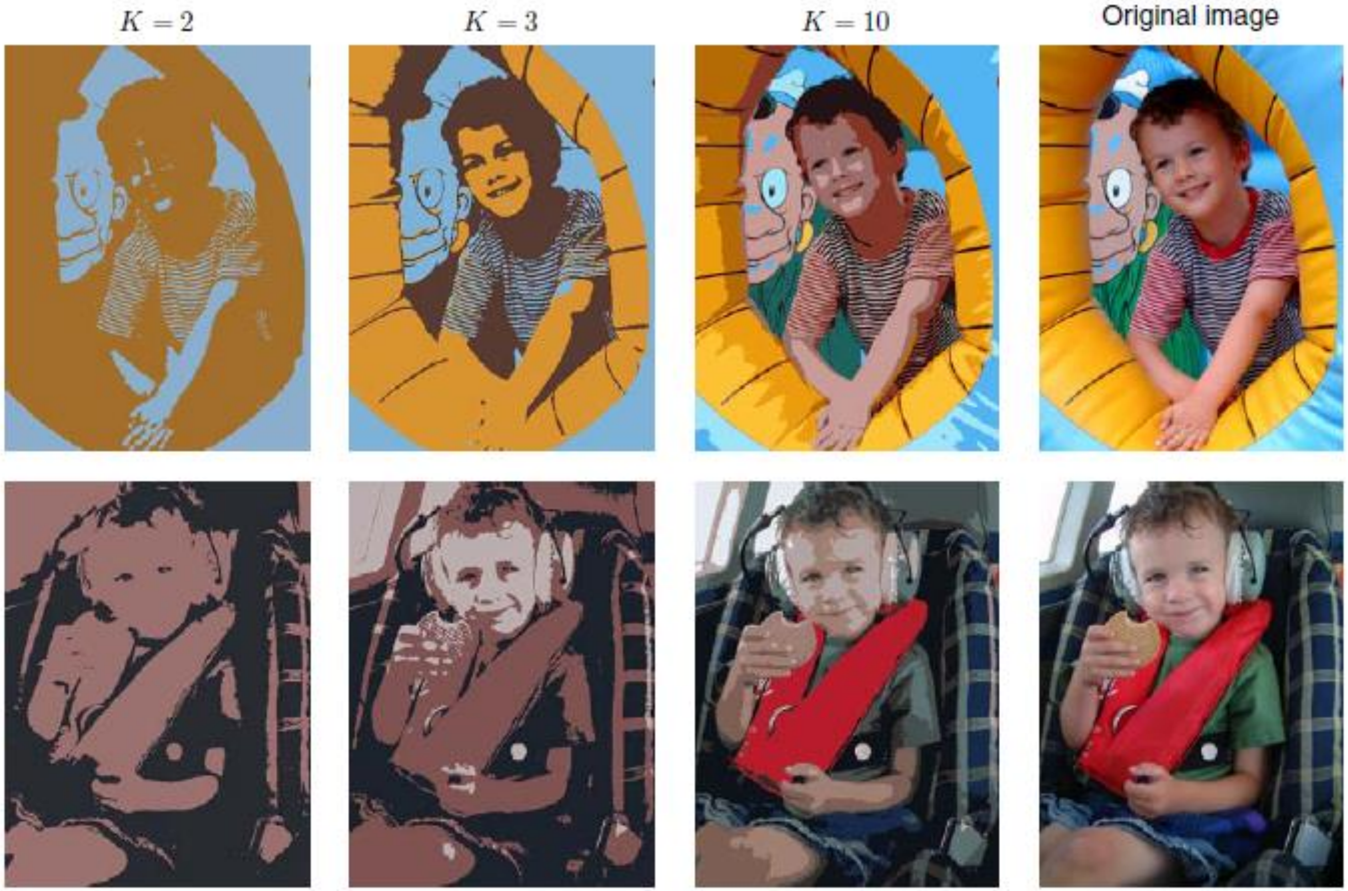


Image source: "Pattern recognition and machine learning", Christopher Bishop

Probabilistic Approach

Can we model the problem with a probabilistic approach?

Gaussian Mixture Models

- "Old Faithful" dataset: 272 measurements of the eruption of the Old Faithful geyser at Yellowstone National Park, USA.
- Data set forms two dominant clumps, a simple Gaussian distribution is unable to capture this structure.
- A linear superposition of two Gaussians gives a better characterization of the data set.

Photo source: "Old Faithful", courtesy of Chen Li, June'18

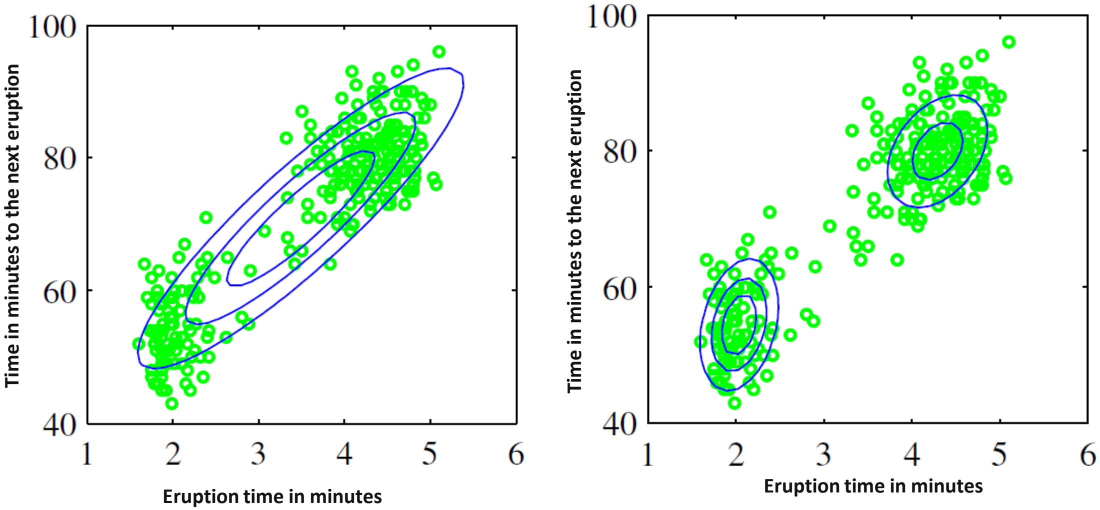


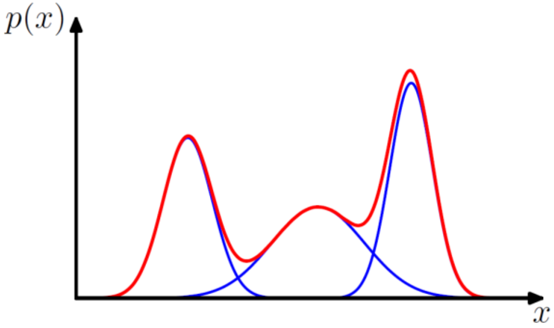
Image source: "Pattern recognition and machine learning", Christopher Bishop

Gaussian Mixture Models

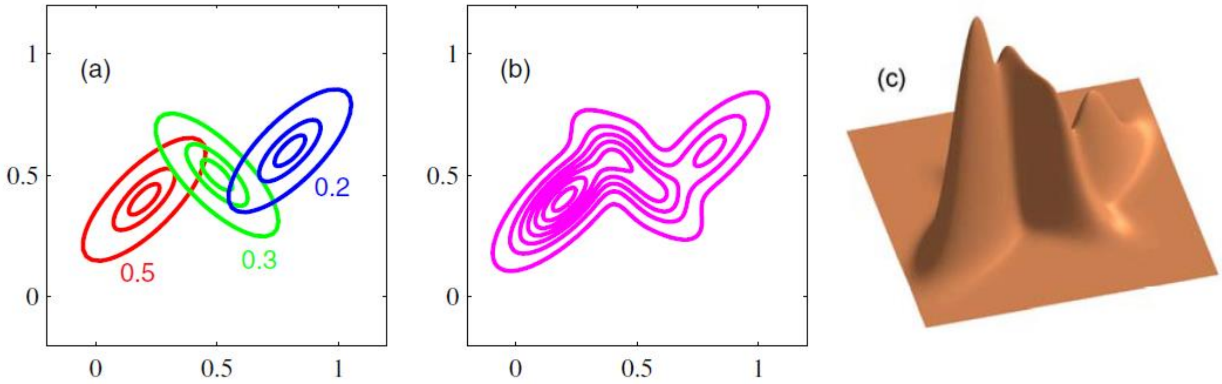
- Mixture distributions: Probabilistic models formed by taking linear combinations of more basic distributions such as Gaussian a.k.a Gaussian Mixture Model (GMM).

- Linear combination of sufficient number of Gaussians give rise to very complex densities that can be used to approximate almost any continuous density with arbitrary accuracy.

Example:



1D GMM (red) formed by 3 Gaussians (blue)



(b)-(c) 2D GMM formed by (a) 3 Gaussians

Image source: “Pattern recognition and machine learning”, Christopher Bishop

Gaussian Mixture Models

- The probability distribution of a mixture of Gaussians is given by the superposition of K Gaussian densities:

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

- Each Gaussian density $\mathcal{N}(\mathrm{x}\,|\mu_k, \Sigma_k)$ is called a component of the mixture, and has its own mean μ_k and covariance Σ_k.

- The parameters $0 \leq \pi_k \leq 1$ is the mixing coefficients, and must sum to one:

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

Gaussian Mixture Models

- Let us introduce a K-dimensional binary random variable Z having a 1-of-K representation.

- $z_k = 1 \ \Rightarrow z_{j \neq k} = 0$ indicates the assignment of the random variable x to the k^{th} Gaussian density.

- The values of Z_{k} must satisfy:

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

- K possible states for the vector Z according to which element is non-zero.

Gaussian Mixture Models

- The marginal distribution of Z is a categorical distribution specified in terms of the mixing coefficients π_k:

$$p(\mathrm{z}) = \prod_{k=1}^{K} \pi_k^{z_k} = \mathrm{cat_z}[\pi] \ ,$$

where the parameter $\pi = [\pi_1, \ldots, \pi_K]$ must be:

$$0 \leq \pi_k \leq 1 \qquad \text{and} \qquad \sum_{k=1}^{K} \pi_k = 1 \ .$$

Gaussian Mixture Models

- Conditional distribution of X given a particular value for Z is a Gaussian:

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

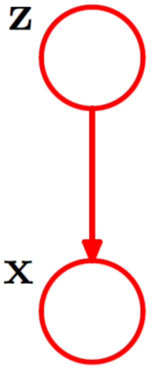
- Which can also be written as:

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

Gaussian Mixture Models

- The joint distribution $p(\mathrm{x}, \mathrm{z})$ is given by the following DGM:

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



- The marginal distribution of X is then obtained by summing the joint distribution over all possible states of the latent variable Z to give:

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

Image source: “Pattern recognition and machine learning”, Christopher Bishop

Gaussian Mixture Models

$$p(\mathrm{x}) = \sum_{\mathrm{z}} p(\mathrm{z}) p(\mathrm{x}|\mathrm{z})$$

$$= \sum_{\mathrm{z}} \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathrm{x} \mid \mu_k, \Sigma_k)^{z_k}$$

$$= \underbrace{\left(\prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathrm{x} \mid \mu_k, \Sigma_k)^{z_k}\right)_{z_{k=1}=1}}_{\pi_{k=1}\mathcal{N}(\mathrm{x} \mid \mu_{k=1}, \Sigma_{k=1})} + \quad \ldots \quad \ldots \quad + \underbrace{\left(\prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathrm{x} \mid \mu_k, \Sigma_k)^{z_k}\right)_{z_{k=K}=1}}_{\pi_{k=K}\mathcal{N}(\mathrm{x} \mid \mu_{k=K}, \Sigma_{k=K})}$$

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

Gaussian Mixture Models

- Another quantity that will play an important role is $p(z_k = 1|\,\mathrm{x})$ denoted as $\gamma(z_k)$, whose value can be found using Bayes' theorem:

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

Prior probability of $z_k = 1$

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

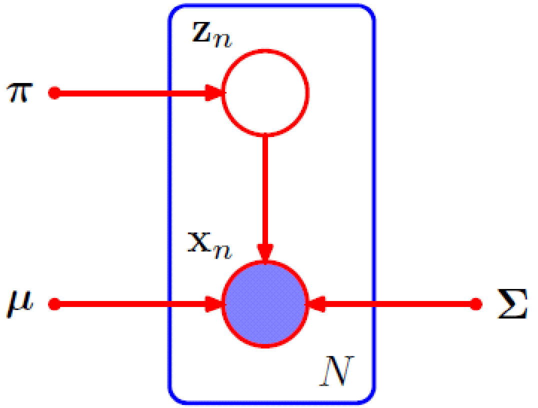
- As we shall see later, $\gamma(z_k)$ can also be viewed as the *responsibility* that component k takes for 'explaining' the observation x.

Maximum Log-Likelihood

- Suppose we have a data set of N i.i.d. observations $\{\mathrm{x}_1, \ldots, \mathrm{x}_N\}: \mathrm{x}_n \in \mathbb{R}^D$, we model the log-likelihood as:

Unknown parameter θ

$$\ln p(\mathrm{x}_1, \ldots \mathrm{x}_N | \, \theta) = \textstyle\sum_{n=1}^{N} \ln \underbrace{\textstyle\sum_{\mathrm{z}_n} p(\mathrm{x}_n, \mathrm{z}_n | \theta)}_{\text{Incomplete data because } \mathrm{Z} = \{\mathrm{z}_1, \ldots, \mathrm{z}_n\} \in \mathbb{R}^{N \times K} \text{ is a latent variable}}$$



$$= \textstyle\sum_{n=1}^{N} \ln \sum_{\mathrm{z}_n} p(\mathrm{z}_n | \pi) p(\mathrm{x}_n | \mathrm{z}_n, \mu, \Sigma)$$

$$= \textstyle\sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathrm{x}_n \, | \mu_k, \Sigma_k),$$

where

$$\theta = \{\pi_1 \ldots \pi_K, \mu_1 \ldots \mu_K, \Sigma_1 \ldots \Sigma_K\}.$$

Image source: “Pattern recognition and machine learning”, Christopher Bishop

Maximum Log-Likelihood

$$\underset{\theta}{\operatorname{argmax}} \ln p(\mathrm{x}_1, \dots \mathrm{x}_N \mid \theta) = \underset{\pi,\mu,\Sigma}{\operatorname{argmax}} \sum_{n=1}^{N} \boxed{\ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathrm{x}_n \mid \mu_k, \Sigma_k)}$$

- The summation term inside the logarithm prevents the logarithm function from acting directly on the Gaussian.

- We shall see that we will no longer obtain a closed-form solution of the unknown parameters by setting the derivatives to zero!

Maximum Log-Likelihood

- Setting the derivatives of $\ln p(\mathrm{x}_1, \ldots \mathrm{x}_N | \, \theta)$ w.r.t. μ_k of the Gaussian components to zero, we obtain:

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

- Multiplying by Σ_k^{-1} (which we assume to be non-singular) and rearranging, we obtain:

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

where

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

Not closed-form since responsibility is a function of π_k, μ_k, Σ_k!

Maximum Log-Likelihood

- If we set the derivative of $\ln p(\mathrm{x}_1, \ldots \mathrm{x}_N | \, \theta)$ w.r.t. Σ_k to zero, we get:

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

Not closed-form since responsibility is a function of π_k, μ_k, Σ_k!

Maximum Log-Likelihood

- Finally, we maximize $\ln p(\mathrm{x}_1, \ldots \mathrm{x}_N \mid \theta)$ w.r.t. π_k subjected to $\sum_k \pi_k = 1$ by maximizing the following auxiliary equation:

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \overset{\text{Lagrange multiplier}}{\lambda} \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

- Which gives:

$$0 = \underbrace{\sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{= N_k/\pi_k} + \lambda$$

- Multiply both sides by π_k and sum over k making use of the constraint $\sum_k \pi_k = 1$, we get:

$$\sum_{n=1}^{N} \cancelto{1}{\frac{\sum_k \pi_k \mathcal{N}(\mathrm{x}_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathrm{x}_n|\mu_j, \Sigma_j)}} = -\lambda \cancelto{1}{\sum_k \pi_k} \quad \Longrightarrow \quad \lambda = -N$$

Maximum Log-Likelihood

- Using $\lambda = -N$ to eliminate λ and rearranging, we get:

$$\boxed{\pi_k = \frac{N_k}{N},} \quad \text{where} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Not closed-form since responsibility is a function of π_k, μ_k, Σ_k!

- This is the average responsibility which the k^{th} component takes for explaining the data points.

Maximum Log-Likelihood

- The maximum log-likelihood estimates of the unknown parameter do not constitute a closed-form solution because of the responsibilities $\gamma(z_{nk})$.
- However, these results do suggest a simple iterative scheme for finding a solution to the maximum likelihood problem!

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function w.r.t. the parameters $\theta = \{\pi_k, \mu_k, \Sigma_k\}$.

1. Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood.

2. Expectation Step: Evaluate the responsibilities $\gamma(\mathrm{Z})$ using the current parameter values

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

$\gamma(\mathrm{Z})$ is a $N \times K$ table where each entry is $\gamma(z_{nk})$

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function w.r.t. the parameters $\theta = \{\pi_k, \mu_k, \Sigma_k\}$.

3. Maximization Step: Re-estimate the parameters using the current responsibilities

$$\begin{aligned}
\boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \\
\boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \\
\pi_k^{\text{new}} &= \frac{N_k}{N}
\end{aligned}$$

where

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

EM for Gaussian Mixtures

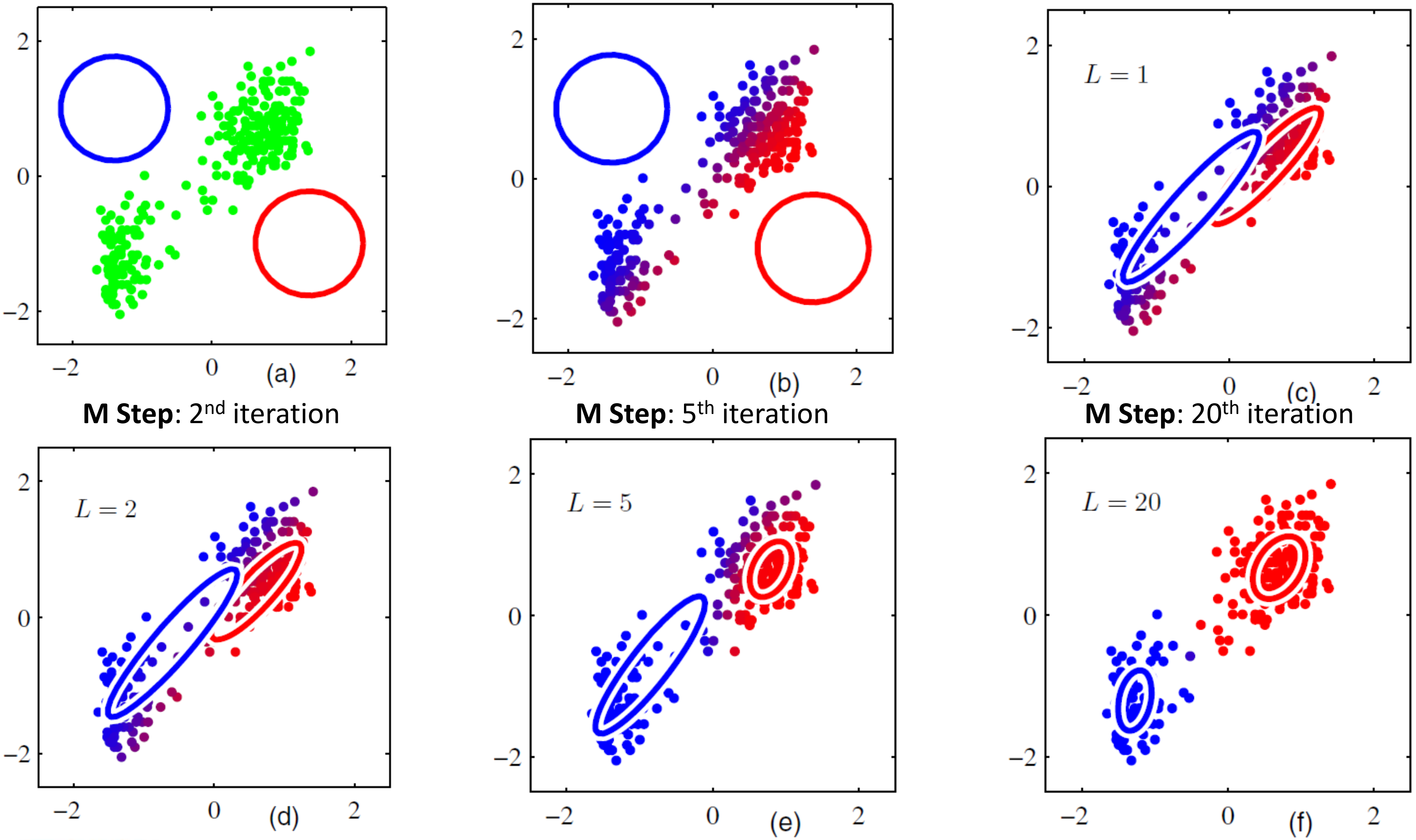
Given a Gaussian mixture model, the goal is to maximize the likelihood function w.r.t. the parameters $\theta = \{\pi_k, \mu_k, \Sigma_k\}$.

4. Evaluate the log likelihood:

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

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is NOT satisfied return to step 2.

Illustration of the EM Algorithm



Initialization: random μ_k, identity Σ_k and $\pi_k = 0.5$

E Step: 1st iteration
(color of dots used to illustrate the strength of the assignment posterior)

M Step: 1st iteration

M Step: 2nd iteration

M Step: 5th iteration

M Step: 20th iteration

Image source: "Pattern recognition and machine learning", Christopher Bishop

EM for GMM: Initialization

- Run the *K*-means algorithm first to find a suitable initialization for a GMM that is subsequently adapted using EM.

Remarks:

- EM algorithm takes many more iterations to reach (approximate) convergence compared with the *K*-means algorithm.

- And each cycle requires significantly more computation.

The General EM Algorithm

- The goal of the EM algorithm is to find maximum likelihood solutions for models having latent variables.

- The log-likelihood function is given by:

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

Discrete latent variable

or

$$\ln p(\mathrm{X}|\theta) = \ln\left\{\int_{\mathrm{Z}} p(\mathrm{X},\mathrm{Z}|\theta)\right\}$$

Continuous latent variable

where

- X: set of all observed data with n^{th} row represents x_n^T
- Z: set of all latent variables with corresponding row z_n^T
- θ: set of all model parameters

The General EM Algorithm

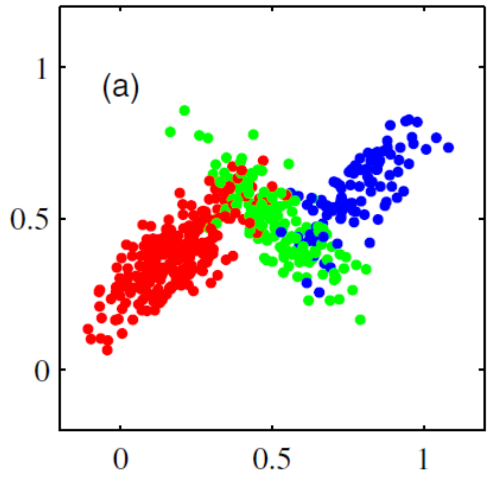
$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \right\} \quad \text{or} \quad \ln p(\mathrm{X}|\theta) = \ln \left\{ \int_{\mathrm{Z}} p(\mathrm{X}, \mathrm{Z}|\theta) \right\}$$

- A key observation is that the summation/integration over the latent variables appears inside the logarithm.

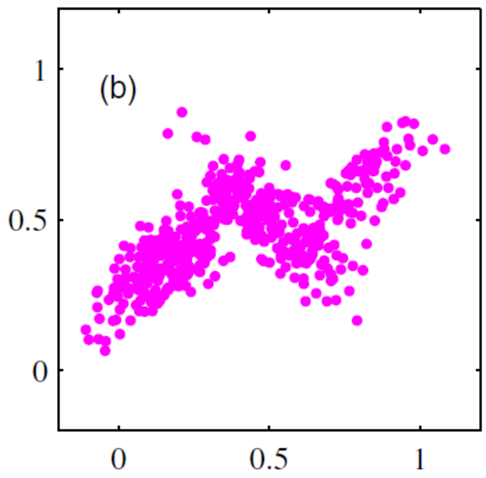
- Marginal distribution $p(\mathrm{X}|\theta)$ does not simplify even if the joint distribution $p(\mathrm{X}, \mathrm{Z}|\theta)$ belongs to the exponential family, e.g. Gaussian.

- Resulting in complicated (non closed-form) expressions for the maximum likelihood solution.

The General EM Algorithm



Assignment of x_n into clusters is known: Complete data



Assignment of x_n into clusters is unknown: Incomplete data

- Complete data: Both observation X and latent variable Z are known.

- Incomplete data: Observation X is known, but latent variable Z is unknown.

Image source: "Pattern recognition and machine learning", Christopher Bishop

The General EM Algorithm

- It is straightforward to maximize the likelihood function for the complete data set $\ln p(\mathrm{X}, \mathrm{Z}|\theta)$.

- In practice, however, we are not given the complete data set $\{\mathrm{X}, \mathrm{Z}\}$, but only the incomplete data X.

- Instead, we consider maximization of the expected value of the complete data log-likelihood $\ln p(\mathrm{X}, \mathrm{Z}|\theta)$ w.r.t. $p(\mathrm{Z}|\mathrm{X}, \theta)$.

- Do the Expectation and Maximization steps iteratively until convergence.

The General EM Algorithm

Expectation Step:

- Use the current parameter values θ^{old} to find the posterior distribution of the latent variables given by $p(\mathrm{Z}|\mathrm{X}, \theta^{old})$.

- We then use $p(\mathrm{Z}|\mathrm{X}, \theta^{old})$ to find the expectation of the complete-data log likelihood evaluated for some general parameter value θ.

- This expectation, denoted $\mathcal{Q}(\theta, \theta^{old})$, is given by:

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

$$= \mathbb{E}_{\mathrm{Z}|\mathrm{X},\theta^{old}}[\ln p(\mathrm{X}, \mathrm{Z}|\theta)]$$

The General EM Algorithm

Maximization Step:

- Determine the revised parameter estimate θ^{new} by maximizing this function:

$$
\begin{aligned}
\boldsymbol{\theta}^{\text{new}} &= \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \\
&= \arg\max_{\boldsymbol{\theta}} \underbrace{\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}_{\text{Log is now inside the summation!}}
\end{aligned}
$$

- Since the logarithm now acts directly on the joint distribution $p(\mathrm{X}, \mathrm{Z}|\theta)$, the corresponding M-step will be tractable.

The General EM Algorithm

- **Given**: a joint distribution $p(\mathrm{X}, \mathrm{Z}|\theta)$ over observed variables X and latent variables Z, governed by parameters θ.

- **Goal**: is to maximize the likelihood function $p(\mathrm{X}|\theta)$ with respect to θ.

The General EM Algorithm

1. Choose an initial setting for the parameters θ^{old}.
2. Expectation step: Evaluate $p(\mathrm{Z}|\mathrm{X}, \theta^{old})$.
3. Maximization step: Evaluate θ^{new} given by:

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

where

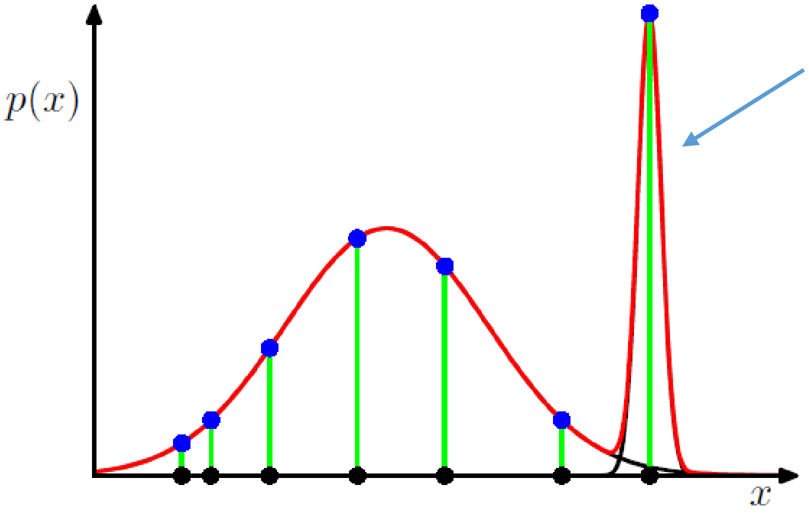
$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

4. Check for convergence of either the log likelihood or the parameter values, if not converged:

$$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$$

Singularities and MAP

Illustration of the “singularity problem”:



A Gaussian gets erroneously fitting to 1 single data, i.e. $\Sigma_k \rightarrow 0$

- Problem can be alleviated by applying MAP on $Q(\theta, \theta^{old})$.

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \ln p(\boldsymbol{\theta}) \leftarrow$$

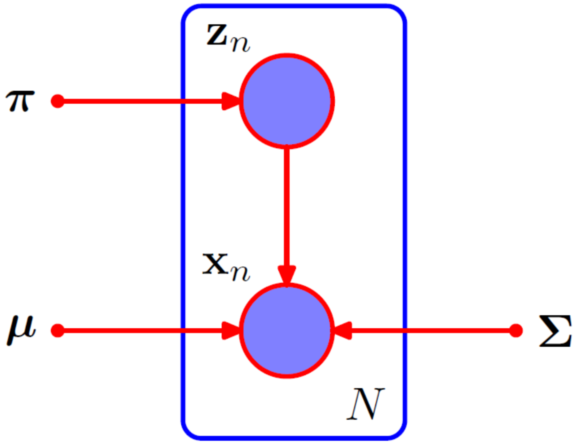
Dirichlet prior on π_k, and normalized inversed Gaussian on (μ_k, σ_k).

Image source: “Pattern recognition and machine learning”, Christopher Bishop

Gaussian Mixture Revisited

- The complete data log-likelihood is given by:

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



$$= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \underbrace{\{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}}$$

Now the log is inside the summation!

Image source: "Pattern recognition and machine learning", Christopher Bishop

Gaussian Mixture Revisited

Complete data log-likelihood:

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

- We can switch the sums and only look at the points associated with the respective mixture component k.

- Maximization of each (μ_k, Σ_k) can be done separately, similarly we can solve for π_k by enforcing the sum to one constraint using Lagrange multiplier.

- **Problem:** the latent variables Z are not observed!!!

Gaussian Mixture Revisited

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

$$\mathbb{E}_Z[\ln p(X, Z|\theta)] = \textstyle\sum_Z p(Z|X, \theta^{old}) \ln p(X, Z|\theta)$$

$$= \textstyle\sum_n \sum_k p(z_{nk}|x_n, \theta^{old}) \ln p(x_n, z_{nk}|\theta)$$

$$= \textstyle\sum_n \sum_k \frac{p\left(x_n \middle| z_{nk}, \theta^{old}\right) p(z_{nk}|\theta^{old})}{p(x_n|\theta^{old})} \ln p(x_n, z_{nk}|\theta)$$

(Bayes Rule)

$$= \textstyle\sum_n \sum_k \underbrace{\frac{\mathcal{N}\left(x_n \middle| \mu_k^{old}, \Sigma_k^{old}\right) \pi_k^{old}}{\sum_j \mathcal{N}\left(x_n \middle| \mu_j^{old}, \Sigma_j^{old}\right) \pi_j^{old}}}_{\gamma(z_{nk})} \ln p(x_n, z_{nk}|\theta)$$

Responsibility that stays fixed

$$= \textstyle\sum_n \sum_k \gamma(z_{nk}) \left[\ln \pi_k + \ln \mathcal{N}(x_n|\mu_k, \Sigma_k)\right]$$

Gaussian Mixture Revisited

$$\mathbb{E}_z[\ln p(X, Z|\theta)] = \sum_k \sum_n \gamma(z_{nk}) \left[\ln \pi_k + \ln \mathcal{N}(x_n|\mu_k, \Sigma_k)\right]$$

- Similar to the complete log-likelihood case, we can switch the sums and only look at the points associated with the respective mixture component k.
- Since $\gamma(z_{nk})$ stays fixed, maximization of each (μ_k, Σ_k) can be done separately, similarly we can solve for π_k by enforcing the sum to one constraint using Lagrange multiplier.

The Theory Behind EM Algorithm

- Maximizing the log-likelihood $\ln p(X|\theta)$ is difficult due to the need of marginalizing over the latent variables inside the logarithm:

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

- It turns out that $\ln p(X|\theta)$ can be maximized by maximizing its lower bound $\mathcal{L}(q, \theta)$ within the EM algorithm.

The Theory Behind EM Algorithm

- Let us rewrite the log-likelihood into:

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \underbrace{\mathcal{L}(q, \boldsymbol{\theta})}_{\text{Lower bound}} + \underbrace{\mathrm{KL}(q\|p)}_{\text{KL Divergence}}$$

where

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

$$\mathrm{KL}(q\|p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})}{\underbrace{q(\mathbf{Z})}_{q(Z)\text{ is a distribution we defined over the latent variable}}} \right\} \geq 0$$

The Theory Behind EM Algorithm

Proof:

$$\mathcal{L}(q,\theta) + \mathrm{KL}(q||p) = \sum_{Z} q(Z)\ln\left\{\frac{p(X,Z|\theta)}{q(Z)}\right\} - \sum_{Z} q(Z)\ln\left\{\frac{p(Z|X,\theta)}{q(Z)}\right\}$$

$$= \sum_{Z} q(Z)\{\underbrace{\ln p(X,Z|\theta)}_{\ln p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}) + \ln p(\mathbf{X}|\boldsymbol{\theta})} - \ln \cancel{q(Z)} - \ln p(Z|X,\theta) + \ln \cancel{q(Z)}\}$$

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

$$= \ln p(X|\theta)$$

□

The Theory Behind EM Algorithm

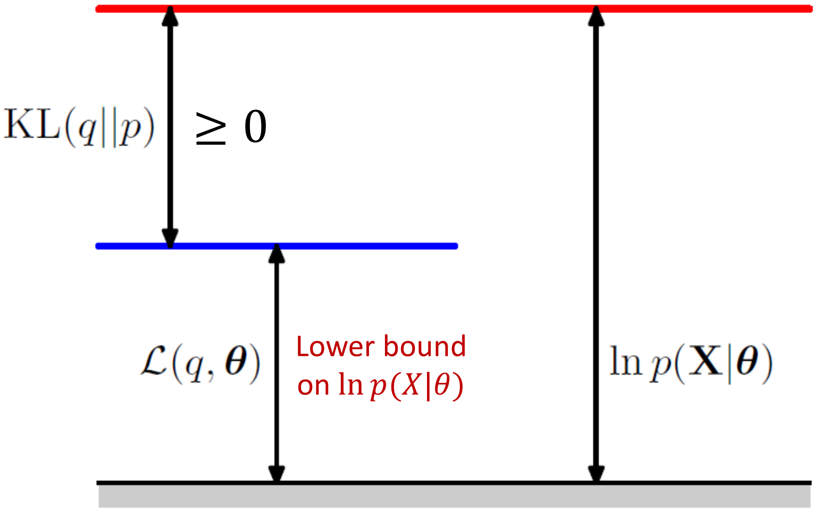

$\mathrm{KL}(q||p)$ ≥ 0
$\mathcal{L}(q, \boldsymbol{\theta})$
Lower bound on $\ln p(X|\theta)$
$\ln p(\mathbf{X}|\boldsymbol{\theta})$

Illustration of the decomposition given by:

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

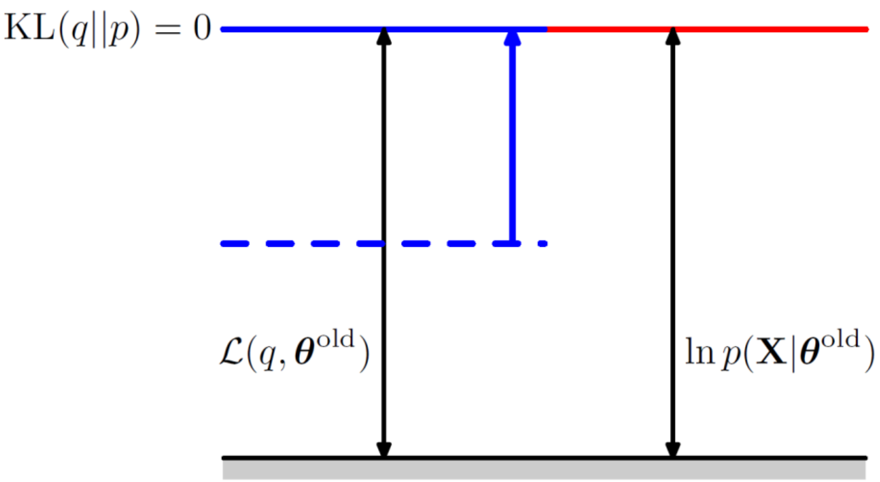
which holds for any choice of $q(\boldsymbol{Z})$.

- Because the Kullback-Leibler divergence satisfies $\mathrm{KL}(q \parallel p) \geq 0$, we see that the quantity $\mathcal{L}(q, \theta)$ is a lower bound on the log-likelihood function $\ln p(X|\theta)$.

Image source: “Pattern recognition and machine learning”, Christopher Bishop

The Theory Behind EM Algorithm

Illustration of the E-Step:


$\mathrm{KL}(q||p) = 0$
$\mathcal{L}(q, \boldsymbol{\theta}^{\text{old}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\text{old}})$

Lower bound $\mathcal{L}(q, \theta)$ is maximized by choosing $q(Z) = p(Z|X, \theta^{old})$!

Proof:

$$\mathrm{KL}(q||p) = -\sum_{Z} q(Z) \ln \cancelto{0}{\left\{\frac{p(Z|X,\theta)}{p(Z|X,\theta)}\right\}}$$

$$= 0$$

$\Longrightarrow \mathcal{L}(q, \theta)$ must be at its maximum since

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

□

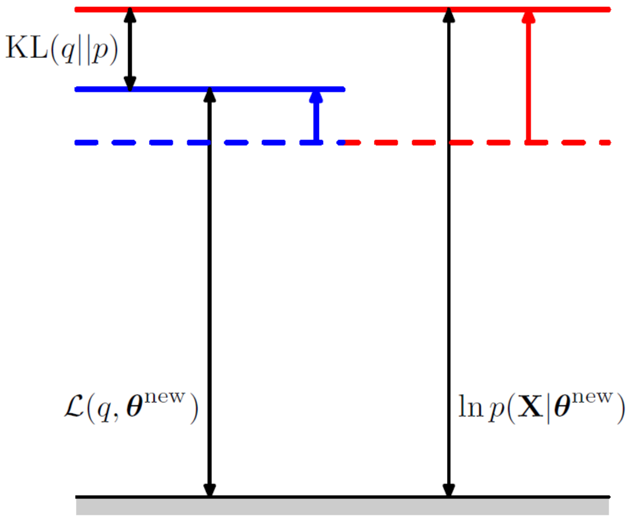
- Equivalence of expectation under the latent variable distribution:

$$\begin{aligned}\mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \\ &= \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \text{const}\end{aligned}$$

Image source: “Pattern recognition and machine learning”, Christopher Bishop

The Theory Behind EM Algorithm

Illustration of the M-Step:


$\mathrm{KL}(q||p)$
$\mathcal{L}(q, \boldsymbol{\theta}^{\mathrm{new}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\mathrm{new}})$

$$\mathcal{L}(q, \theta^{new}) = \sum_{Z} p(Z|X, \theta^{old}) \ln p(X, Z|\theta^{new}) + \text{const}$$

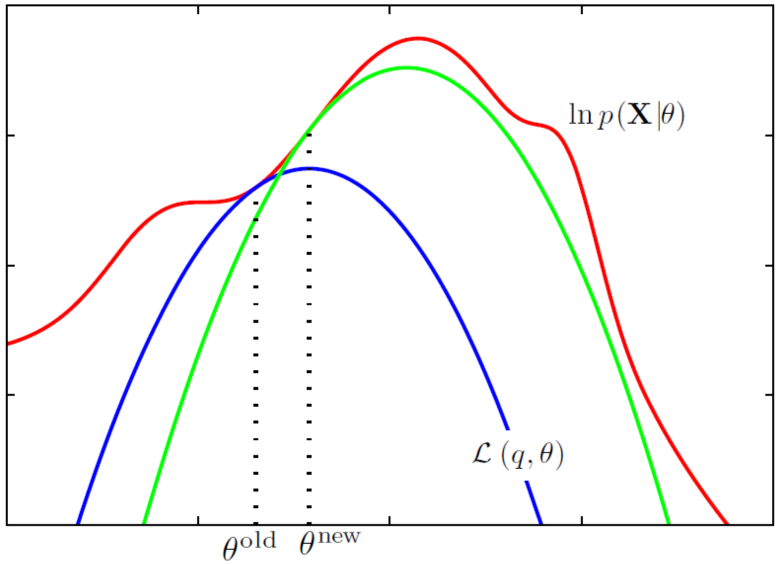
$$\mathrm{KL}(q||p) = -\sum_{Z} p(Z|X, \theta^{old}) \ln \left\{ \frac{p(Z|X, \theta^{new})}{p(Z|X, \theta^{old})} \right\}$$

- The distribution $q(Z)$ is held fixed and the lower bound $\mathcal{L}(q, \boldsymbol{\theta})$ is maximized w.r.t θ to give a revised value θ^{new}.

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

Image source: “Pattern recognition and machine learning”, Christopher Bishop

The Theory Behind EM Algorithm



- **E-step**: we compute the convex lower bound given the old parameters θ^{old} (blue curve).
- **M-step**: we maximize this lower bound to get new parameters θ^{new}.
- This is repeated (green curve) until convergence.

Image source: "Pattern recognition and machine learning", Christopher Bishop

Summary

- We have looked at how to:

1. Use the non-probabilistic k-mean algorithm to solve the clustering problem.

2. Describe the Gaussian-mixture model.

3. Apply the Expectation-Maximization algorithm for estimation of both the unknown parameters and latent variables.