# Mathematics of Big Data, I

# Lecture 7: Unsupervised Learning, K-mean Clustering, Gaussian Mixture, Jensen's inequality, and EM

**Weiqing Gu**
Professor of Mathematics
Director of the Mathematics Clinic

Harvey Mudd College
Summer 2017

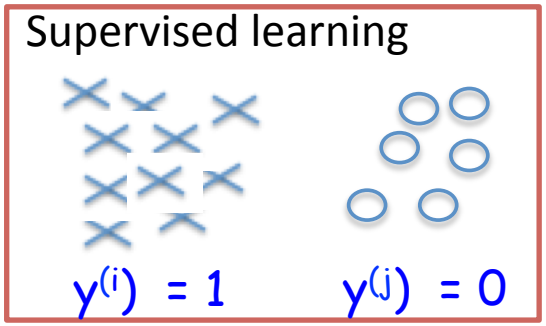
# Today's topics

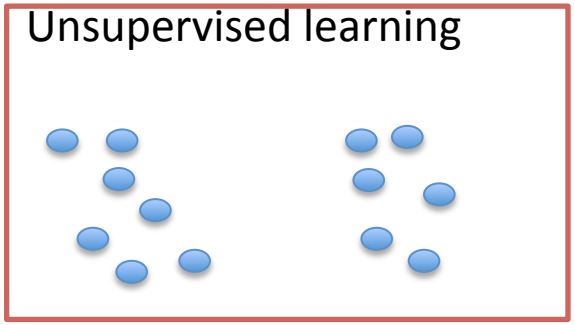
- The k-means clustering algorithm
- Mixtures of Gaussians
- Jensen's inequality
- The EM (Expectation-Maximization) Algorithm

# What is a clustering problem?

A clustering problem is an unsupervised learning problem

- Given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$, here each $x^{(i)}$ is in $\mathbf{R}^n$.
- Goal: want to group the data into a few cohesive "clusters."
- Note: the difference between unsupervised learning and supervised learning is that no labels y(i) are given.




In general, if only $\{x^{(1)}, \ldots, x^{(m)}\}$ given for a problem, but no labels $y^{(i)}$ are given, then the problem is an unsupervised learning problem!

# The k-means clustering algorithm

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

   For every $i$, set

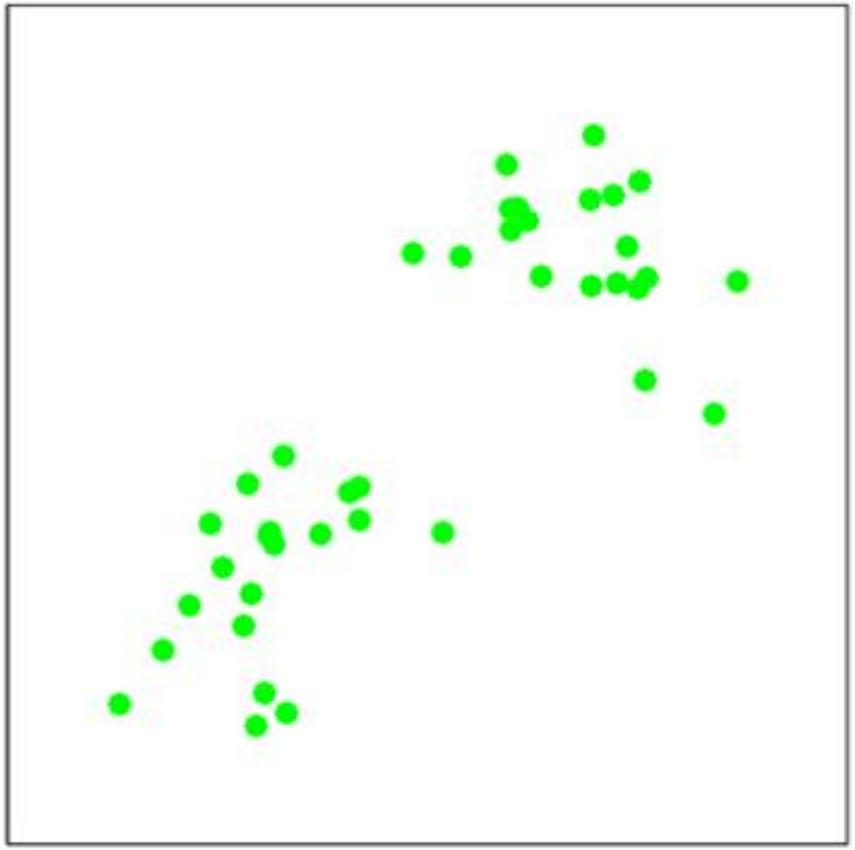
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each $j$, set

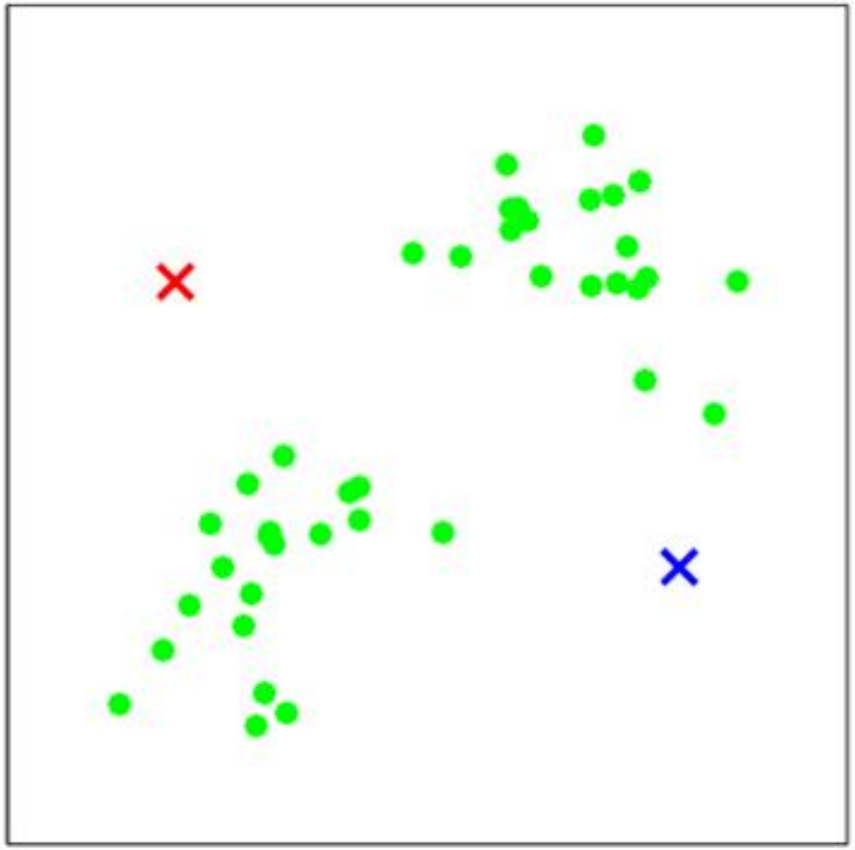
   $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

   }

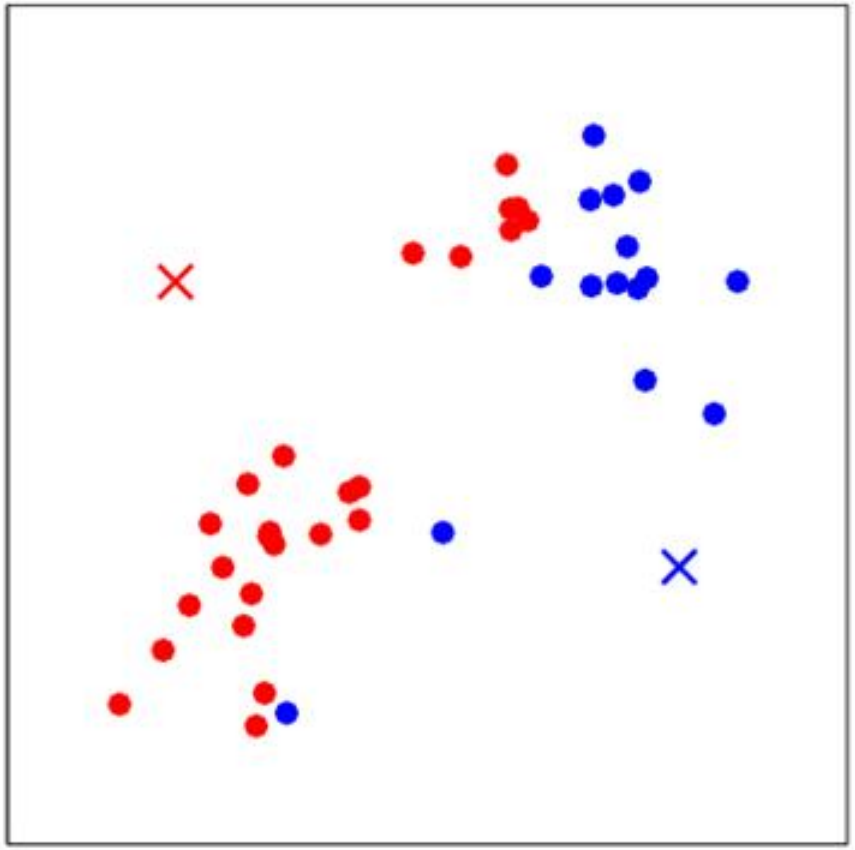
# Example of K-mean clustering



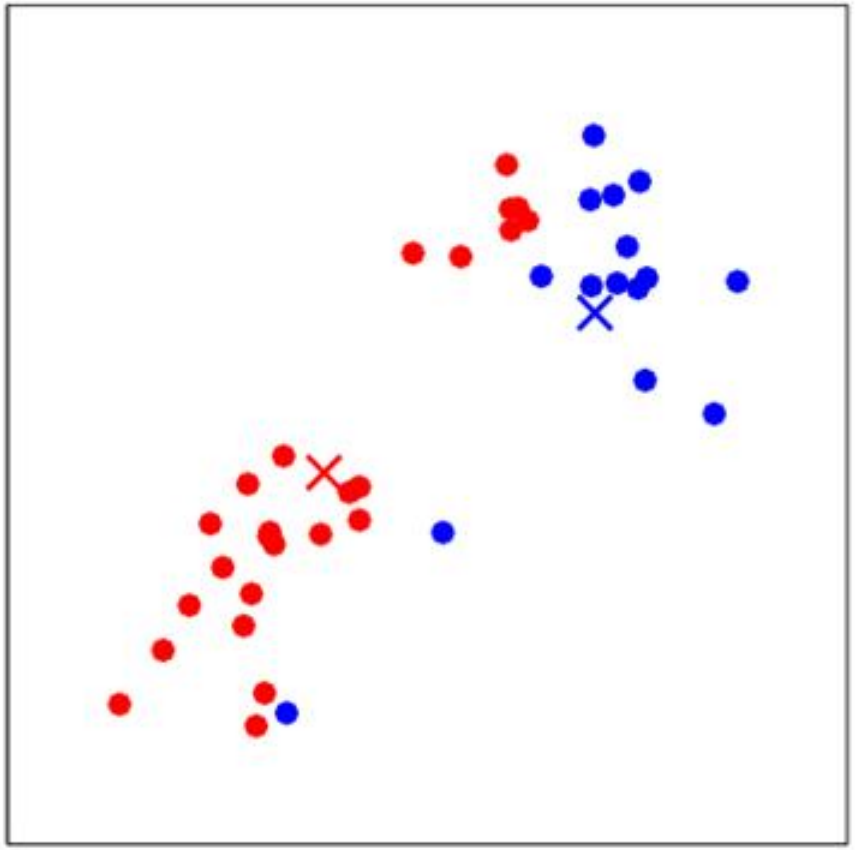
# Example of K-mean clustering



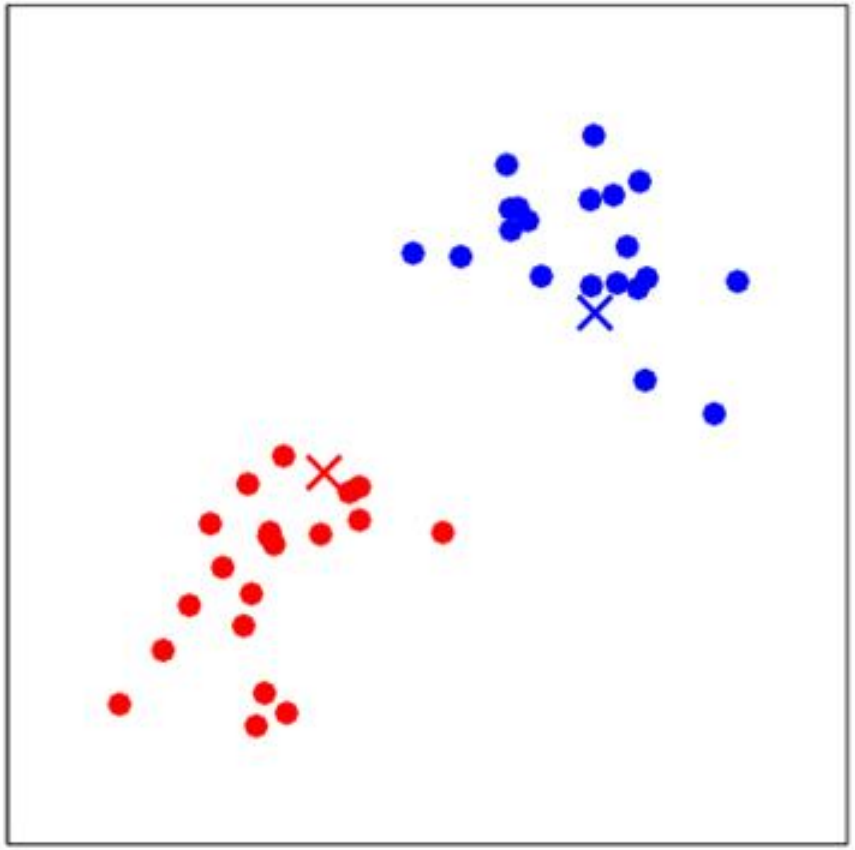
# Example of K-mean clustering



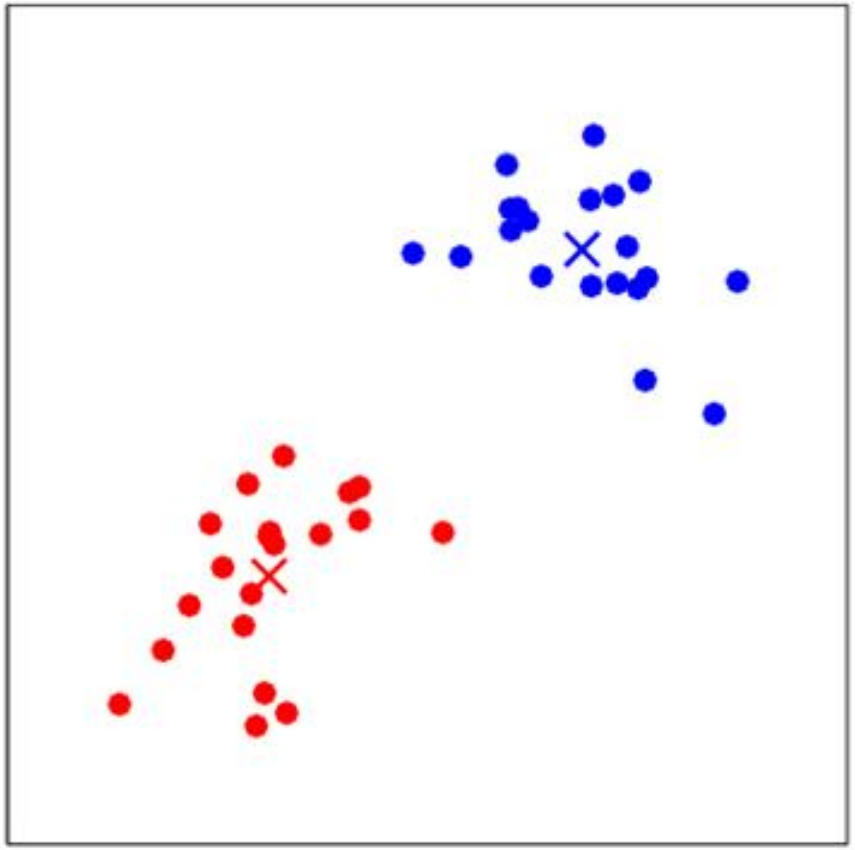
# Example of K-mean clustering



# Example of K-mean clustering



# Example of K-mean clustering



# Example of K-mean clustering

- In the Figure above for K-means algorithm: Training examples are shown as dots, and cluster centroids are shown as crosses.
- (a) Original dataset.
- (b) Random initial cluster centroids (in this instance, not chosen to be equal to two training examples).
- (c-f) Illustration of running two iterations of k-means.

- In each iteration, we assign each training example to the closest cluster centroid (shown by "painting" the training examples the same color as the cluster centroid to which is assigned); then we move each cluster centroid to the mean of the points assigned to it. (Best viewed in color.)
- Images courtesy Michael Jordan.

# Q: Is the k-means algorithm guaranteed to converge?

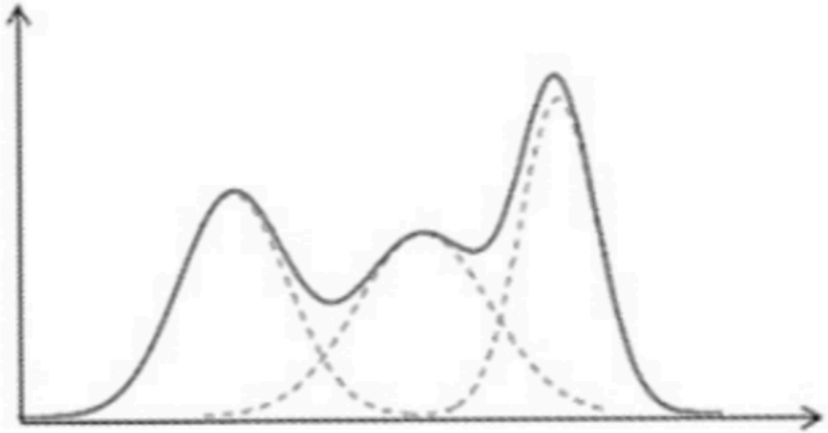
- Yes it is, but it might convergent to a local optimization point instead a global one in following sense.
- Define the distortion function to be: $J(c, \mu) = \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$
- Here J measures the sum of squared distances between each training example $x^{(i)}$ and the cluster centroid $\mu_{c(i)}$ to which it has been assigned.
- It can be shown that k-means is exactly coordinate descent on J.
- This means that the inner-loop of k-means repeatedly minimizes J with respect to c while holding μ fixed, and then minimizes J with respect to μ while holding c fixed.
- Thus, J must monotonically decrease, and the value of J must converge. (Usually, this implies that c and μ will converge too.)
- In theory, it is possible k-means to oscillate between a few different clusterings—i.e., a few different values for c and/or μ—that have exactly the same value of J, but this almost never happens in practice.)

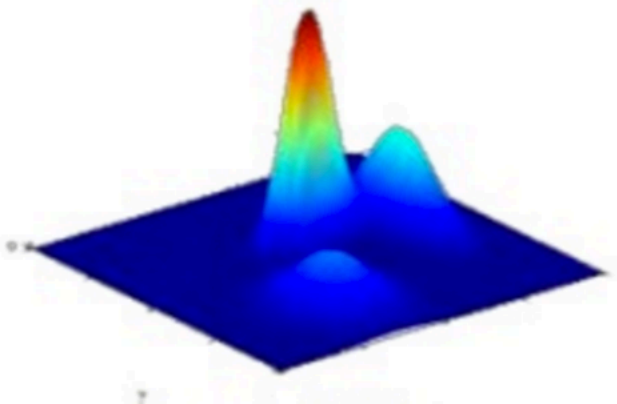
# Note: The distortion function J is non-convex, so no global minimum is guaranteed.

- That is to say the coordinate descent on J is not guaranteed to converge to the global minimum: k-means can be susceptible to local optima.
- But very often k-means will work fine and come up with very good clusterings despite this.
- Try heuristic method if you are worried about getting stuck in bad local minima:
- Run k-means many times (using different random initial values for the cluster centroids $\mu_j$).
- Then, out of all the different clusterings found, select the one that gives the lowest distortion J(c, μ).

Change gears:

# Multivariate Gaussian Mixture Model





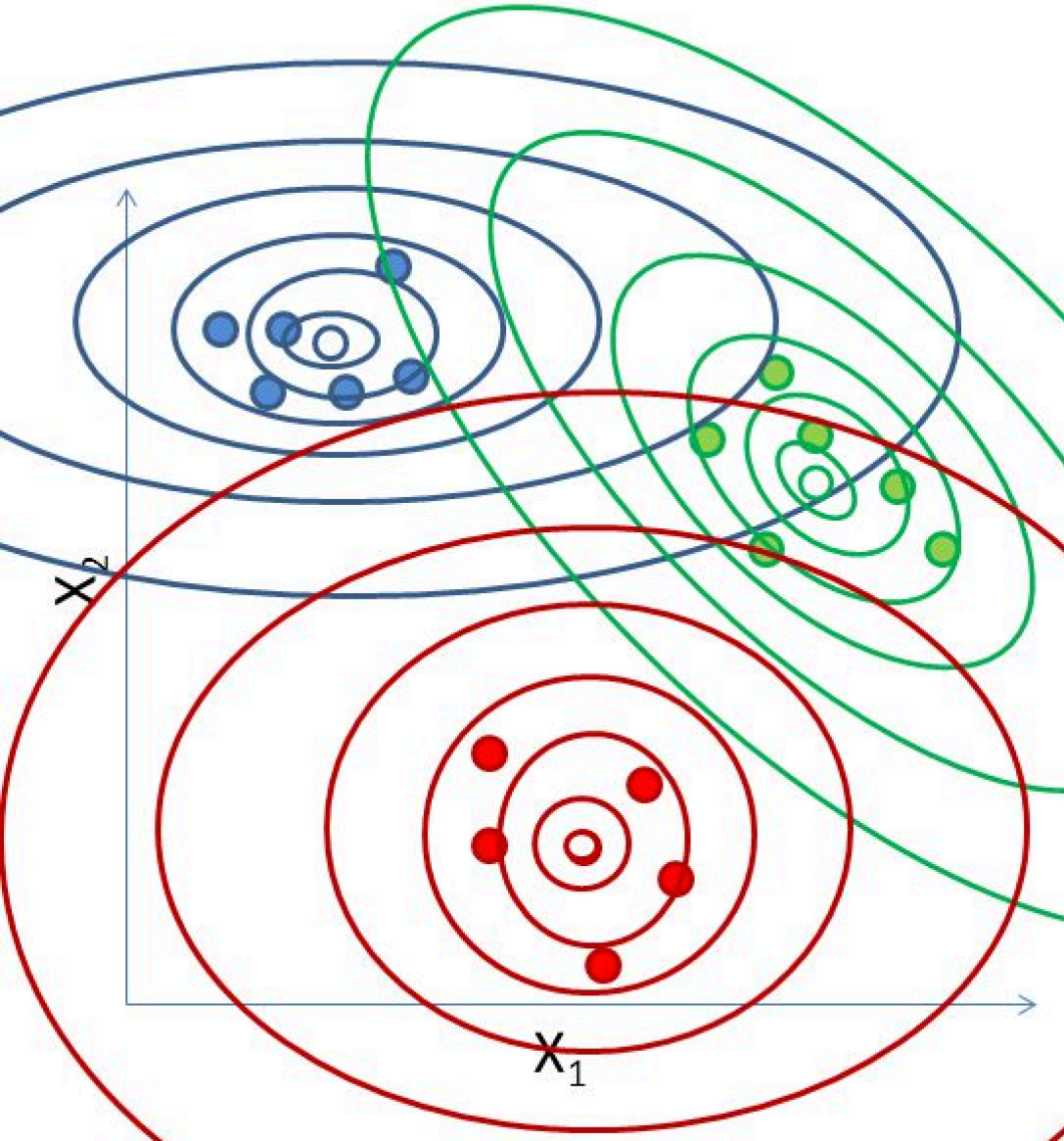
*Mimic linear combination of vectors, here is a convex combination.*

$$p(\boldsymbol{\theta}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$$

$$\phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$$

where the $i^{th}$ vector component is characterized by normal distributions weights $\phi_i$, means $\boldsymbol{\mu_i}$ and covariance matrices $\boldsymbol{\Sigma_i}$.

# Gaussian Mixture Models




Like K-Means, GMM clusters have centers.

In addition, they have probability distributions that indicate the probability that a point belongs to the cluster.

These ellipses show "level sets": lines with equal probability of belonging to the cluster.

Notice that green points still have SOME probability of belonging to the blue cluster, but it's much lower than the blue points.

This is a more complex model than K-Means: distance from the center can matter more in one direction than another.

**A soft clustering methods. Sign more responsibility to one of the Gaussians.**

# How Gaussian mixture model and Expectation-Maximization (EM) related?

**Key:** the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ is *also* a Gaussian mixture mode !

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \sum_{i=1}^{K} \tilde{\phi}_i \mathcal{N}(\tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i)$$

with new parameters $\tilde{\phi}_i$, $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\boldsymbol{\Sigma}}_i$ that are updated using the EM algorithm.

What is an EM algorithm?

EM = **Expectation-Maximization**

# The EM (Expectation-Maximization) Algorithm

- Expectation of what?
- Maximization of what?

- *Work out details with students on the board.*

# EM (Expectation-Maximization) Algorithm

- Given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$
- Note: since we are in the unsupervised learning setting, so these points do not come with any labels.
- Goal: Model the data by specifying a joint distribution $p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$.

Here, $z^{(i)} \sim \text{Multinomial}(\phi)$ $\phi_j \geq 0, \sum_{j=1}^{k} \phi_j = 1$ and the parameter $\phi_j$ gives $p(z^{(i)} = j$

and $x^{(i)}|z^{(i)} = j \sim \mathcal{N}(\mu_j, \Sigma_j)$

$k$ denote the number of values that the $z^{(i)}$'s can take on.

# Work out details with students on the board

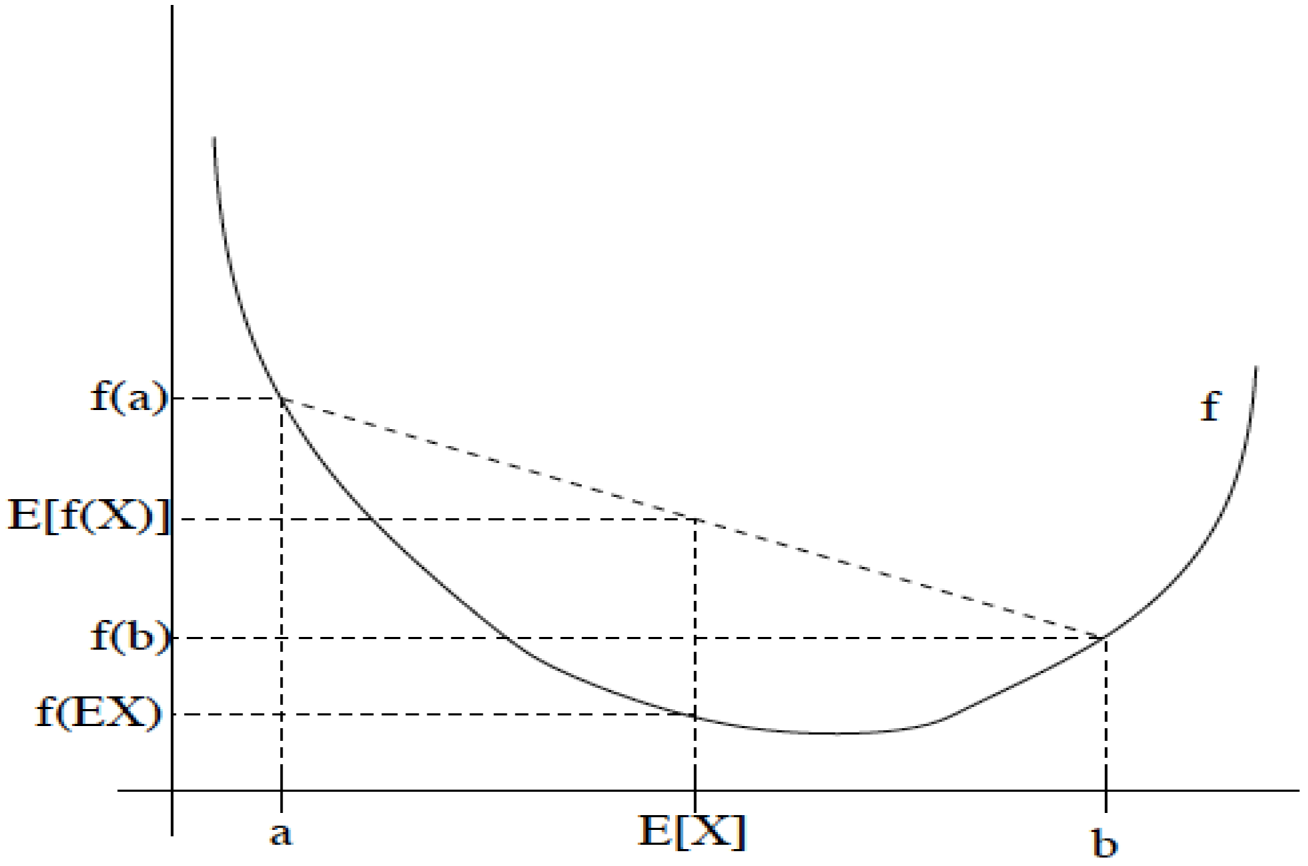
- The parameters of our model are thus $\phi$, $\mu$ and $\sum$ . To estimate them, write:

# Jensen's inequality

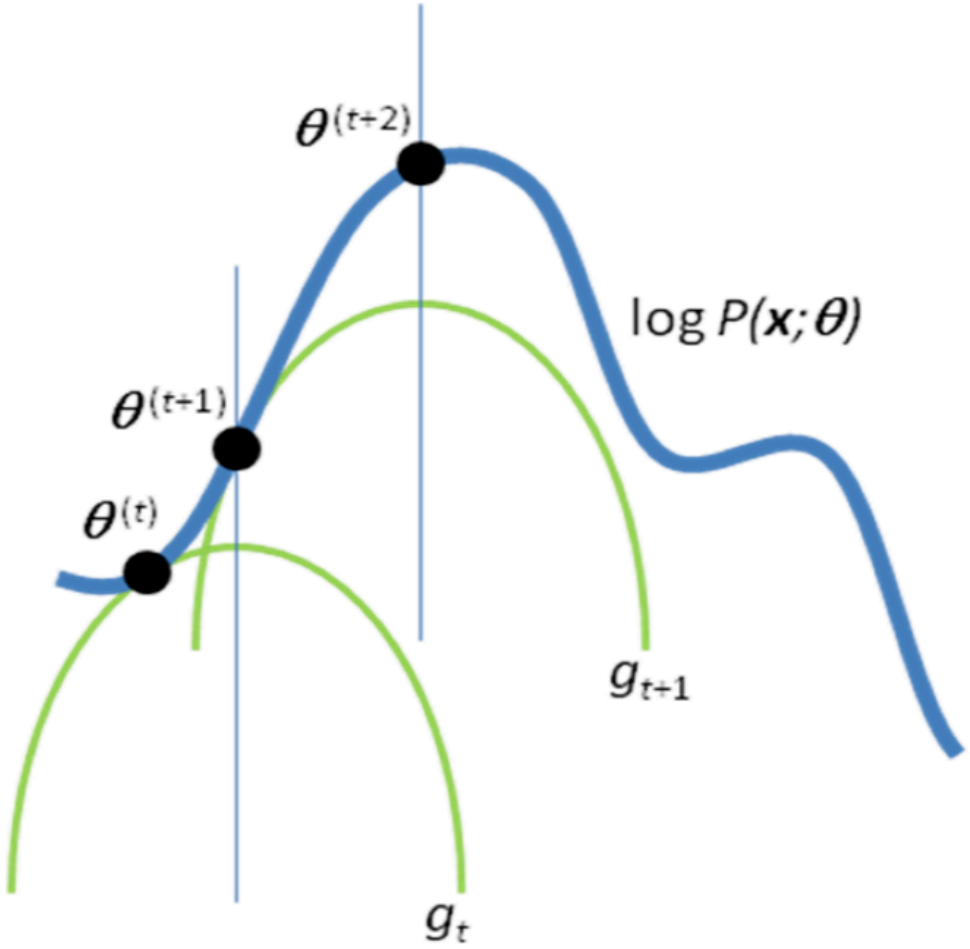
**Theorem.** Let $f$ be a convex function, and let $X$ be a random variable. Then:

$$\mathrm{E}[f(X)] \geq f(\mathrm{E}X).$$

Moreover, if $f$ is strictly convex, then $\mathrm{E}[f(X)] = f(\mathrm{E}X)$ holds true if and only if $X = \mathrm{E}[X]$ with probability 1 (i.e., if $X$ is a constant).

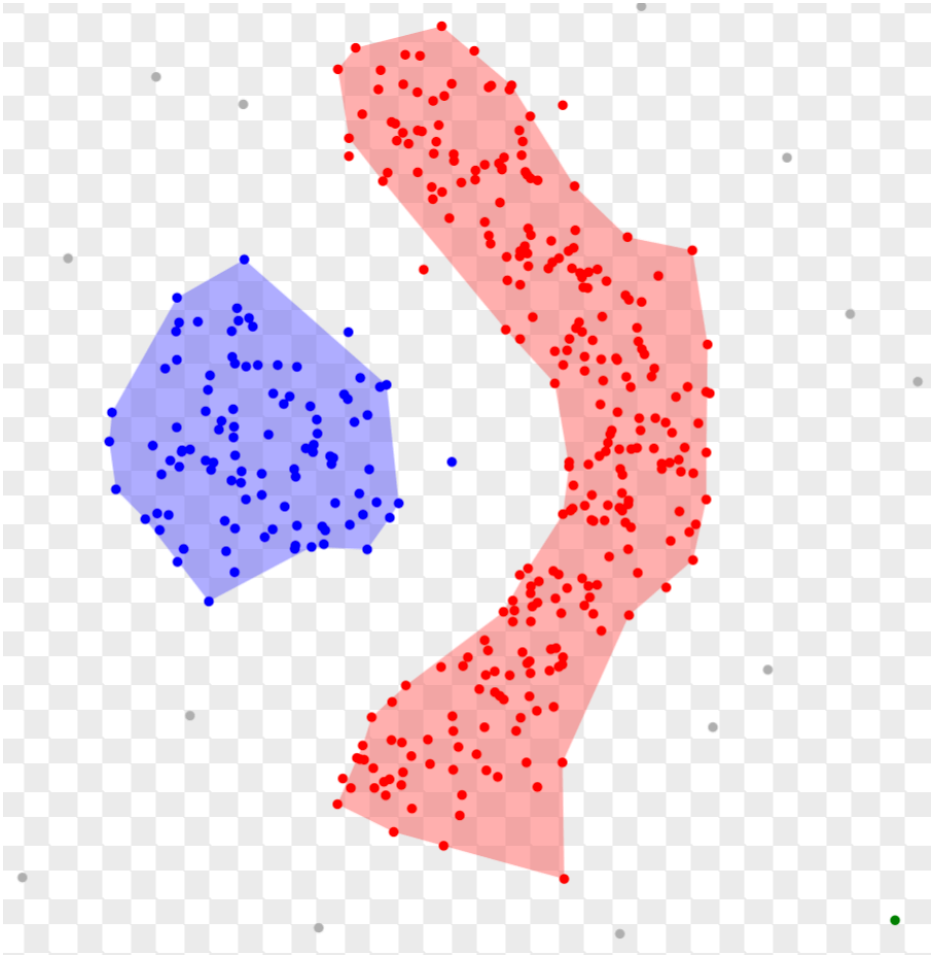


# Geometry of EM algorithm

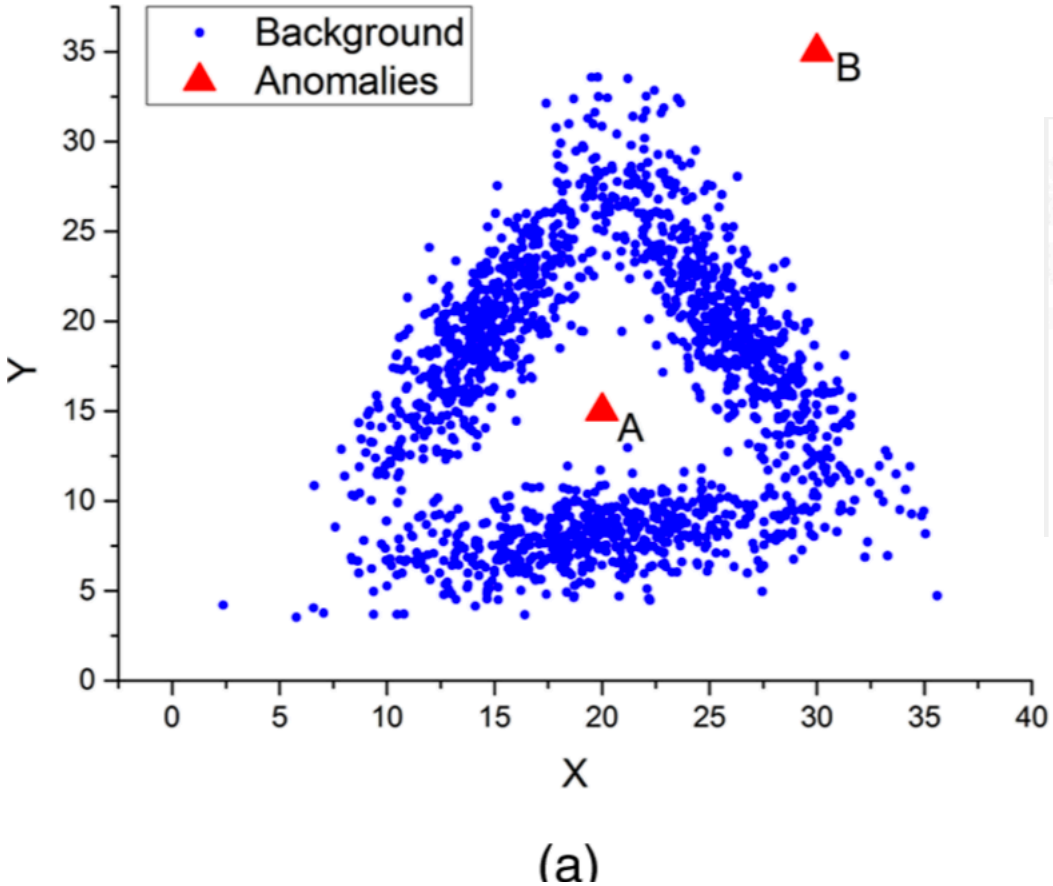


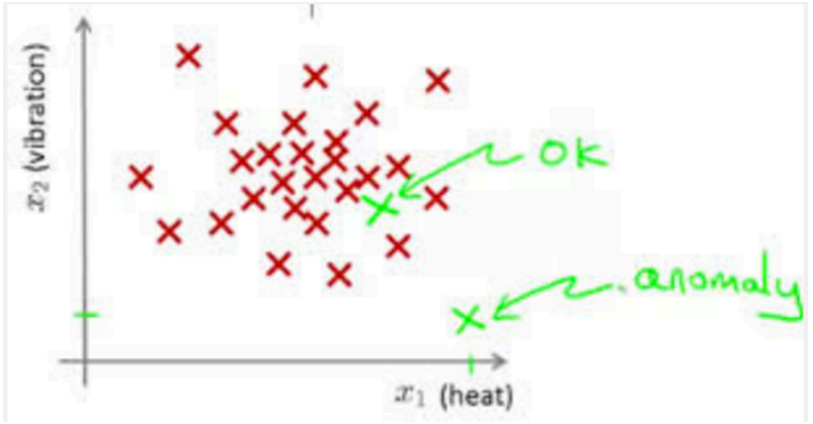

**Supplementary Figure 1** Convergence of the EM algorithm. Starting from initial parameters $\theta^{(t)}$, the E-step of the EM algorithm constructs a function $g_t$ that lower-bounds the objective function $\log P(x;\theta)$. In the M-step, $\theta^{(t+1)}$ is computed as the maximum of $g_t$. In the next E-step, a new lower-bound $g_{t+1}$ is constructed; maximization of $g_{t+1}$ in the next M-step gives $\theta^{(t+2)}$, etc.

# Density estimation using EM Algorithm



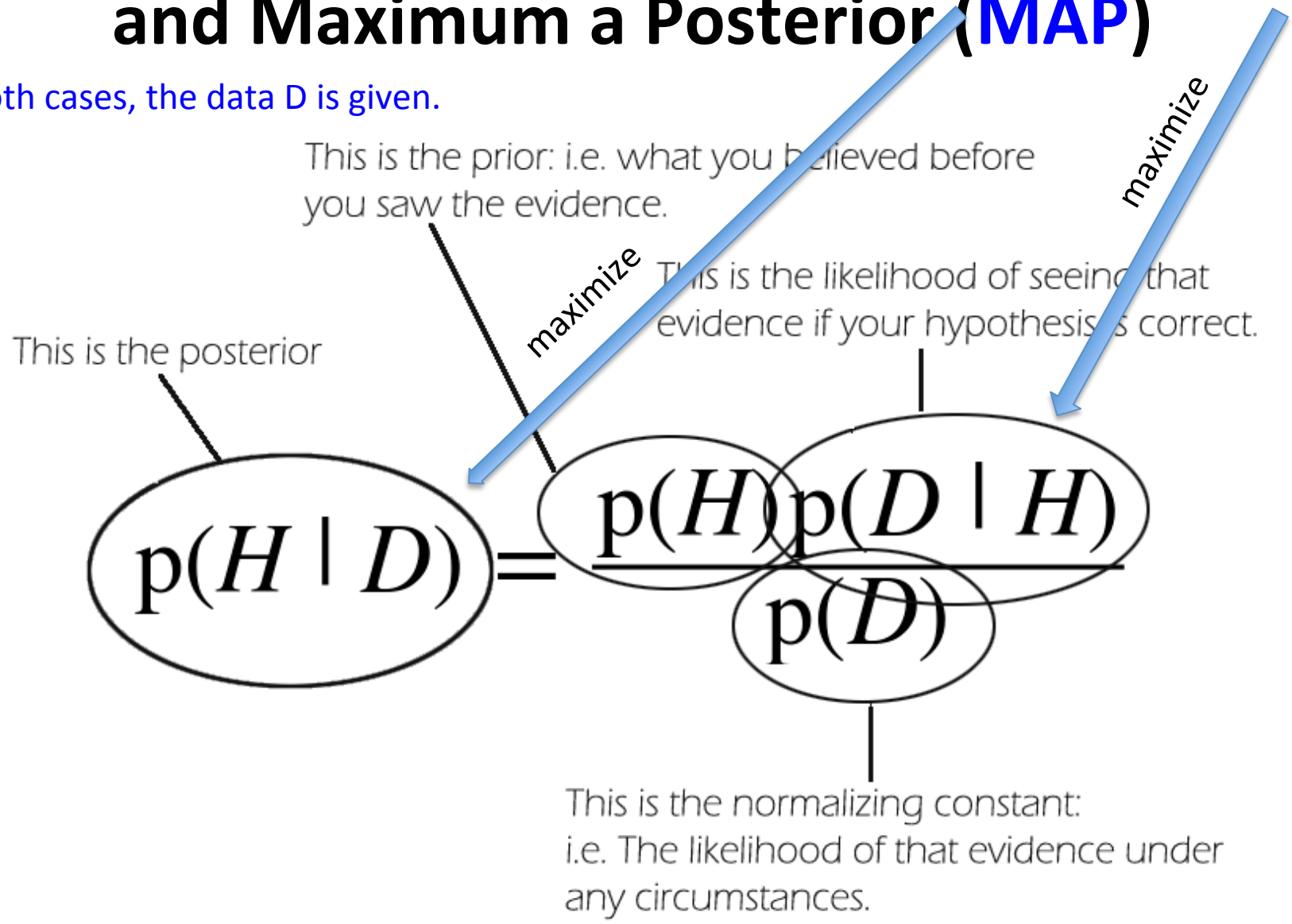
# Anomaly Detection using Density Estimation



# Recall: Maximum Likelihood Estimation (MLE) and Maximum a Posterior (MAP)

In both cases, the data D is given.



# Recall: MLE =Maximum Likelihood Estimation

- In statistics, maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model given observations, by finding the parameter values that maximize the likelihood of making the observations given the parameters.

# Recall: MLE = Maximum Likelihood Estimate

Assume that we want to estimate an unobserved population parameter $\theta$ on the basis of observations $x$. Let $f$ be the sampling distribution of $x$, so that $f(x \mid \theta)$ is the probability of $x$ when the underlying population parameter is $\theta$. Then the function:

$$\theta \mapsto f(x \mid \theta)$$

is known as the likelihood function and the estimate:

$$\hat{\theta}_{\mathrm{ML}}(x) = \arg\max_{\theta} f(x \mid \theta)$$

is the maximum likelihood estimate of $\theta$.

# Recall: MAP

- Maximum a posteriori (MAP) estimation is a model of posterior distribution.
- The MAP can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data.

Now assume that a prior distribution $g$ over $\theta$ exists. This allows us to treat $\theta$ as a random variable as in Bayesian statistics. We can calculate the posterior distribution of $\theta$ using Bayes' theorem:

$$\theta \mapsto f(\theta \mid x) = \frac{f(x \mid \theta)\, g(\theta)}{\displaystyle\int_{\vartheta \in \Theta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta}$$

where $g$ is density function of $\theta$, $\Theta$ is the domain of $g$.

The method of maximum a posteriori estimation then estimates $\theta$ as the mode of the posterior distribution of this random variable:

$$\hat{\theta}_{\mathrm{MAP}}(x) = \arg\max_{\theta} f(\theta \mid x) = \arg\max_{\theta} \frac{f(x \mid \theta)\, g(\theta)}{\displaystyle\int_{\vartheta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta} = \arg\max_{\theta} f(x \mid \theta)\, g(\theta).$$

The denominator of the posterior distribution (so-called marginal likelihood) does not depend on $\theta$