NUS National University of Singapore | School of Computing

CS5340
Uncertainty Modeling in AI

Lecture 7:
Mixture Models and the EM Algorithm
(or "Learning with Missing Data")

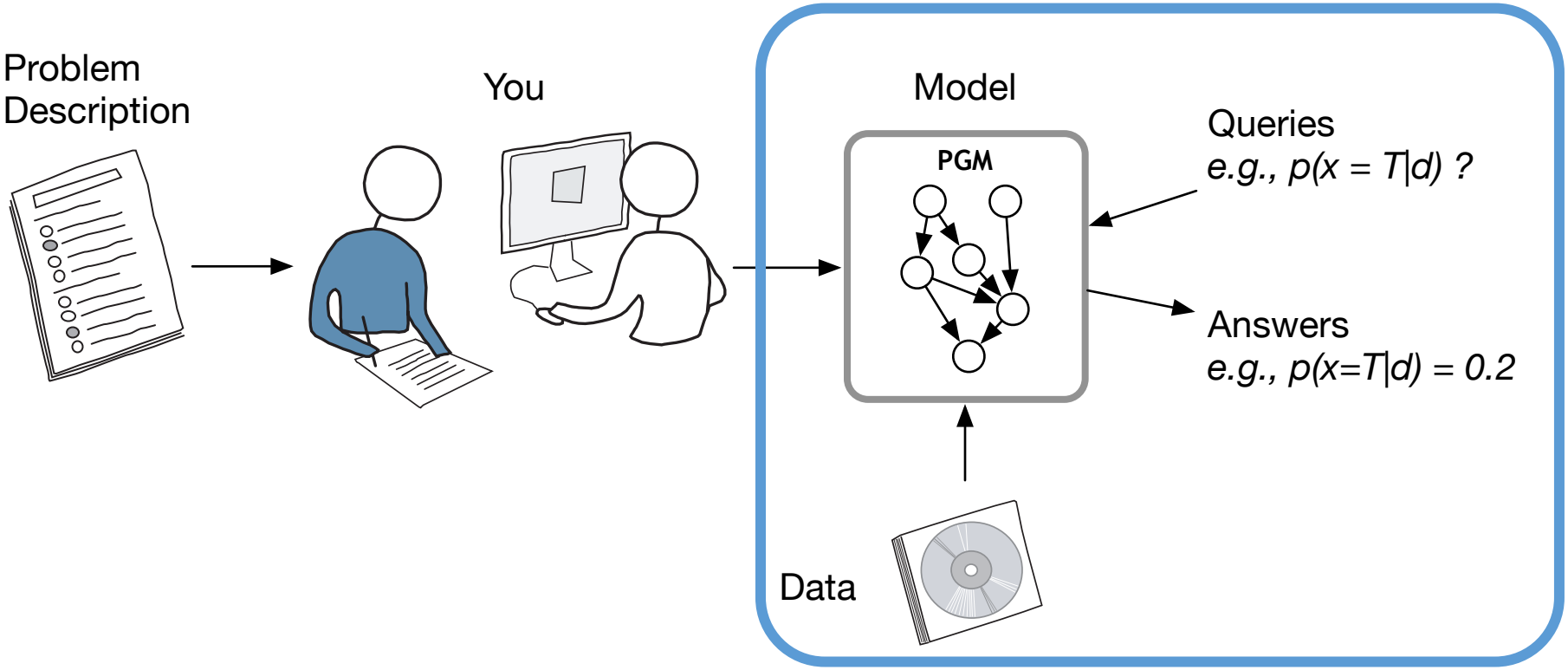
Asst. Prof. Harold Soh

AY 2022/23

Semester 2

CS5340 in a nutshell

CS5340 is about how to "**represent**" and "**reason**" with uncertainty in a computer.



Course Schedule

Week	Date	Lecture Topic	Tutorial Topic
1	12 Jan	Introduction to Uncertainty Modeling + Probability Basics	Introduction
2	19 Jan	Simple Probabilistic Models	Probability Basics
3	26 Jan	Bayesian networks (Directed graphical models)	More Basic Probability
4	2 Feb	Markov random Fields (Undirected graphical models)	DGM modelling and d-separation
5	9 Feb	Variable elimination and belief propagation	MRF + Sum/Max Product
6	16 Feb	Factor graph and the junction tree algorithm	**Quiz 1**
-	-	RECESS WEEK	
7	2 Mar	Mixture Models and Expectation Maximization (EM)	Linear Gaussian Models
8	9 Mar	Hidden Markov Models (HMM)	Probabilistic PCA
9	**16** Mar	Monte-Carlo Inference (Sampling)	Linear Gaussian Dynamical System
10	23 Mar	Variational Inference	MCMC + Sequential VAE
11	30 Mar	Inference and Decision-Making (Special Topic)	**Quiz 2**
12	6 Apr	Gaussian Processes (Special Topic)	Wellness Day
13	13 Apr	**Project Presentations**	Closing

Learning Outcomes

- Students should be able to:

1. Use the non-probabilistic k-means 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.

4. Explain why the EM algorithm works.

Acknowledgements

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

1. "Pattern Recognition and Machine Learning", Christopher Bishop, Chapter 8 & 9.
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.
6. Gim Hee's slides

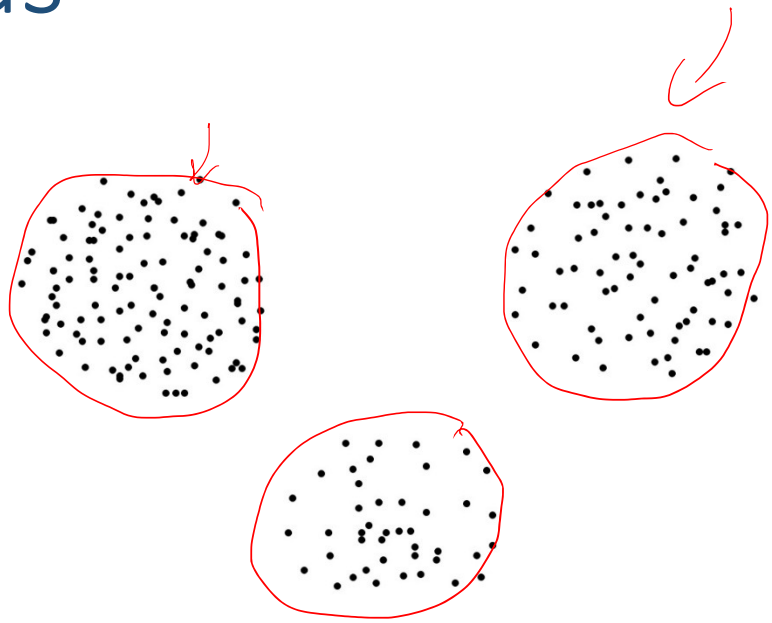
NUS National University of Singapore | School of Computing

Clustering & K-Means

A non-probabilistic clustering algorithm

Key Ideas

- The Clustering Problem
  - Has many applications
- Problem-Solving Approach
- Non-probabilistic Algorithm
  - K-means (iterative method)
  - Limitations of k-means



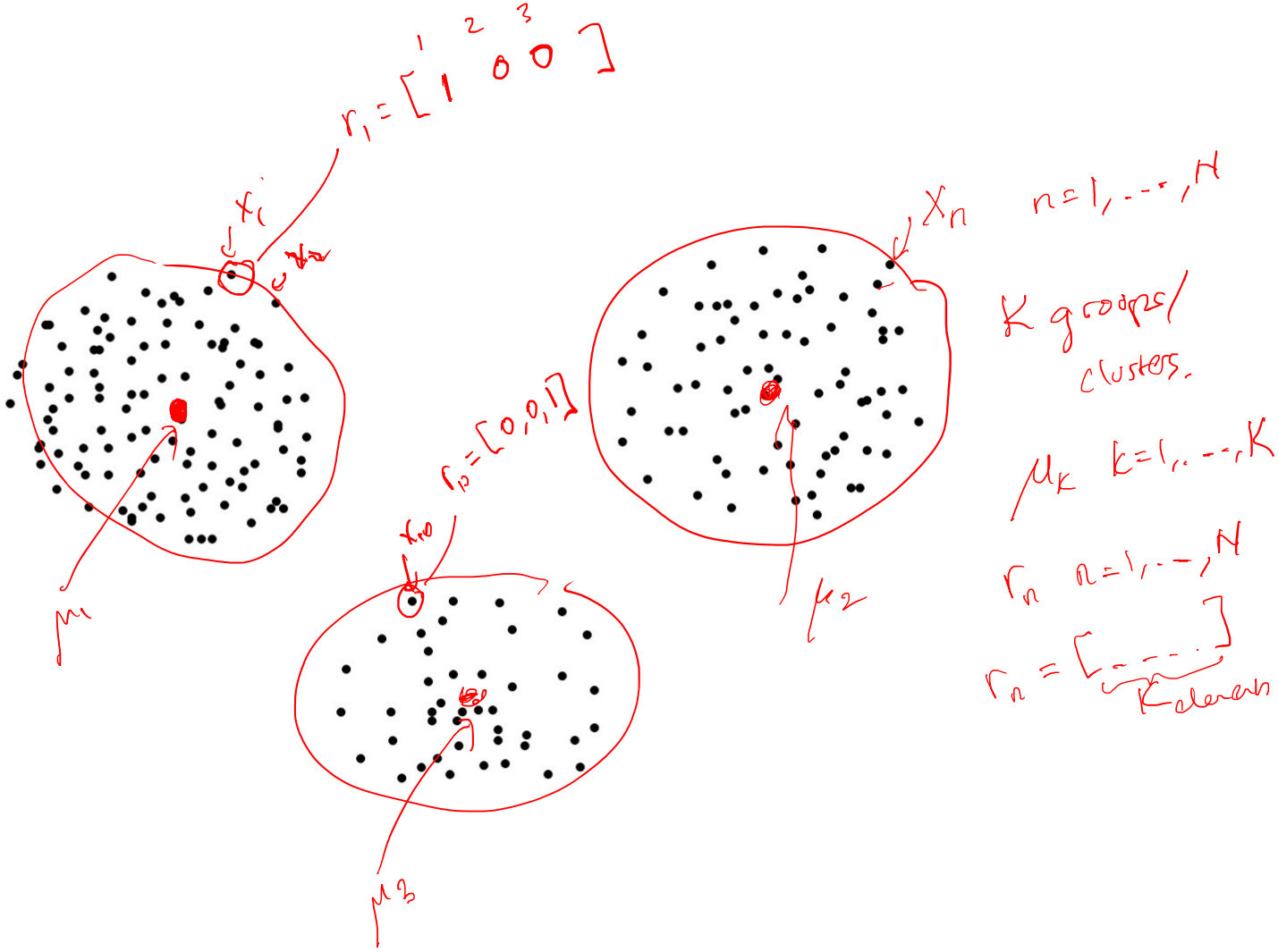
One Problem Solving Approach

1. What are the variables in our problem?
   - Write down the variables precisely
2. What are we interested in finding?
   - Have an intuitive notion of what we want
3. How should we formalize the problem?
   - "Convert the intuition to math"
   - Commonly as a optimization problem
4. How can we solve the optimization problem?
   - Derive an algorithm to solve our problem.

One Problem Solving Approach

1. What are the variables in our problem?
   - Write down the variables precisely
2. What are we interested in finding?
   - Have an intuitive notion of what we want
3. How should we formalize the problem?
   - "Convert the intuition to math"
   - Commonly as a optimization problem
4. How can we solve the optimization problem?
   - Derive an algorithm to solve our problem.

Variables of Interest



Variables of Interest

1. Data set $\{x_1, \ldots x_N\}$ of observations (each $x_n \in \mathbb{R}^D$)
2. Number of clusters K
3. The K cluster centers $\{\mu_1, \ldots, \mu_K\}$, i.e. unknown parameters (assume each cluster is a circle and $\mu_k \in \mathbb{R}^D$)
4. Assignment $\{r_1, \ldots, r_N\}$ of each point x_n to a cluster center k, i.e. data association

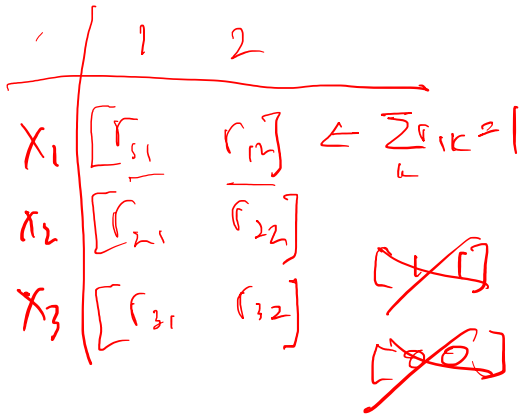
Variables of Interest

Assignments: 1-of-K coding

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

$$r_{nk} \in \{0,1\} \;\; \forall k = 1, \ldots K,$$

$$\text{s.t.} \quad \sum_k r_{nk} = 1. \quad \text{(1-of-}K\text{ constraint)}$$



- which of the K clusters the x_n is assigned to.

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

One Problem Solving Approach

1. What are the variables in our problem?
   - Write down the variables precisely
2. What are we interested in finding?
   - Have an intuitive notion of what we want
3. How should we formalize the problem?
   - "Convert the intuition to math"
   - Commonly as a optimization problem
4. How can we solve the optimization problem?
   - Derive an algorithm to solve our problem.

What do we want?

1. Want unknown parameters $\{\mu_1, \ldots, \mu_K\}$ of the clusters (the cluster centers).
2. And which cluster does each data belong to $\{r_1, \ldots, r_K\}$

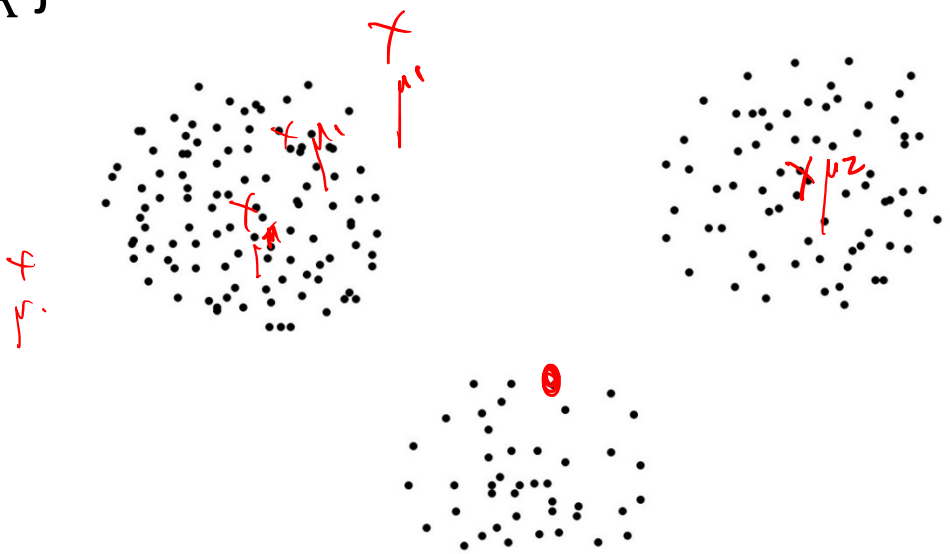
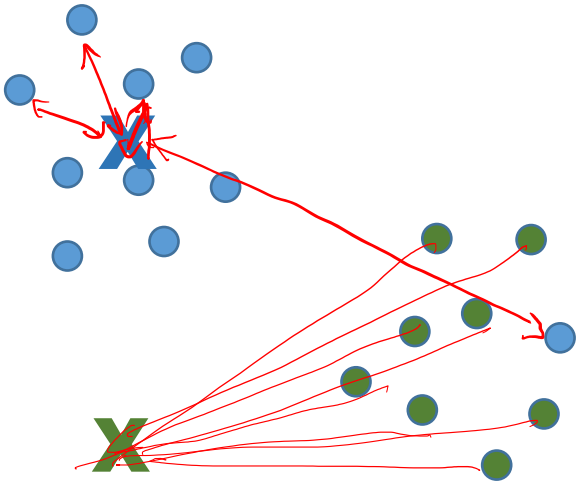


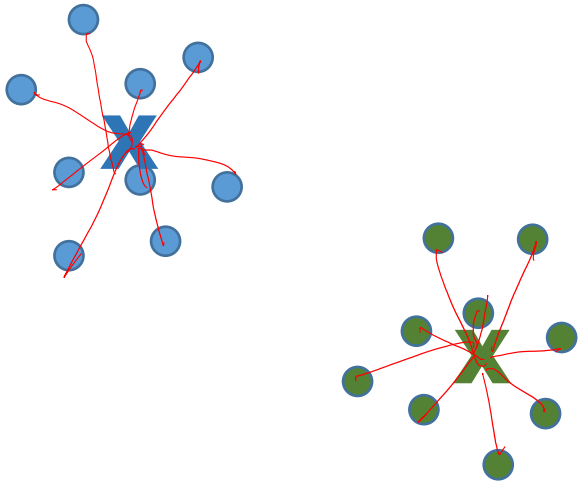
Image source: https://stab-iitb.org/electronics-club/blog/2016/05/deep-learning-based-image-classification/

What do we want?

Bad Clustering



Good Clustering



One Problem Solving Approach

1. What are the variables in our problem?
   - Write down the variables precisely
2. What are we interested in finding?
   - Have an intuitive notion of what we want
3. How should we formalize the problem?
   - "Convert the intuition to math"
   - Commonly as a optimization problem
4. How can we solve the optimization problem?
   - Derive an algorithm to solve our problem.

Formalizing our Problem

Given:

1. Data set $\{x_1, \ldots x_N\}$ 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 $\{r_1, \ldots, r_K\}$ of each point x_n to a cluster center k, i.e. data association

Objective Function

- The **objective**: find $\{r_{nk}\}$ and $\{\mu_k\}$ to minimize a "distortion measure":

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

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

One Problem Solving Approach

1. What are the variables in our problem?
   - Write down the variables precisely
2. What are we interested in finding?
   - Have an intuitive notion of what we want
3. How should we formalize the problem?
   - "Convert the intuition to math"
   - Commonly as a optimization problem
4. How can we solve the optimization problem?
   - Derive an algorithm to solve our problem.

Formalizing our Problem

Given:

1. Data set $\{x_1, \ldots x_N\}$ of observations
2. Number of clusters K

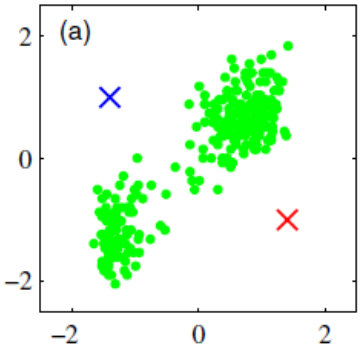
Find:

1. The K cluster centers $\{\mu_1, \ldots, \mu_K\}$, i.e. unknown parameters (assume each cluster is a circle)
2. Assignment $\{r_1, \ldots, r_K\}$ of each point x_n to a cluster center k, i.e. data association

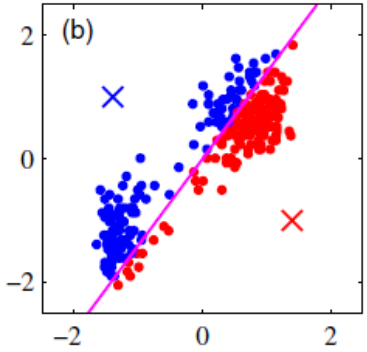
Minimize: $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$, s.t. $\sum_k r_{nk} = 1$.

K-means: Intuitive idea

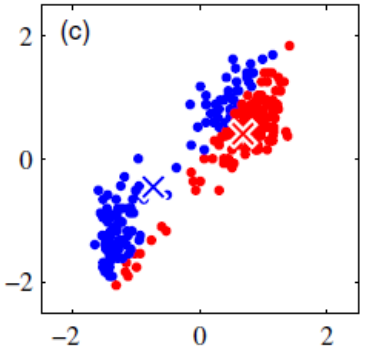
Initialization ($K = 2$):
Choose μ_k



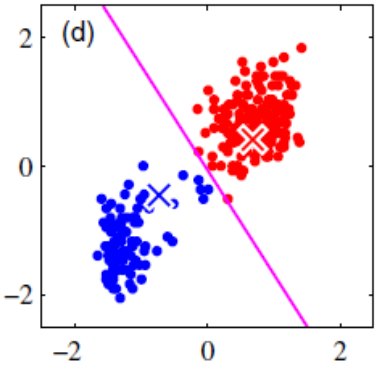
Iteration (1):
Assignment of r_{nk}



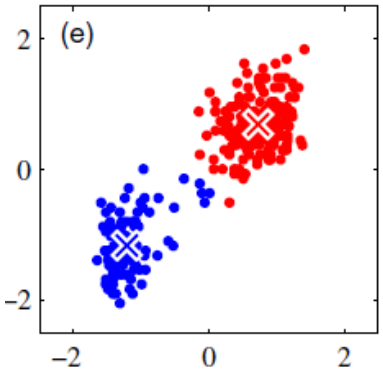
Iteration (1):
Update of μ_k



Iteration (2):
Assignment of r_{nk}



Iteration (2):
Update of μ_k



Iteration (3):
Assignment of r_{nk}

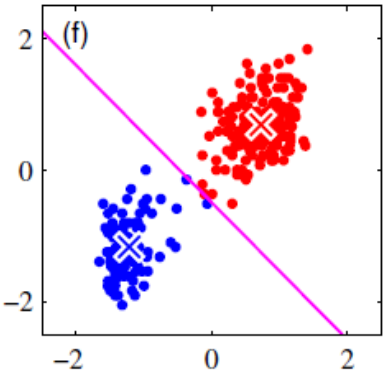


Image source: "Pattern recognition and machine learning", Christopher Bishop

Non-Probabilistic Approach: K-Means

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 for each x_n:

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

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

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

Non-Probabilistic Approach: K-Means

Update (Cluster Parameters) Step:

- We can optimize each μ_k separately

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

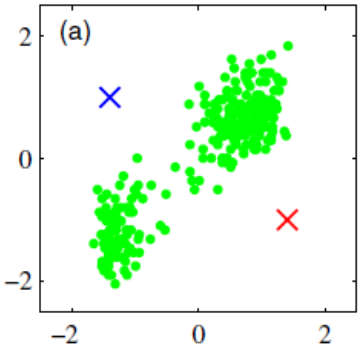
- Differentiation of L w.r.t. μ_k, and equate to zero gives:

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

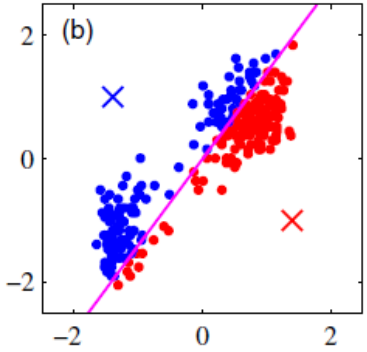
Mean of all points assigned to cluster k, hence "k-means"!

Non-Probabilistic Approach: K-Means

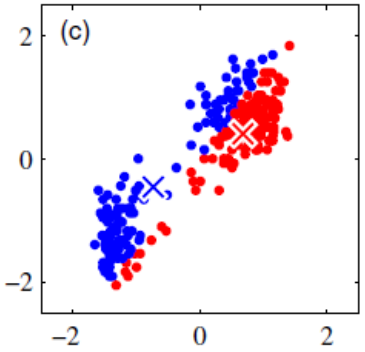
Initialization ($K = 2$):
Choose μ_k



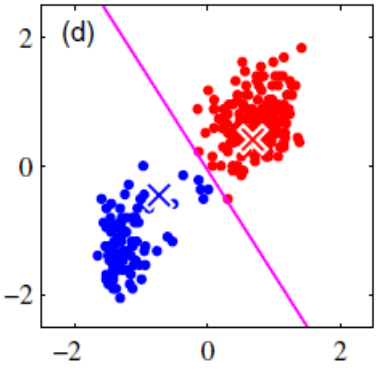
Iteration (1):
Assignment of r_{nk}



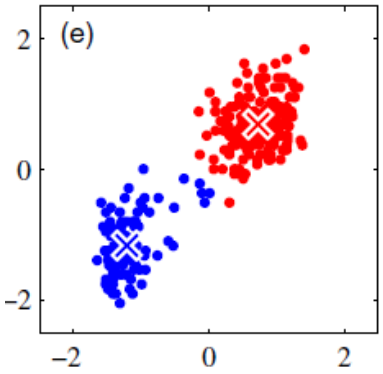
Iteration (1):
Update of μ_k



Iteration (2):
Assignment of r_{nk}



Iteration (2):
Update of μ_k



Iteration (3):
Assignment of r_{nk}

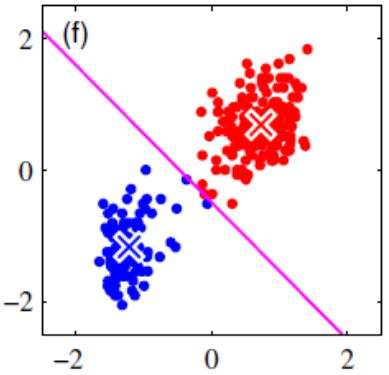
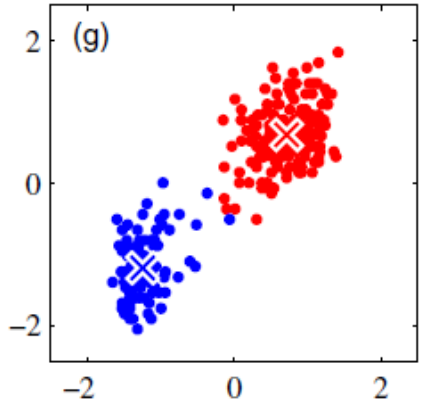


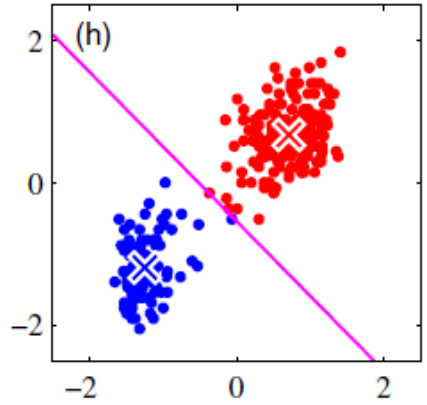
Image source: "Pattern recognition and machine learning", Christopher Bishop

Non-Probabilistic Approach: K-Means

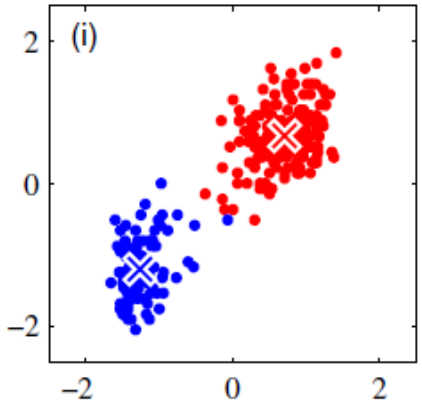
Iteration (3):
Update of μ_k

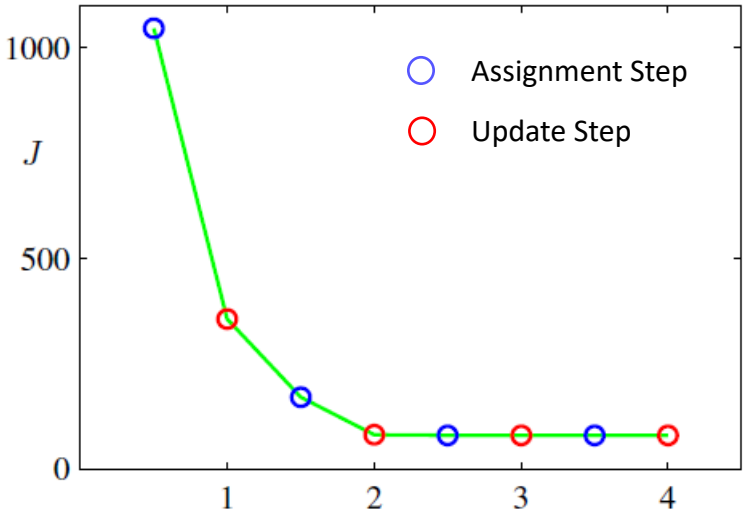


Iteration (4):
Assignment of r_{nk}



Iteration (4):
Update of μ_k





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

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

One Problem Solving Approach

1. What are the variables in our problem?
   - Write down the variables precisely
2. What are we interested in finding?
   - Have an intuitive notion of what we want
3. How should we formalize the problem?
   - "Convert the intuition to math"
   - Commonly as a optimization problem
4. How can we solve the optimization problem?
   - Derive an algorithm to solve our problem.

Some Problems with kMeans

- Convergence to local minima

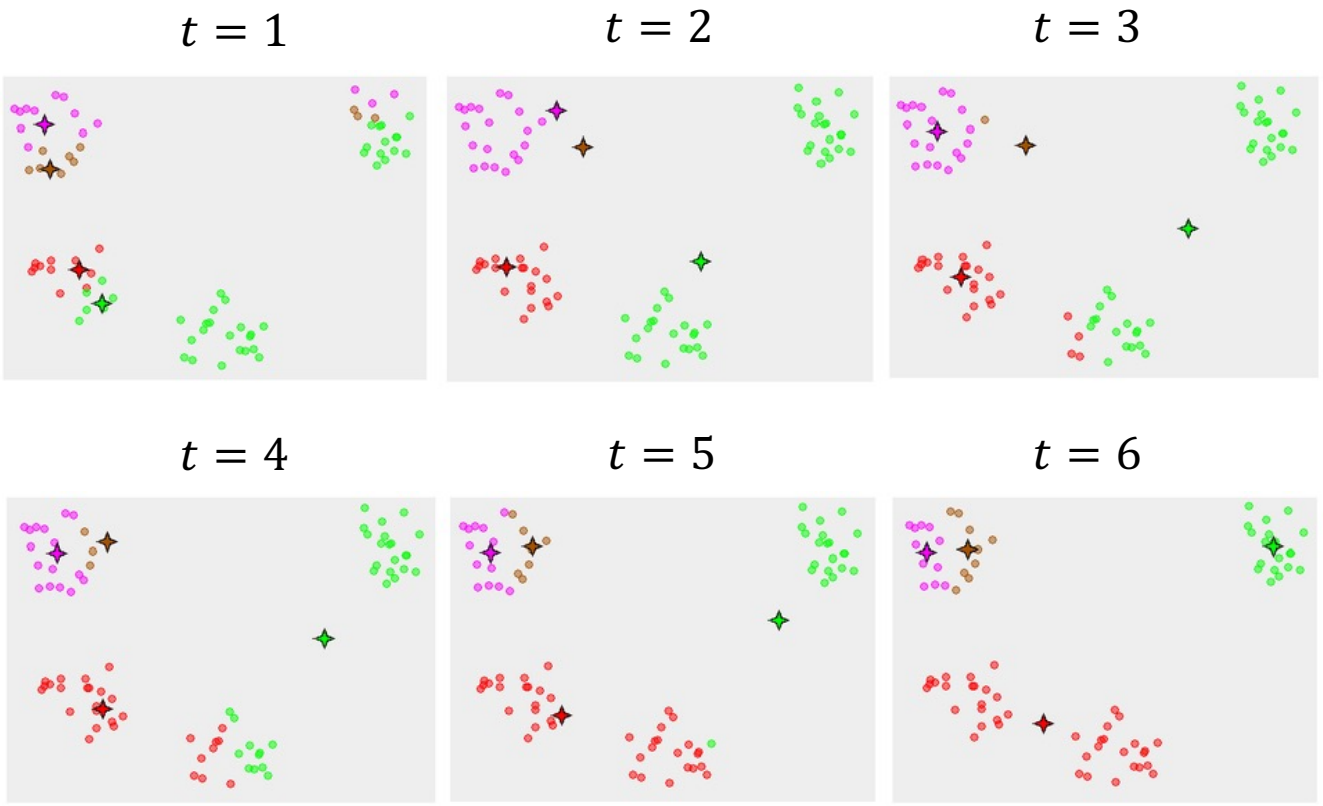
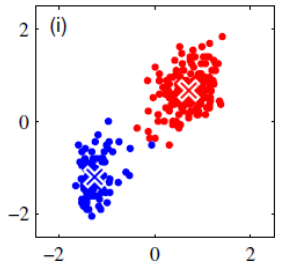


Image from: https://en.wikipedia.org/wiki/K-means_clustering

Some Problems with kMeans



- Have to choose K (number of clusters)
- Deterministic assignments ("Hard Labels")
- Spherical clusters

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

K-means

- Equal sized clusters
- Sensitive to outliers

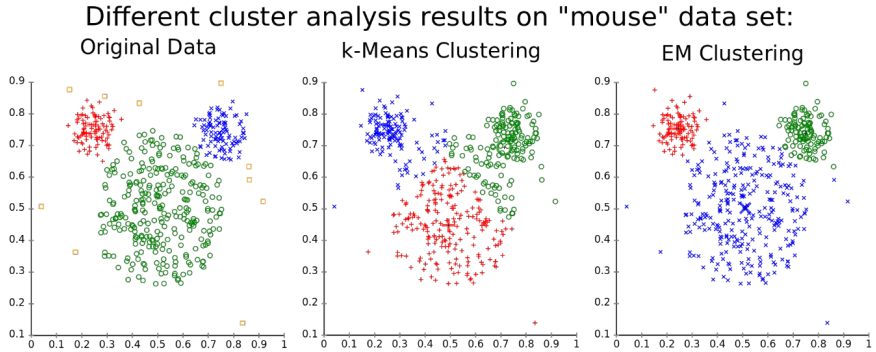
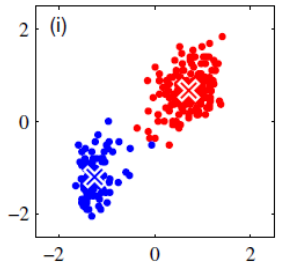


Image from: https://en.wikipedia.org/wiki/K-means_clustering
https://towardsdatascience.com/the-anatomy-of-k-means-c22340543397

Probabilistic Clustering

Introduction to Gaussian Mixture Models (GMMs)

Some Problems with kMeans



- Have to choose K (number of clusters)
- Deterministic assignments ("Hard Labels")
- Spherical clusters

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

K-means

- Equal sized clusters
- Sensitive to outliers

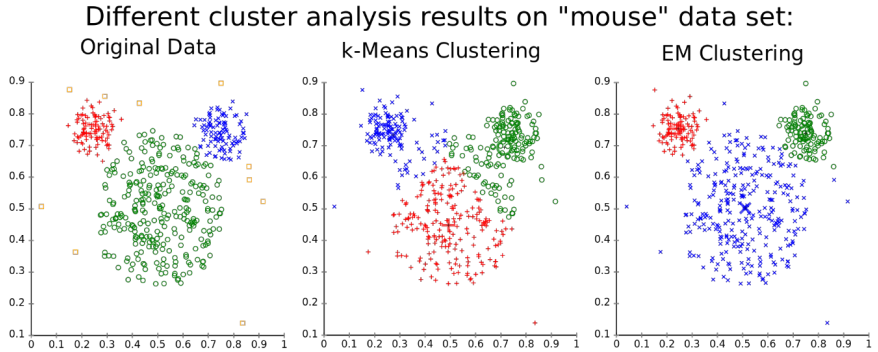


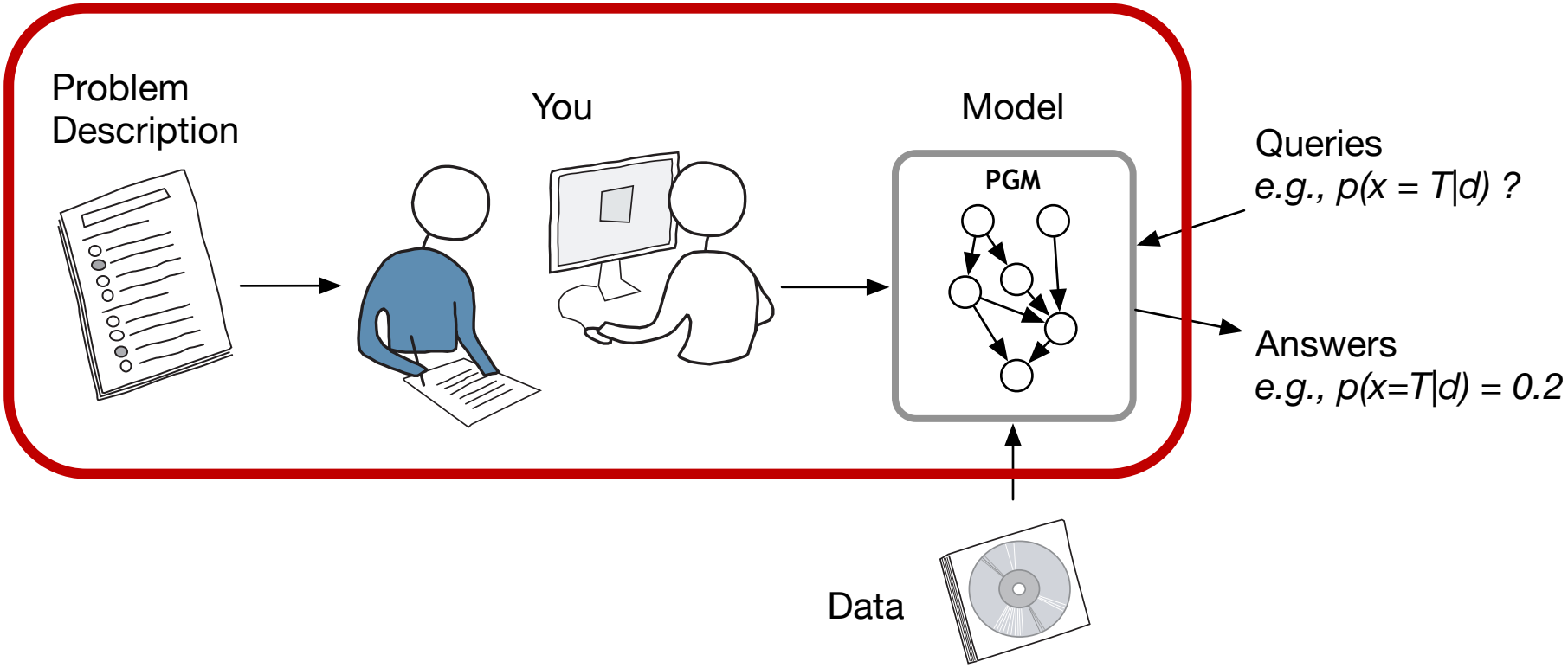
Image from: https://en.wikipedia.org/wiki/K-means_clustering
https://towardsdatascience.com/the-anatomy-of-k-means-c22340543397

Key Ideas

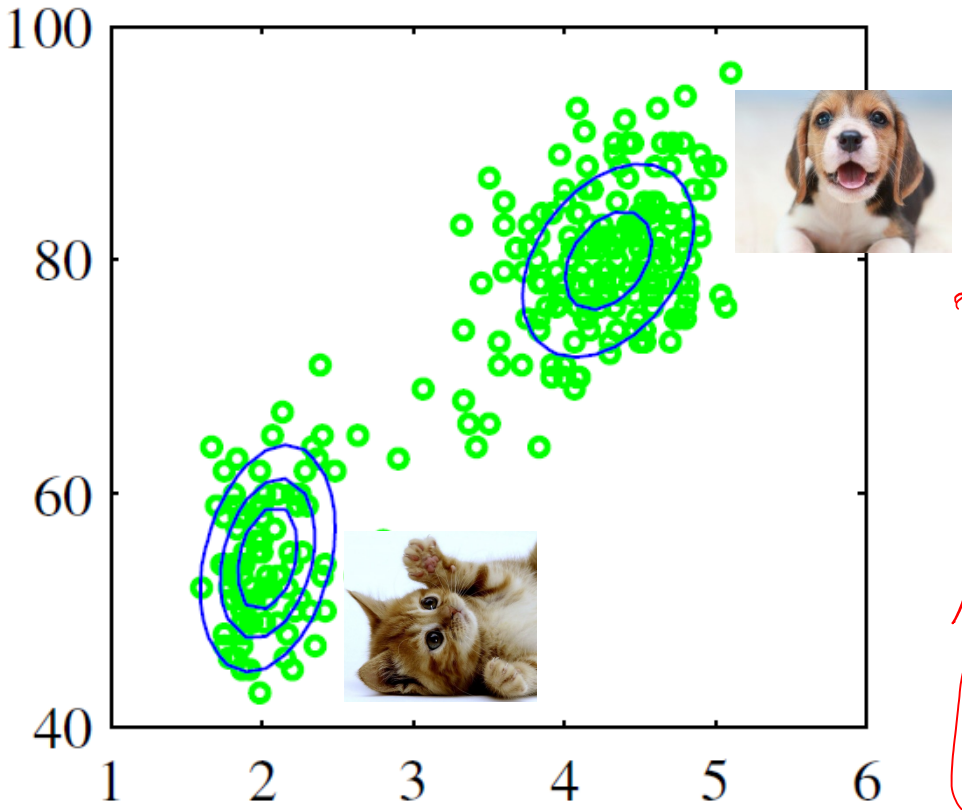
- Clustering from a Probabilistic Perspective
  - A Missing Data Problem
- Gaussian Mixture Model
  - Mixture Models in general

CS5340 in a nutshell

CS5340 is about how to "**represent**" and "**reason**" with uncertainty in a computer.



Intuition: Generative Story



$z_n \sim P(z)$

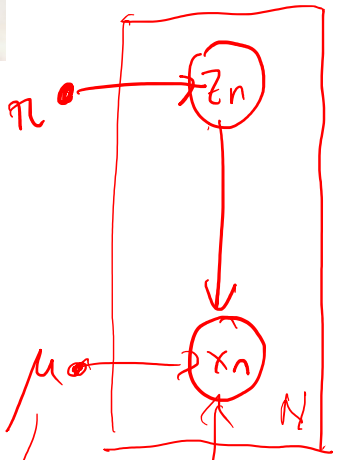
$x_n \sim P(x_n | z_n)$

$p(z)$?

$p(z) = \text{Cat}(\pi)$

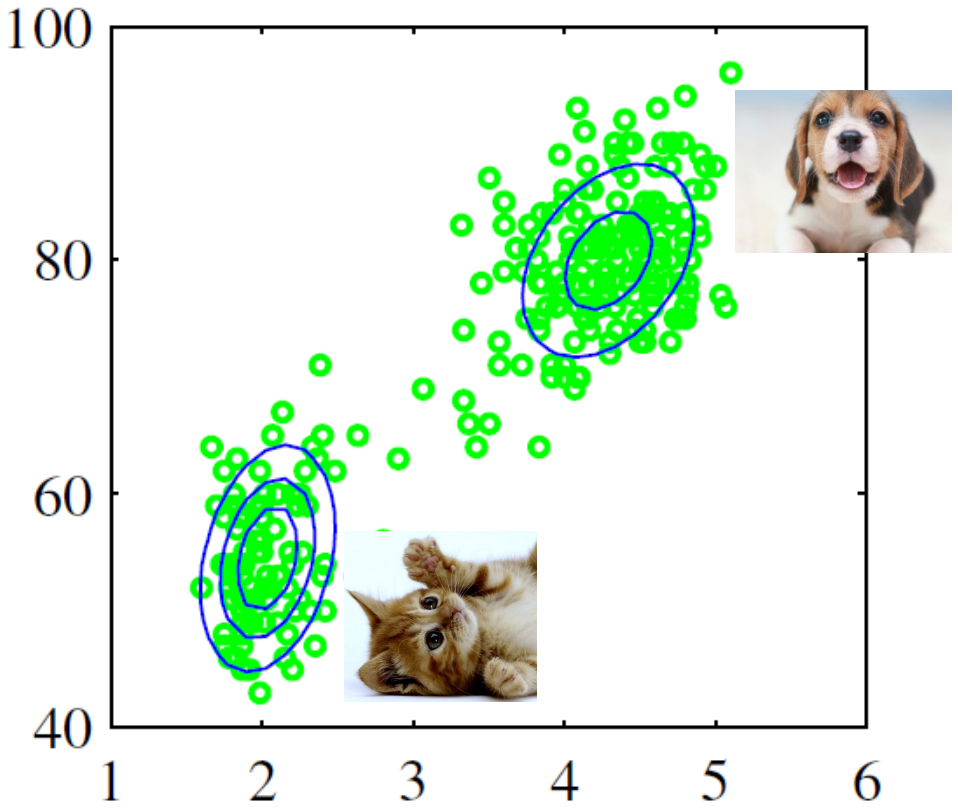
$p(x|z)$?

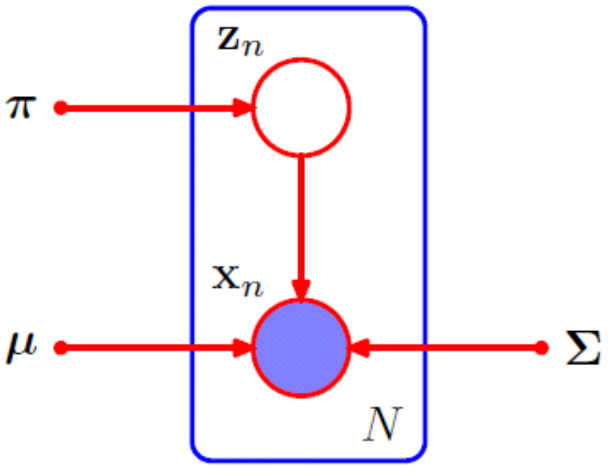
$p(x|z) = N(\mu, \Sigma)$



$\mu = \{\mu_1, \mu_2\}$ $\Sigma = \{\Sigma_1, \Sigma_2\}$

Intuition: Generative Story





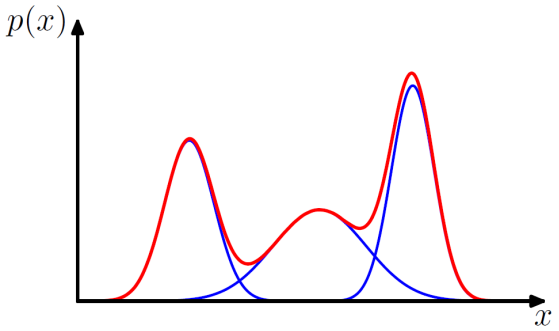
Gaussian Mixture Models

- Mixture distributions: linear combinations of more basic distributions e.g.,
  - Gaussian ⇒ Gaussian Mixture Model (GMM).
  - Poisson ⇒ Poisson Mixture Model
  - Bernoulli ⇒ Bernoulli Mixture Model

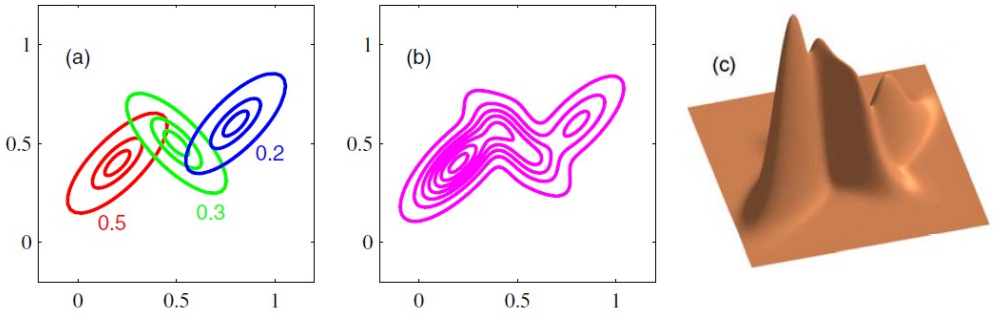
$p(x|z)$

- GMMs can approximate almost any continuous density with arbitrary accuracy.

Examples:



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



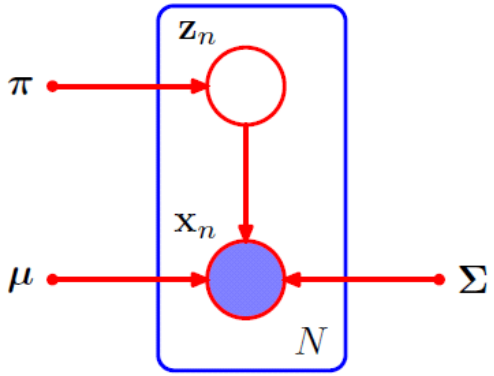
(b)-(c) 2D GMM formed by (a) 3 Gaussians

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

Gaussian Mixture Models

- The probability distribution of a GMM 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:

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

From Lecture 1 (Appendix): Multivariate Normal Distribution

- Multivariate normal distribution describes a *D*-dimensional continuous variable $\boldsymbol{X}$, i.e. $\boldsymbol{x} \in \mathbb{R}^D$.
- *D*-dimensional mean $\boldsymbol{\mu} \in \mathbb{R}^D$, and $D \times D$ symmetrical positive definite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}_+^{D \times D}$.

$$p(\boldsymbol{X} = \boldsymbol{a} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\{ -0.5(\boldsymbol{a} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{a} - \boldsymbol{\mu}) \}, \quad \boldsymbol{a} \in \mathbb{R}^D$$

Or

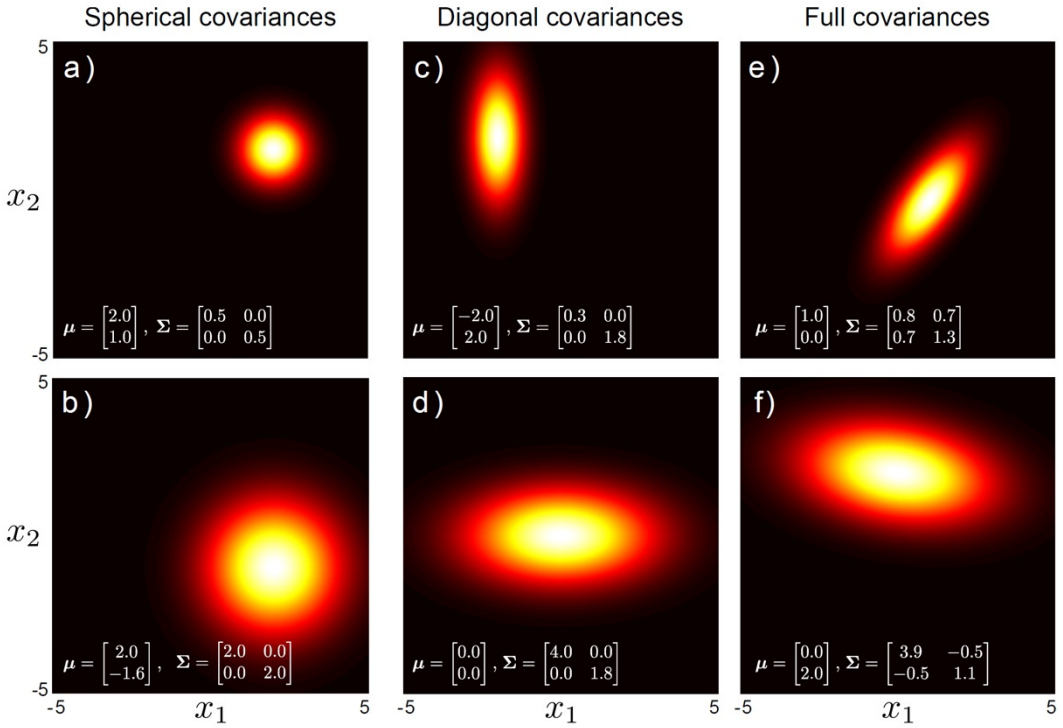
$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\{ -0.5(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \}$$

$$p(\boldsymbol{x}) = \mathrm{Norm}_{\boldsymbol{x}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}]$$

Types of Covariance

- Covariance matrix has three forms: spherical, diagonal and full.

$$\Sigma_{spher} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \quad \Sigma_{diag} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \quad \Sigma_{full} = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 \end{bmatrix}$$

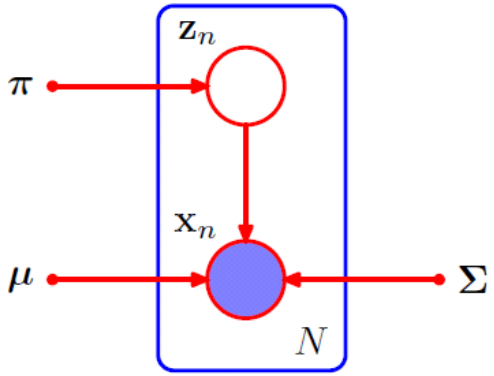


Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

Gaussian Mixture Models

- The probability distribution of a GMM 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.

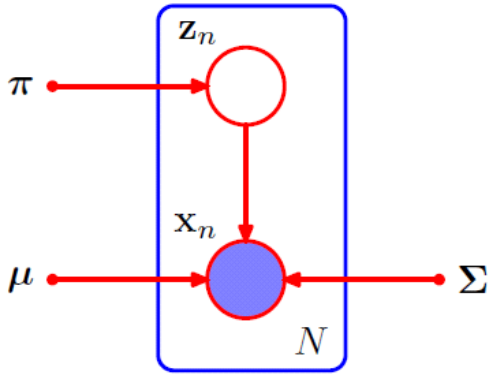
- The parameters $0 \leq \pi_k \leq 1$ is the mixing coefficients, and must sum to one:

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

GMM: Cluster Assignments

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

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



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

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

From Lecture 1 (Appendix): Categorical Distribution

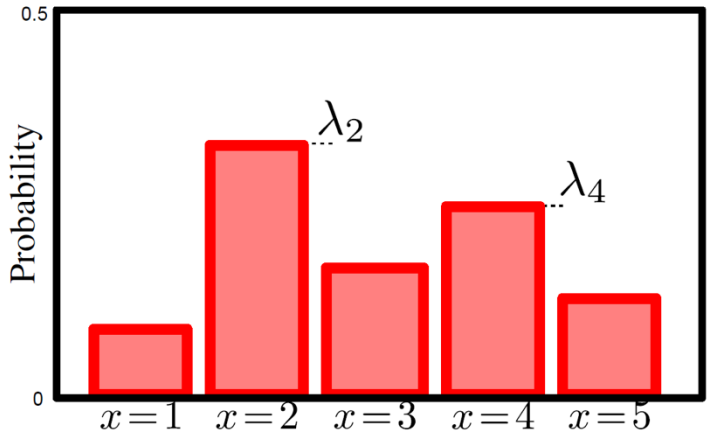
- Discrete variables $\boldsymbol{X}$ that take on 1-of-*K* possible mutually exclusive states, e.g. a *K*-faced die.
- $\boldsymbol{x}$ is represented by a *K*-dimensional vector $\mathbf{e}_k$ in which one of the elements $x_k = 1$, and $\sum_{k=1}^{K} x_k = 1$.
- e.g. $K = 5$, and $\boldsymbol{x} = \mathbf{e}_3 = [0,0,1,0,0]^T$.
- *K* parameters $\lambda = [\lambda_1, \ldots, \lambda_K]^T$, where $\lambda \geq 0,\ \sum_k \lambda_k = 1$.

$$p(\boldsymbol{X} = \mathbf{e}_k \mid \lambda) = \lambda_k$$

Or

$$p(\boldsymbol{x}) = \prod_{k=1}^{K} \lambda_k^{x_k} = \lambda_k,$$

$$p(\boldsymbol{x}) = \mathrm{Cat}_x[\lambda]$$



Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

GMM: Cluster Assignments

- 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: Conditional Distribution

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

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

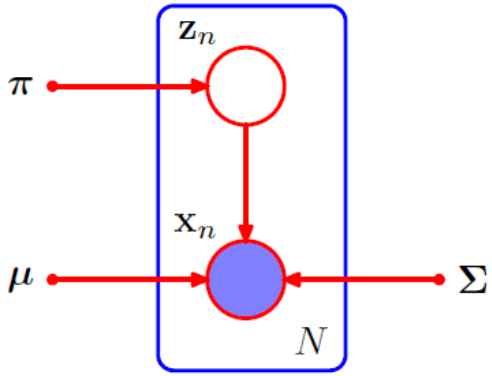
- Which can also be written as:

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

Gaussian Mixture Models

- The probability distribution of a GMM 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.

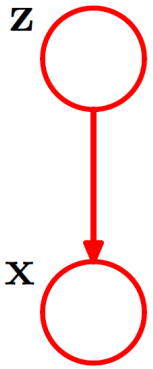
- The parameters $0 \leq \pi_k \leq 1$ is the mixing coefficients, and must sum to one:

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

Gaussian Mixture Models

- 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:

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

$p(x, z)$

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

Gaussian Mixture Models

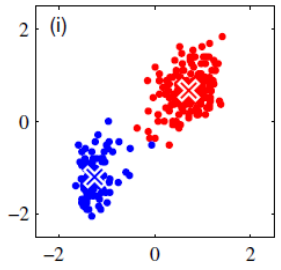
$$p(\mathrm{x}) = \sum_{\mathrm{z}} p(\mathrm{z}) p(\mathrm{x}|\mathrm{z})$$

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

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

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

Some Problems with kMeans



- Have to choose K (number of clusters)
- Deterministic assignments ("Hard Labels")
- Spherical clusters

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

K-means

- Equal sized clusters
- Sensitive to outliers

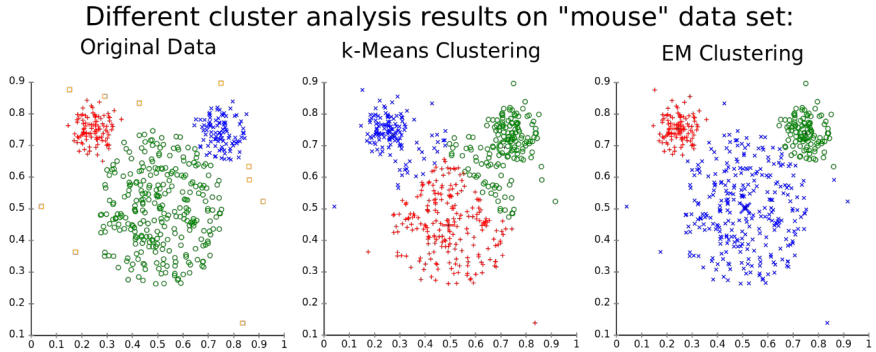


Image from: https://en.wikipedia.org/wiki/K-means_clustering
https://towardsdatascience.com/the-anatomy-of-k-means-c22340543397

GMMs: Soft Assignments

- "Responsibility" $\gamma(z_k) = p(z_k = 1|\, \mathrm{x})$.
- Via Bayes rule

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

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

- $\gamma(z_k)$ is viewed as the responsibility that component k takes for 'explaining' the observation x.
  - It will be important in the EM algorithm for learning.

Let's try it out.

https://github.com/crslab/CS5340-notebooks

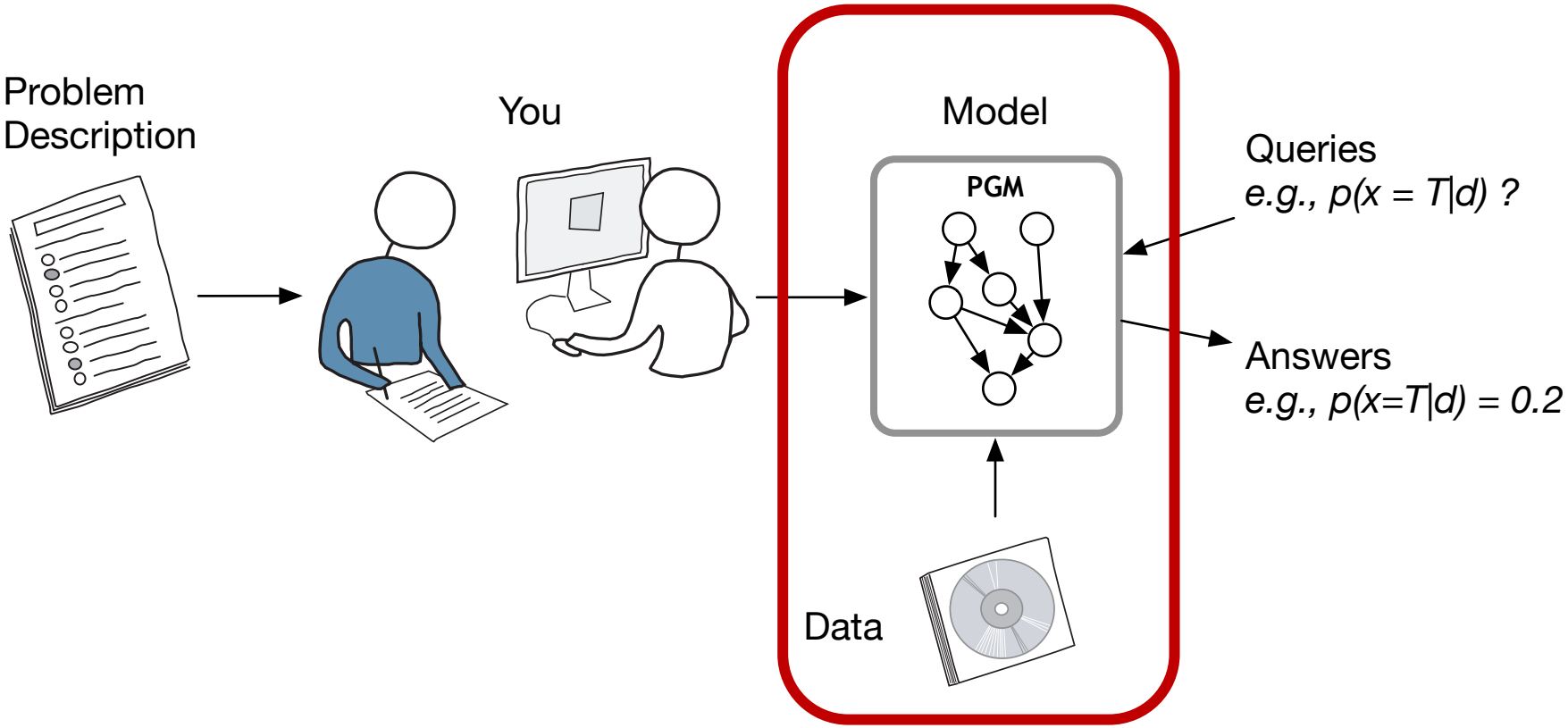
NUS National University of Singapore | School of Computing

Learning GMMs

Expectation Maximization for GMMs

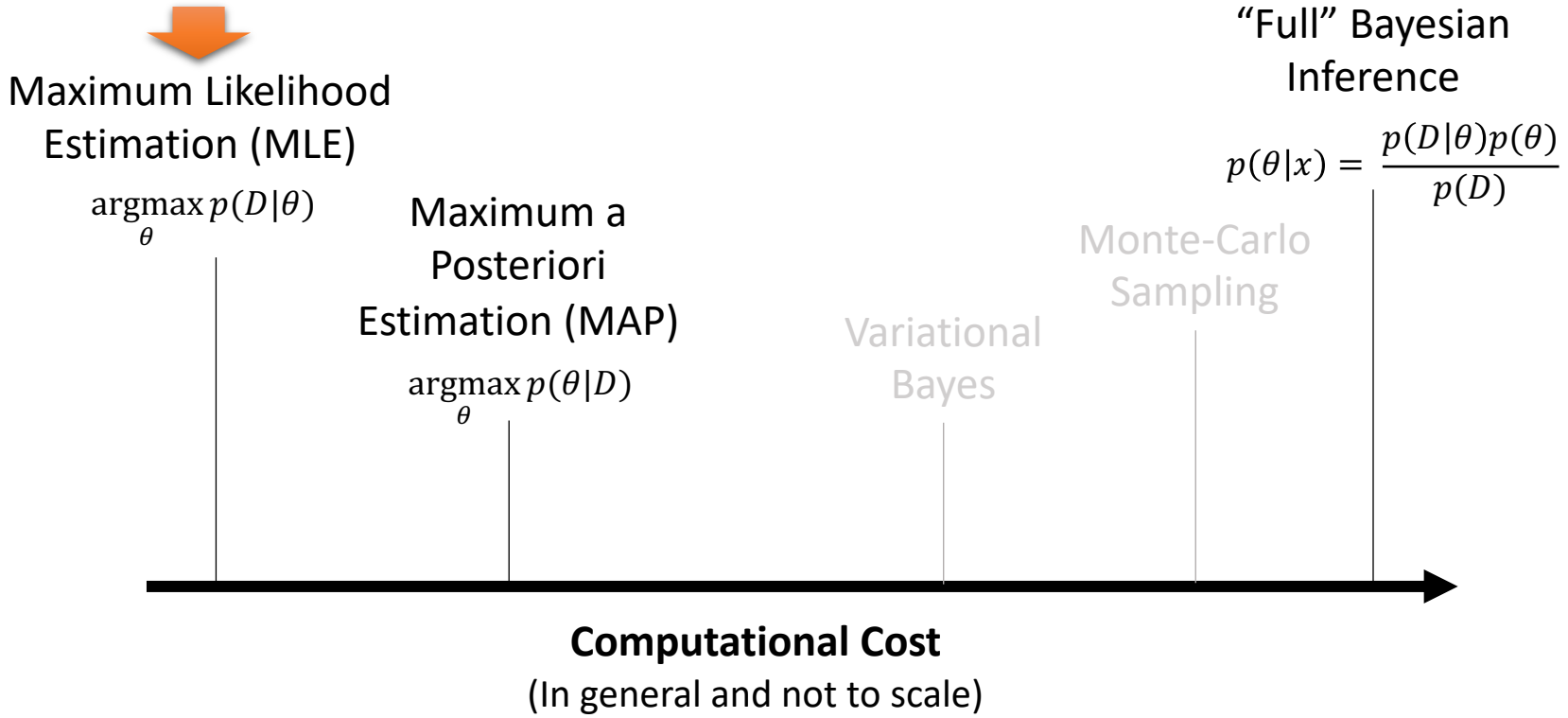
CS5340 in a nutshell

CS5340 is about how to "**represent**" and "**reason**" with uncertainty in a computer.



Learning Parameters

- Common approaches to learn the unknown parameters θ from a set of given data $\mathcal{D} = \{x[1], \ldots, x[N]\}$:



Key Ideas

- Maximum Likelihood Estimation (MLE) for GMMs does not have a closed form solution

Remember from Lecture 2: Univariate Normal Distribution

Approach 1: Maximum Likelihood Estimation (MLE)

$$\hat{\mu}, \hat{\sigma}^2 = \underset{\mu,\sigma^2}{\operatorname{argmax}} \sum_{i=1}^{N} \log\left[\operatorname{Norm}_{x[i]}[\mu, \sigma^2]\right]$$

$$= \underset{\mu,\sigma^2}{\operatorname{argmax}} \underbrace{\left[-0.5N\log\left[2\pi\right] - 0.5N\log\sigma^2 - 0.5\sum_{i=1}^{N}\frac{(x[i]-\mu)^2}{\sigma^2}\right]}_{L}$$

Maximization can be done in closed-form by taking derivative w.r.t. the variable and equate to zero:

$$\frac{\partial L}{\partial \mu} = \sum_{i=1}^{N}\frac{(x[i]-\mu)}{\sigma^2} = \frac{\sum_{i=1}^{N} x[i]}{\sigma^2} - \frac{N\mu}{\sigma^2} = 0, \qquad \frac{\partial L}{\partial \sigma^2} = -\frac{N}{\sigma^2} + \sum_{i=1}^{N}\frac{(x[i]-\mu)^2}{\sigma^4} = 0$$

$$\Rightarrow \quad \hat{\mu} = \frac{\sum_{i=1}^{N} x[i]}{N} = \bar{x}, \qquad \Rightarrow \quad \hat{\sigma}^2 = \frac{\sum_{i=1}^{N}(x[i]-\mu)^2}{N}$$

Key Ideas

- Maximum Likelihood Estimation (MLE) for GMMs does not have a closed form solution
- Derive iterative solution
  - Expectation Maximization (EM) for GMM
  - Prelude to the General EM “recipe”

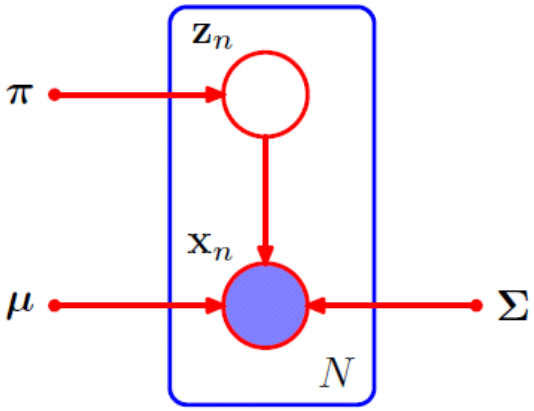
Maximum Log-Likelihood

- Data set of N i.i.d. observations $\{\mathrm{x}_1, \ldots, \mathrm{x}_N\}: \mathrm{x}_n \in \mathbb{R}^D$, the log-likelihood is:

Unknown parameter θ

$$\ln p(\mathrm{x}_1, \ldots \mathrm{x}_N | \theta) = \sum_{n=1}^{N} \ln \sum_{\mathrm{z}_n} p(\mathrm{x}_n, \mathrm{z}_n | \theta)$$

Incomplete data because $\mathrm{Z} = \{\mathrm{z}_1, \ldots, \mathrm{z}_n\} \in \mathbb{R}^{N \times K}$ is a latent variable



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

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

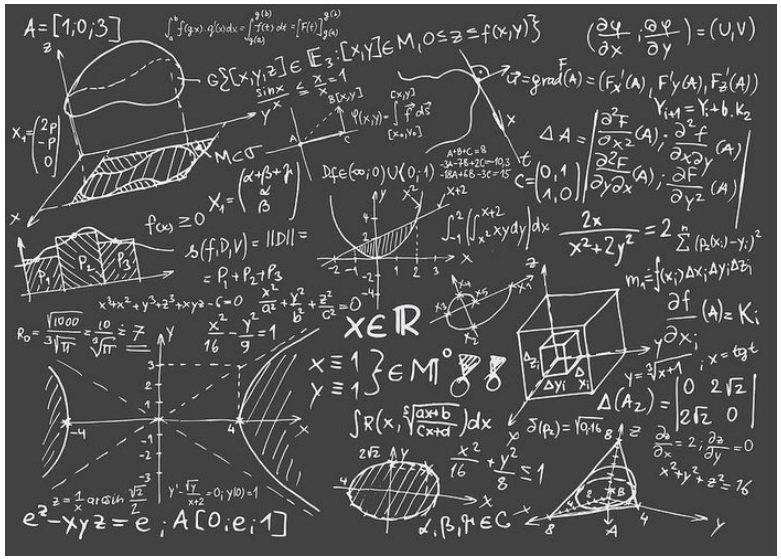
where

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

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

To the Blackboard!

- We only see $\{x_1, \ldots, x_N\}$
- Let's start with an intuitive approach
- Derivation on "blackboard"



Maximum Log-Likelihood

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

- The summation term inside the logarithm prevents the logarithm function from acting directly on the Gaussian.
- no longer 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:

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

$\gamma(z_{nk})$: Responsibility

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

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

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:

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

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

Maximum Log-Likelihood

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

Lagrange multiplier

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

- 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:

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

Maximum Log-Likelihood

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

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

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

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 $\gamma(z_{nk})$.

- Iterative scheme for finding a solution to the maximum likelihood problem.

Iterative Learning for GMMs

Given a Gaussian mixture model, 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
3. Maximization Step: Re-estimate the parameters θ using the current responsibilities
4. Evaluate the log likelihood and check for convergence

Iterative Learning for GMMs

Given a Gaussian mixture model, 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 $\gamma(z_{nk})$

Iterative Learning for GMMs

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

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

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

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

where

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

Iterative Learning for GMMs

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

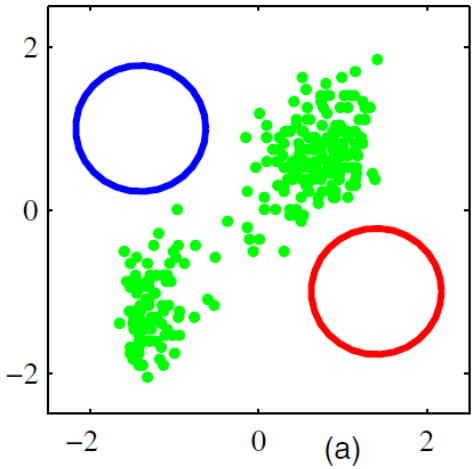
4. Evaluate the log likelihood:

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

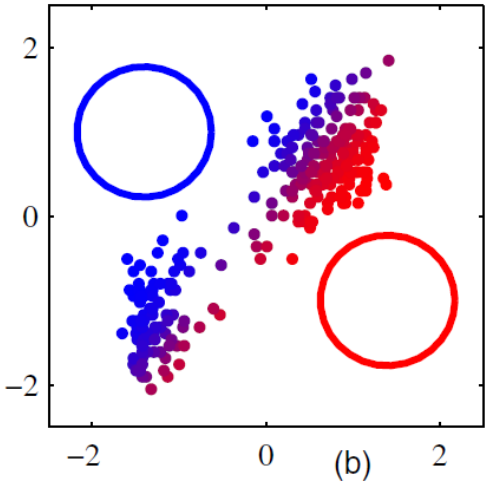
and check for convergence of either the parameters or the log likelihood. If the convergence criterion is NOT satisfied return to step 2.

Illustration of the EM Algorithm

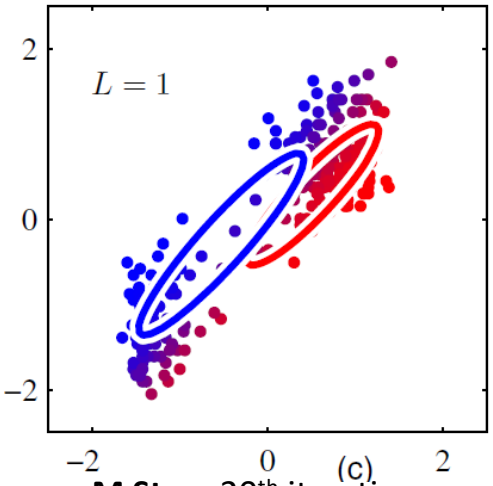
Initialization: random μ_k, identity Σ_k and $\pi_k = 0.5$



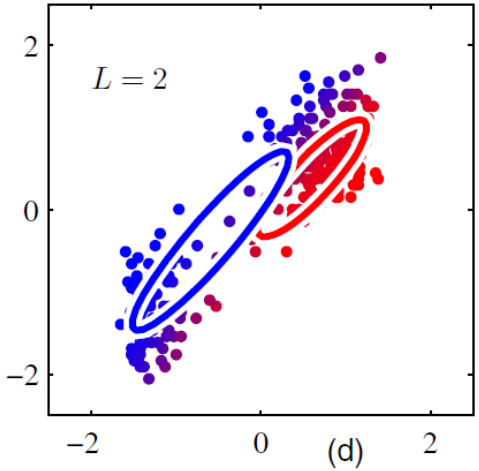
E Step: 1st iteration (color of dots used to illustrate strength of mixing coefficients)



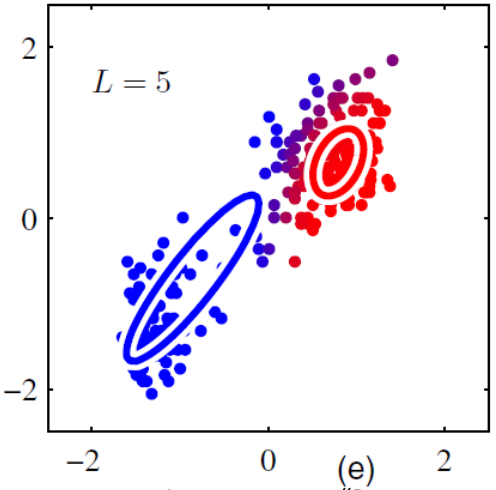
M Step: 1st iteration



M Step: 2nd iteration



M Step: 5th iteration



M Step: 20th iteration

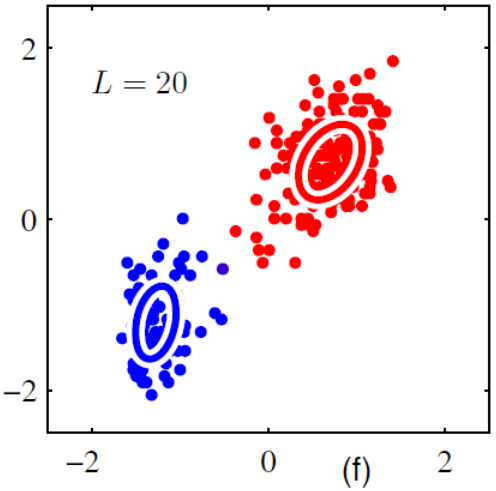


Image source: "Pattern recognition and machine learning", Christopher Bishop

Expectation Maximization for GMMs

Given a Gaussian mixture model, 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
3. Maximization Step: Re-estimate the parameters θ using the current responsibilities
4. Evaluate the log likelihood and check for convergence

EM for GMM: Initialization

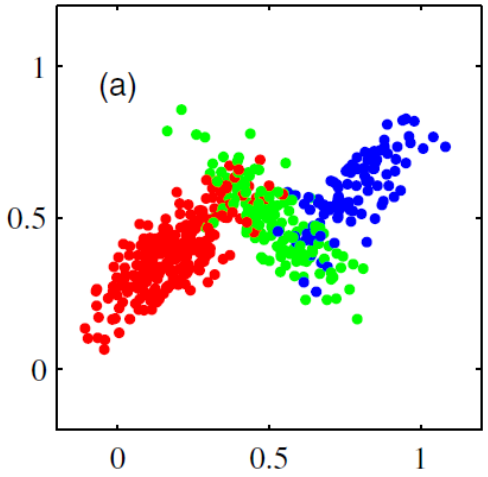
- Usually many more iterations to reach convergence compared with the *K*-means algorithm.
- And each cycle requires significantly more computation.
- In practice, can run the *K*-means algorithm first to find a suitable initialization for a GMM that is subsequently adapted using EM.

NUS National University of Singapore | School of Computing

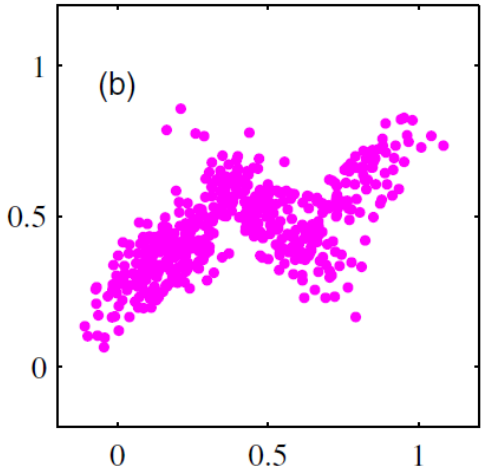
The General EM Algorithm

A "Recipe" for Learning with Missing Data

Missing Data



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

- **Goal:** 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

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

- **Key observation:** 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.
- Results in complicated (non closed-form) expressions MLE.

The General EM Algorithm

- 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 X.
- **Idea:** maximize 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

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:

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

where

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

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

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

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:

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

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

Log is now inside the summation!

- 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

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:

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

where

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

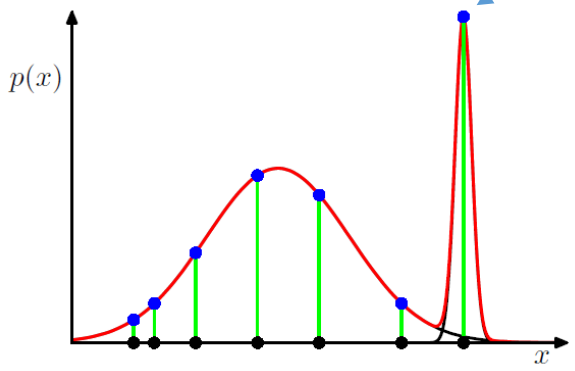
4. Check for convergence of either the log likelihood or the parameter values, if not converged:

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

Tutorial

- Apply General EM Recipe to GMMs!
- Extra: Singularities and MAP

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



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

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

Prior over parameters

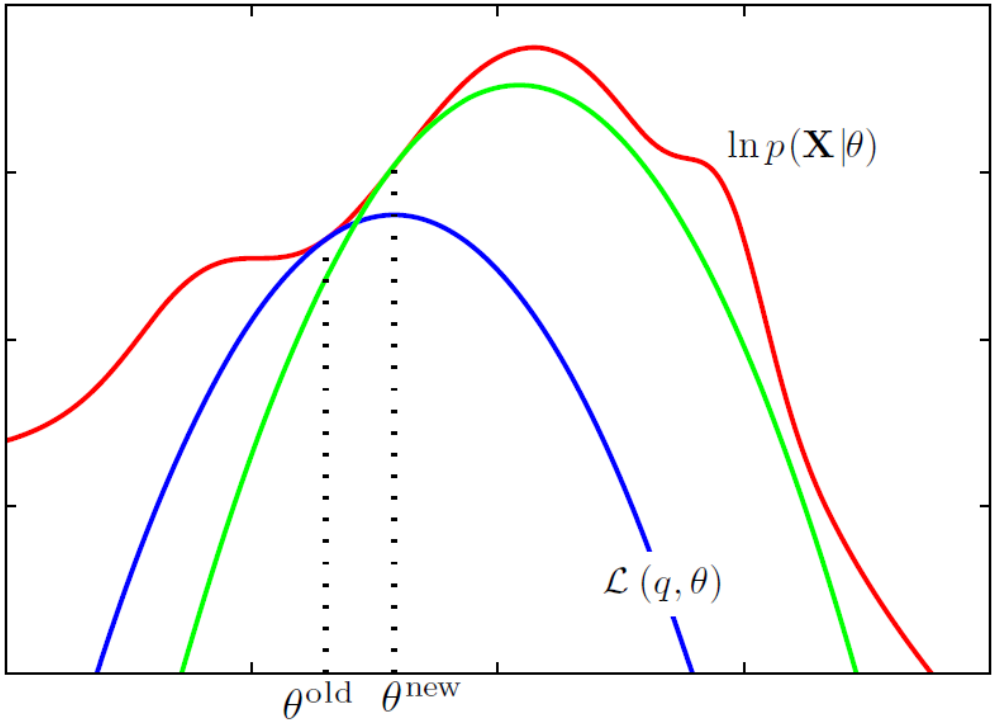
NUS National University of Singapore | School of Computing

EM Derivation

Why does EM work?

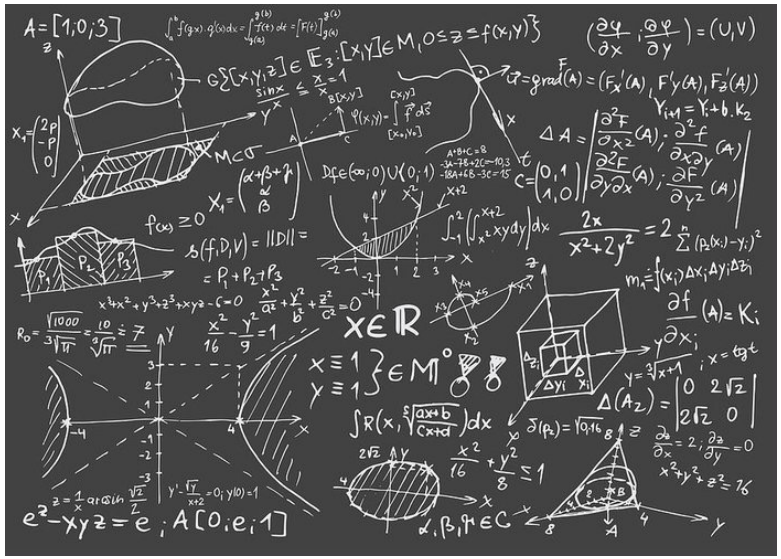
Key Ideas

- How did we derive the (general) EM recipe?
- Why does EM work?



To the Blackboard!

- Instead of maximizing $\log p(x|\theta)$, we will optimize a lower bound $\mathcal{L}(q, \theta)$
- Derivation on "blackboard"



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

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

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

$q(Z)$ is a distribution we defined over the latent variable

The Theory Behind EM Algorithm

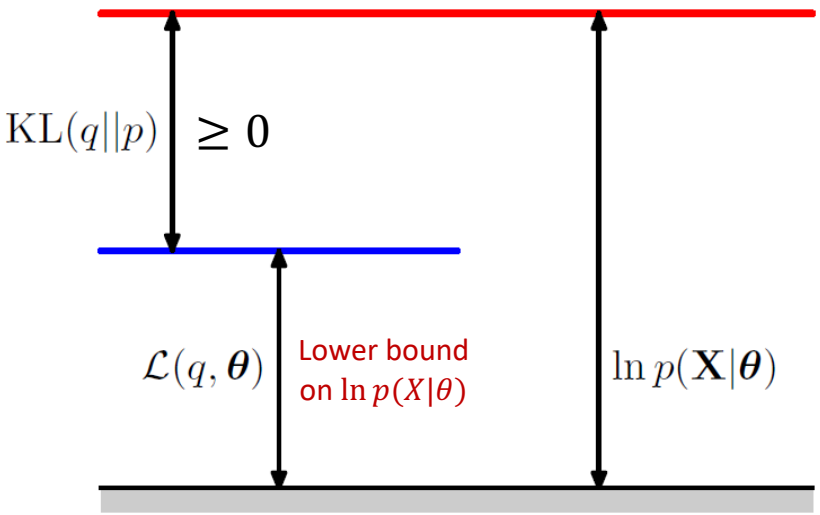


Illustration of the decomposition given by:

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

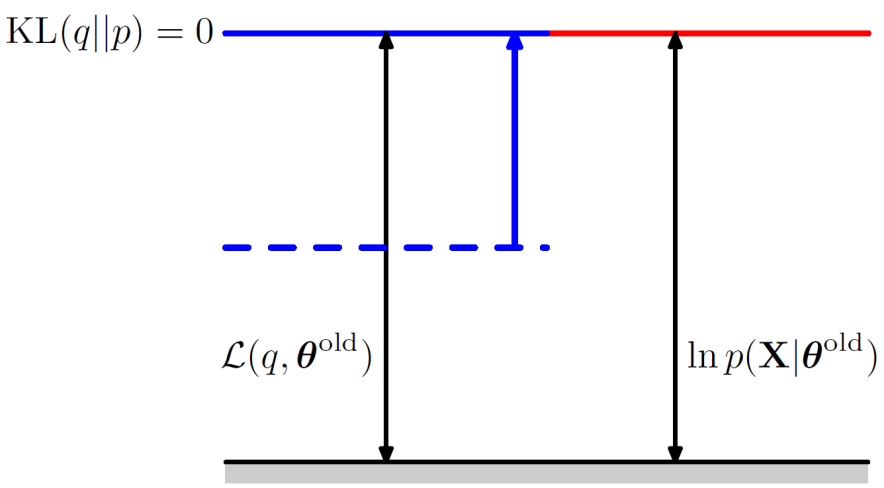
which holds for any choice of $q(\boldsymbol{Z})$.

- Because the Kullback-Leibler divergence satisfies $\mathrm{KL}(q||p) \geq 0$, 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:



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

Proof:

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

($p(Z|X,\theta)$ cancels to 0)

$$= 0$$

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

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

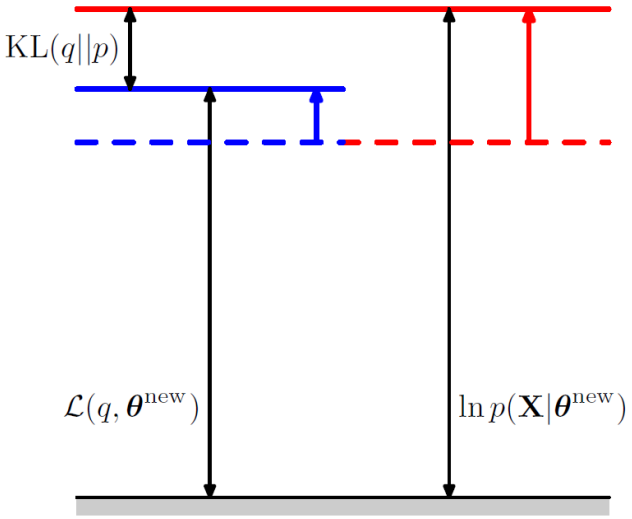
- Equivalence of expectation under the latent variable distribution:

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

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

The Theory Behind EM Algorithm

Illustration of the M-Step:



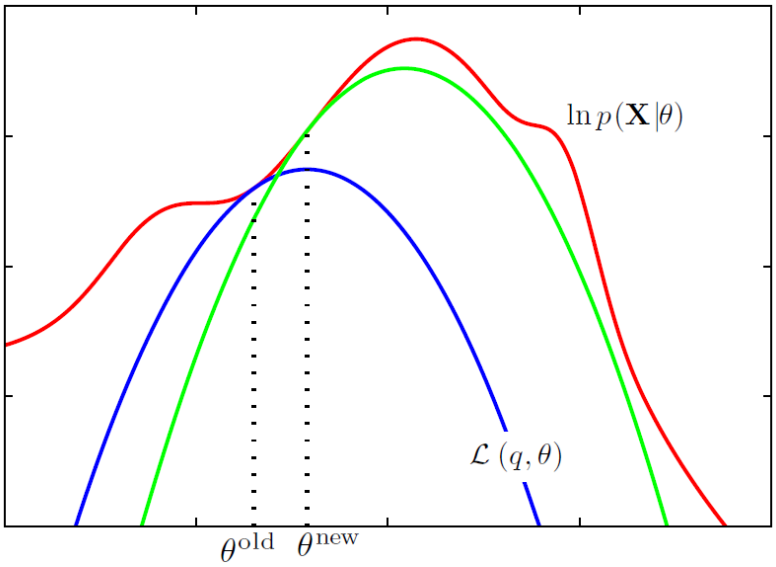
$$\mathcal{L}(q, \theta^{new}) = \sum_{Z} p\left(Z\middle|X, \theta^{old}\right) \ln p(X, Z|\theta^{new}) + \text{const}$$

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

- 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

Learning Outcomes

- Students should be able to:

1. Use the non-probabilistic k-means 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.

4. Explain why the EM algorithm works.