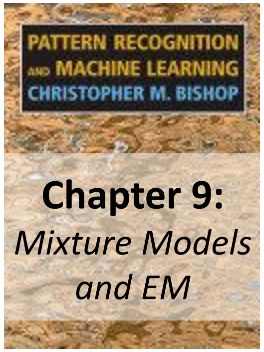
Pattern Analysis and Recognition

Lecture 7: Clustering, Expectation Maximization

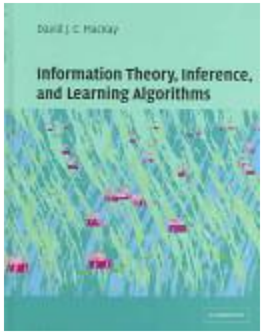
Resources

Some of the material in this slides was borrowed from:

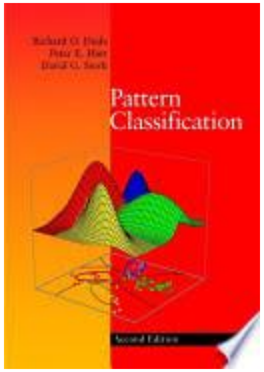
C. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006
Some related material available:
http://research.microsoft.com/en-us/um/people/cmbishop/prml/index.htm



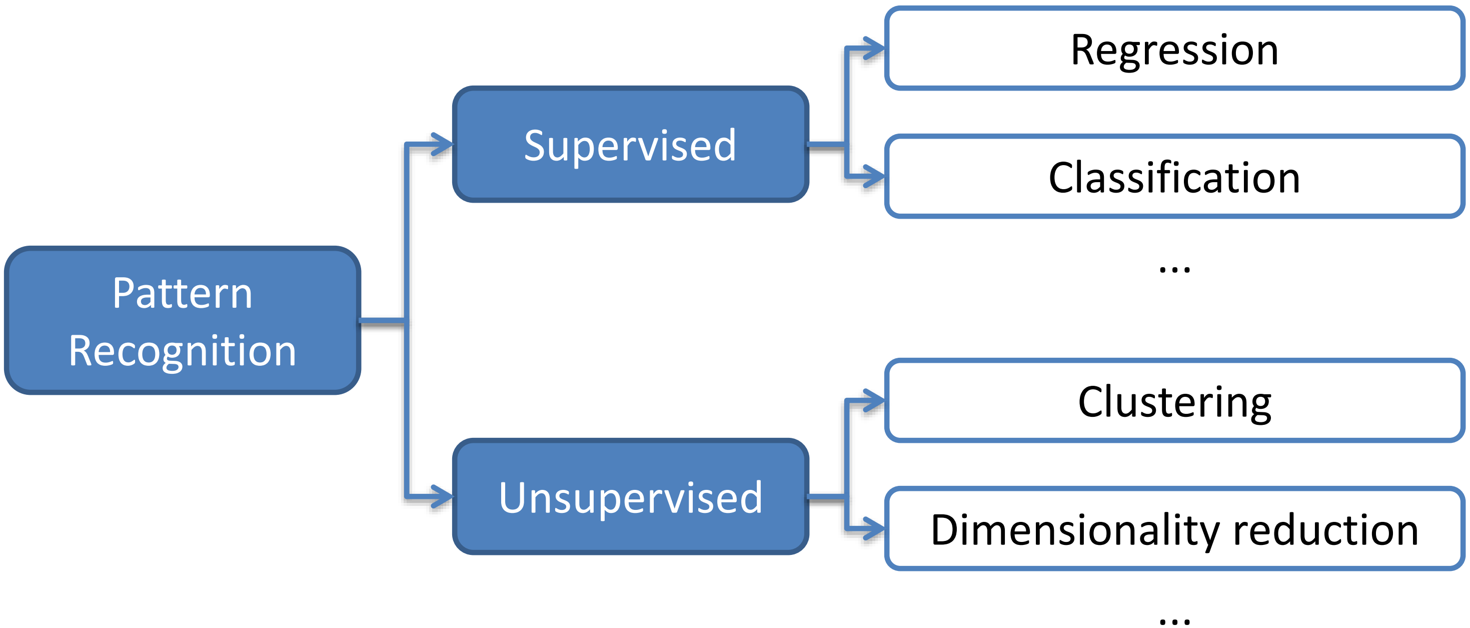
D. MacKay, “Information Theory, Inference and Learning Algorithms”, Cambridge University Press, 2003.
Book available online:
http://www.inference.phy.cam.ac.uk/mackay/



R.O. Duda, P.E. Hart, D.G. Stork, “Pattern Classification”, Wiley & Sons, 2000
Have a look inside at selected chapters:
http://books.google.es/books/about/Pattern_Classification.html?id=Br33IRC3PkQC&redir_esc=y

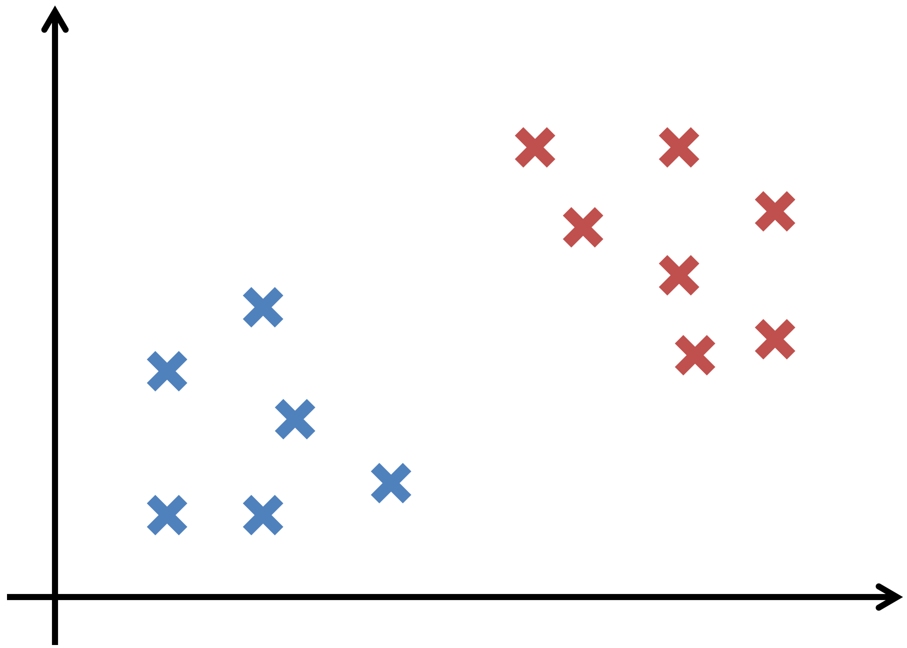


Pattern recognition algorithms



There are many pattern recognition algorithms. Specific scenarios exist, such as when samples come in a sequence. Different branches such as statistical, syntactic and structural pattern recognition…

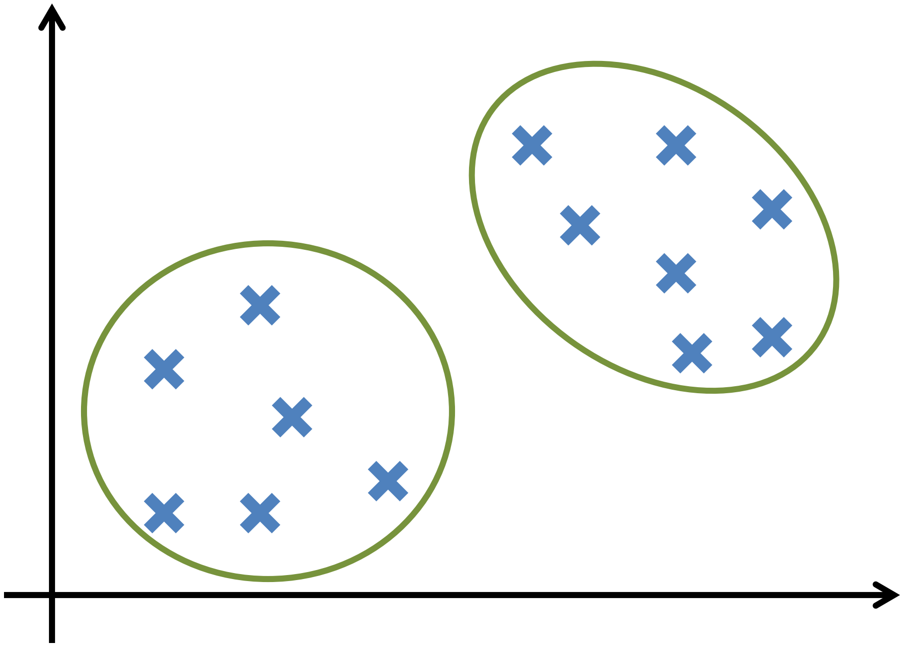
Supervised learning



In supervised learning, the “right answers” (ground truth) are given

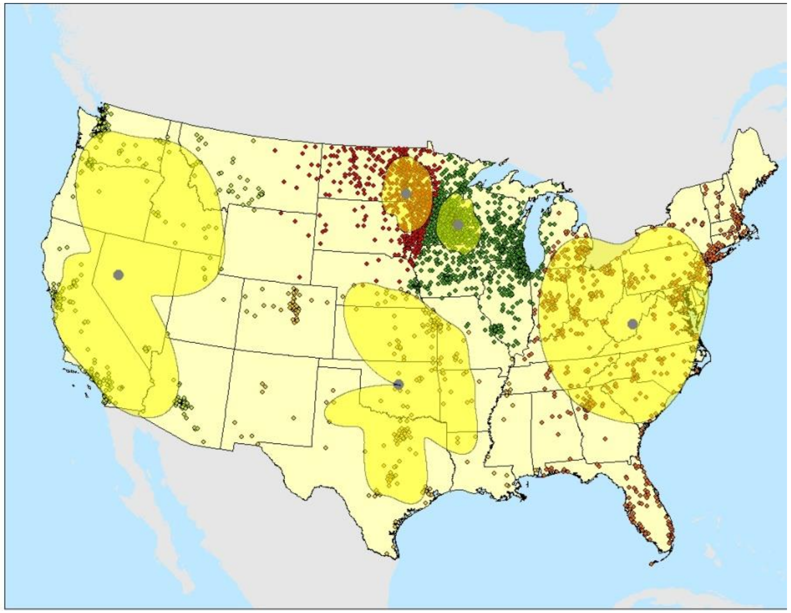
Training Set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

Unsupervised learning

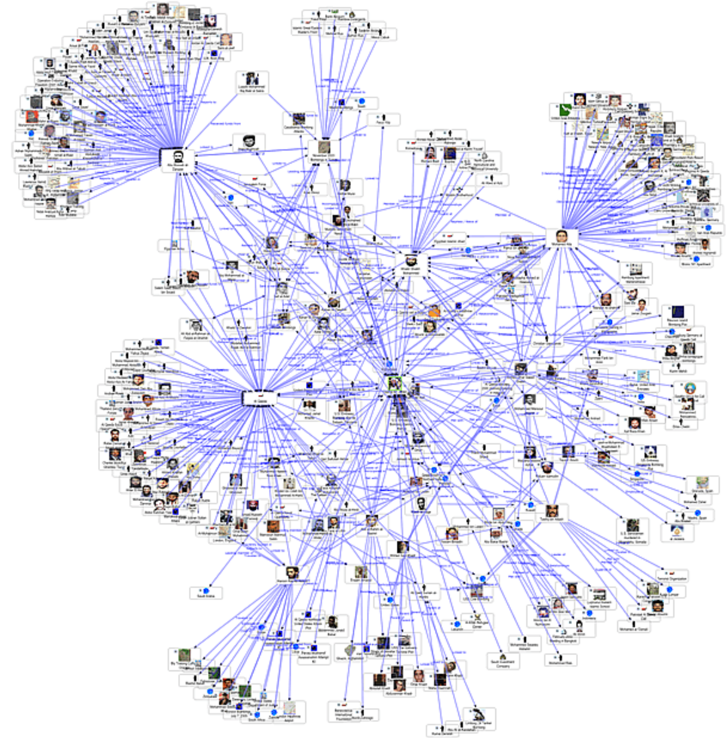


In unsupervised learning, the "right answers" (ground truth) are not given

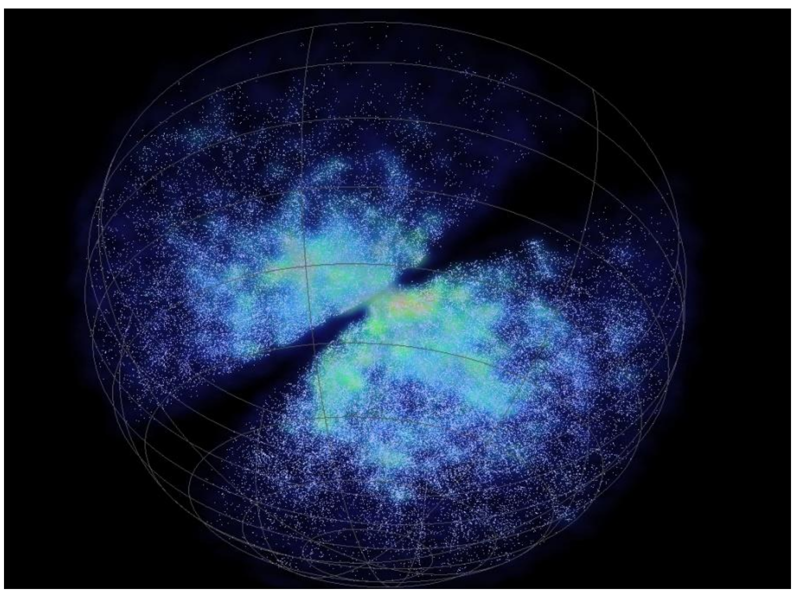
Training Set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$



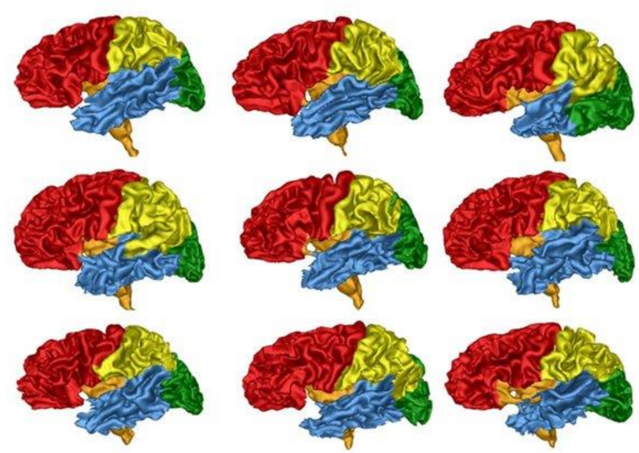
Market segmentation



Social network analysis



Astronomical data analysis



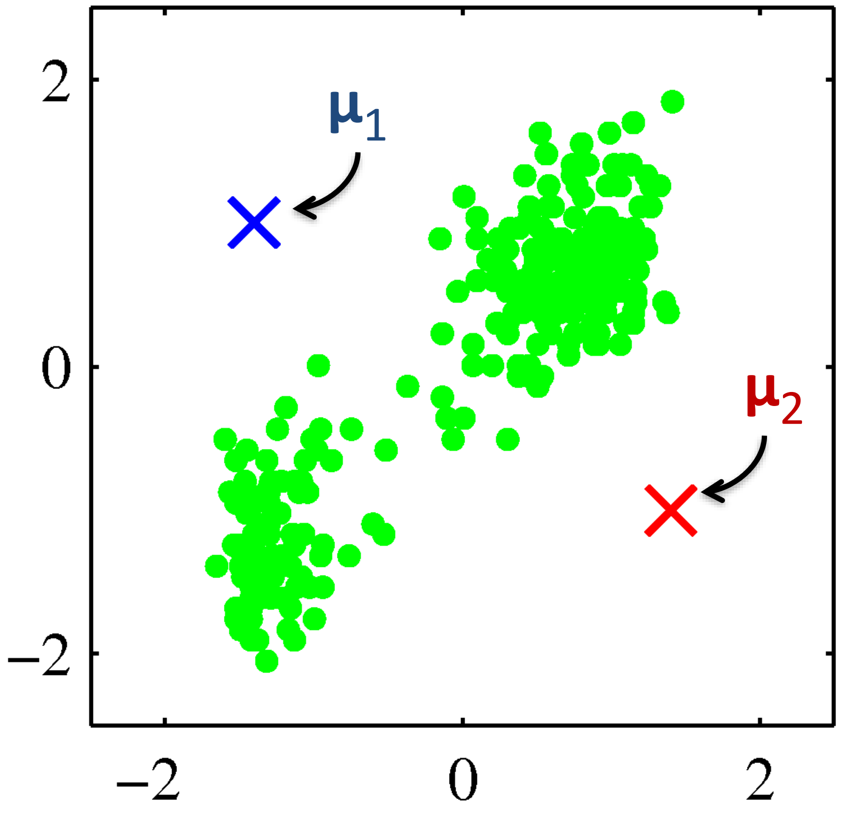
Medical Image Analysis

K-MEANS CLUSTERING ALGORITHM

K-Means Algorithm

Step 1 | Initialize K "means" $\boldsymbol{\mu}_k$, one for each class

Hint: we can use random starting points, or choose k points randomly from the set of samples

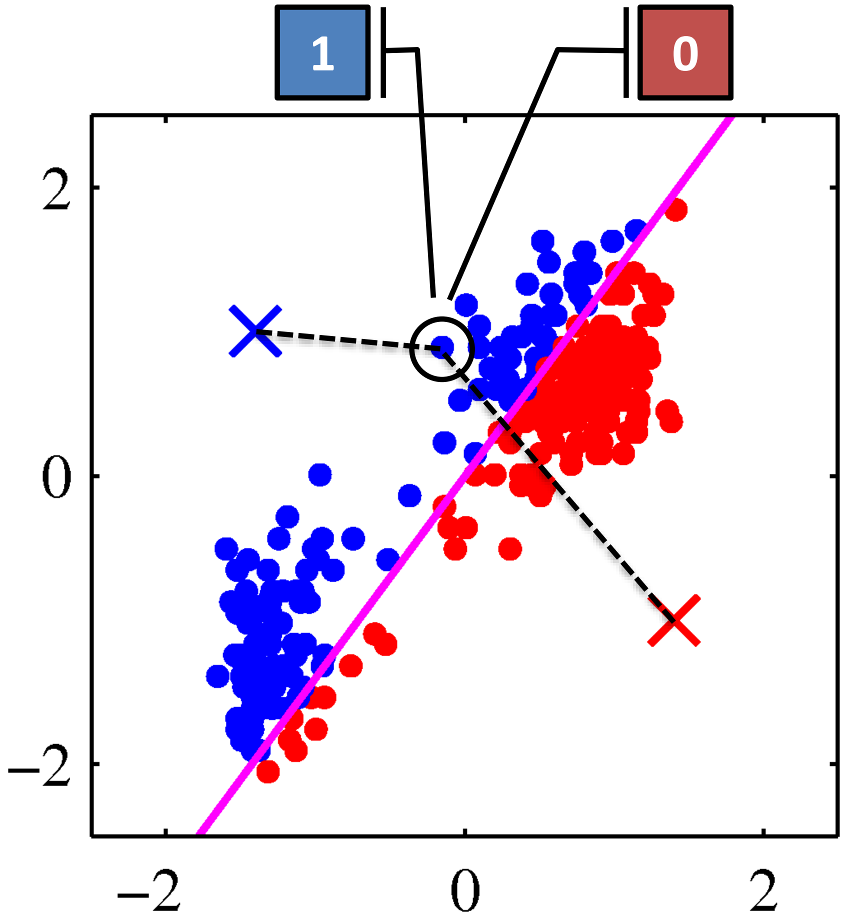


K=2

K-Means Algorithm

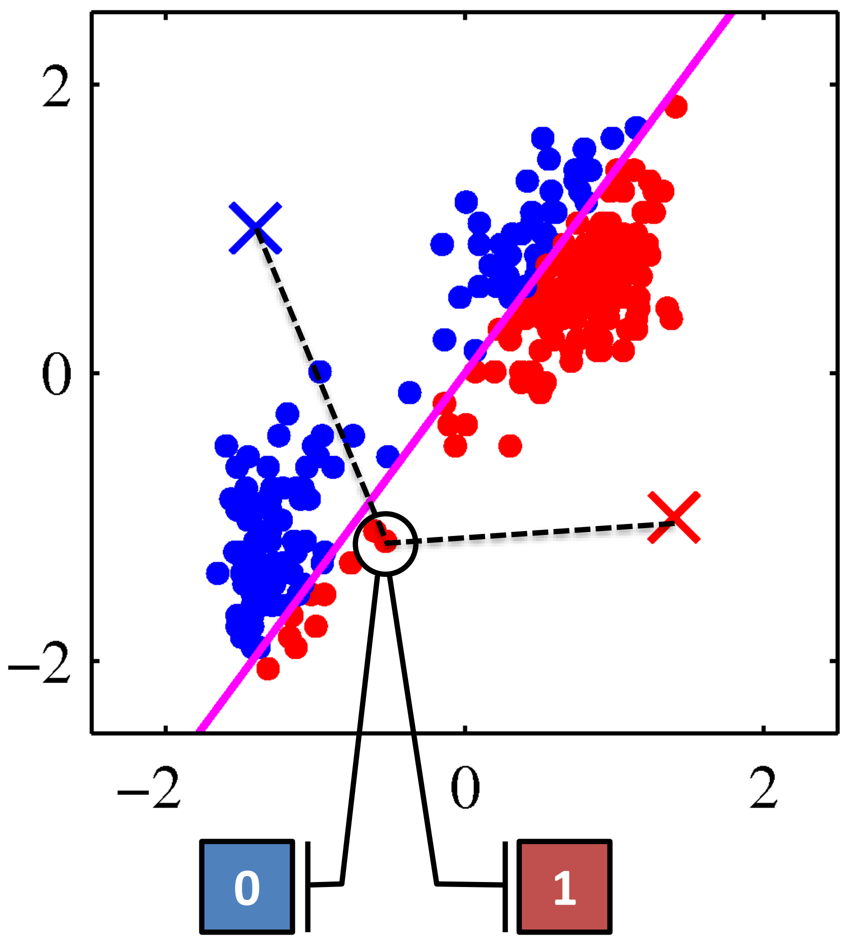
Step 1 | Initialize K "means" $\boldsymbol{\mu}_k$, one for each class

Step 2 | Assign each point to the closest mean $\boldsymbol{\mu}_k$



K-Means Algorithm

Step 1	Initialize K "means" $\boldsymbol{\mu}_k$, one for each class
Step 2	Assign each point to the closest mean $\boldsymbol{\mu}_k$

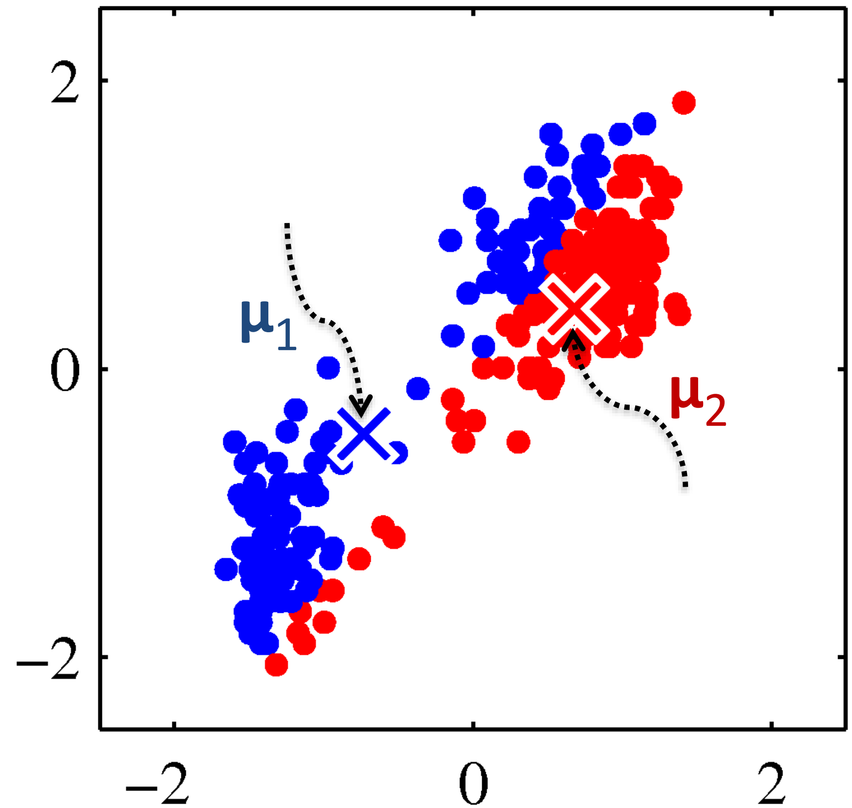


K-Means Algorithm

Step 1 Initialize K "means" $\boldsymbol{\mu}_k$, one for each class

Step 2 Assign each point to the closest mean $\boldsymbol{\mu}_k$

Step 3 Update the means $\boldsymbol{\mu}_k$ of the new clusters



K=2

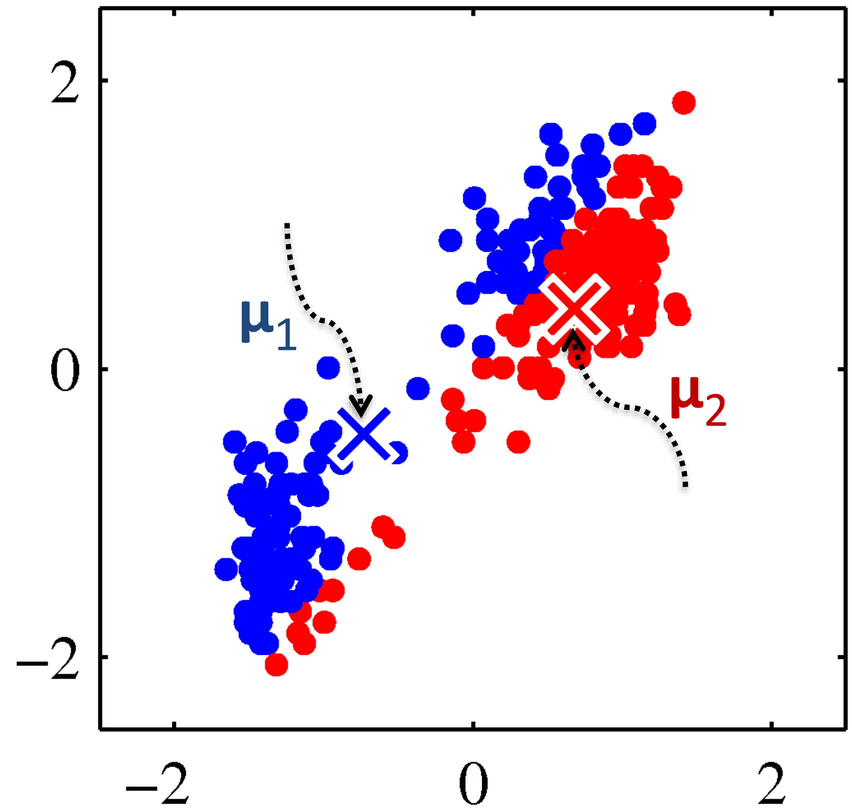
K-Means Algorithm

Step 1 Initialize K "means" $\boldsymbol{\mu}_k$, one for each class

Step 2 Assign each point to the closest mean $\boldsymbol{\mu}_k$

Step 3 Update the means $\boldsymbol{\mu}_k$ of the new clusters

Step 4 Terminate if means do not change anymore



K=2

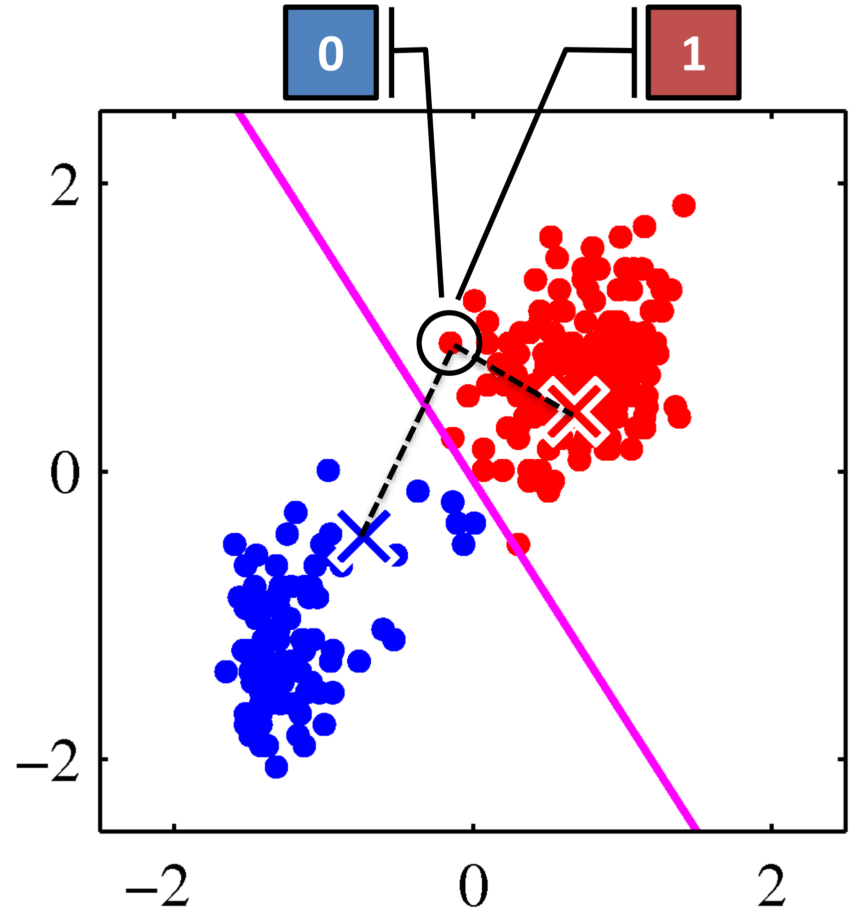
K-Means Algorithm

Step 1 — Initialize K "means" $\boldsymbol{\mu}_k$, one for each class

Step 2 — Assign each point to the closest mean $\boldsymbol{\mu}_k$

Step 3 — Update the means $\boldsymbol{\mu}_k$ of the new clusters

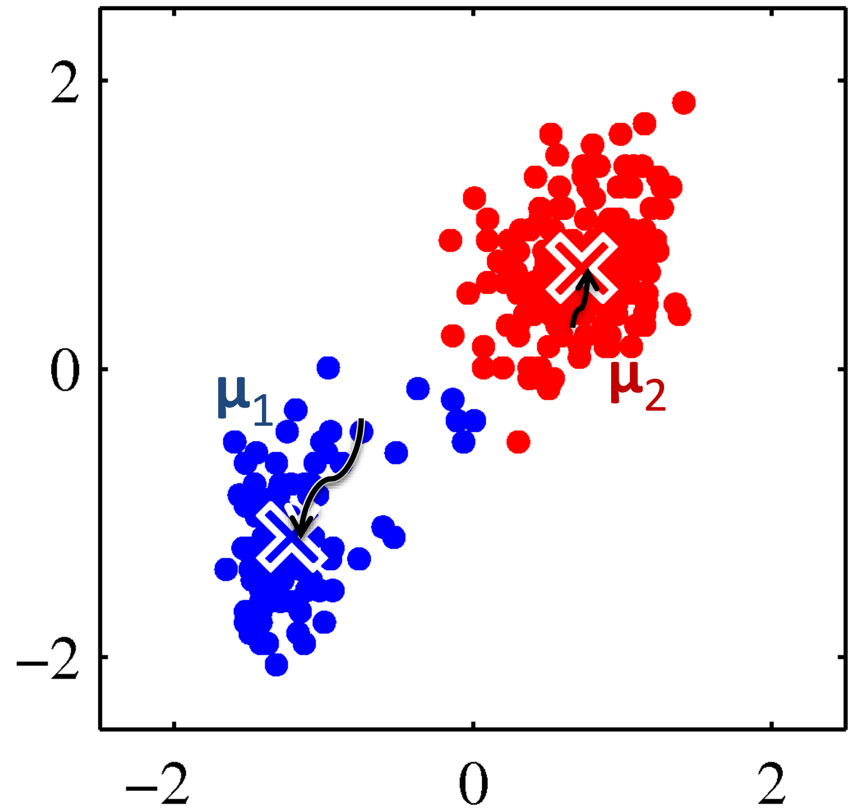
Step 4 — Terminate if means do not change anymore



K=2

K-Means Algorithm

Step 1	Initialize *K* “means” $\boldsymbol{\mu}_k$, one for each class
Step 2	Assign each point to the closest mean $\boldsymbol{\mu}_k$
Step 3	Update the means $\boldsymbol{\mu}_k$ of the new clusters
Step 4	Terminate if means do not change anymore



K=2

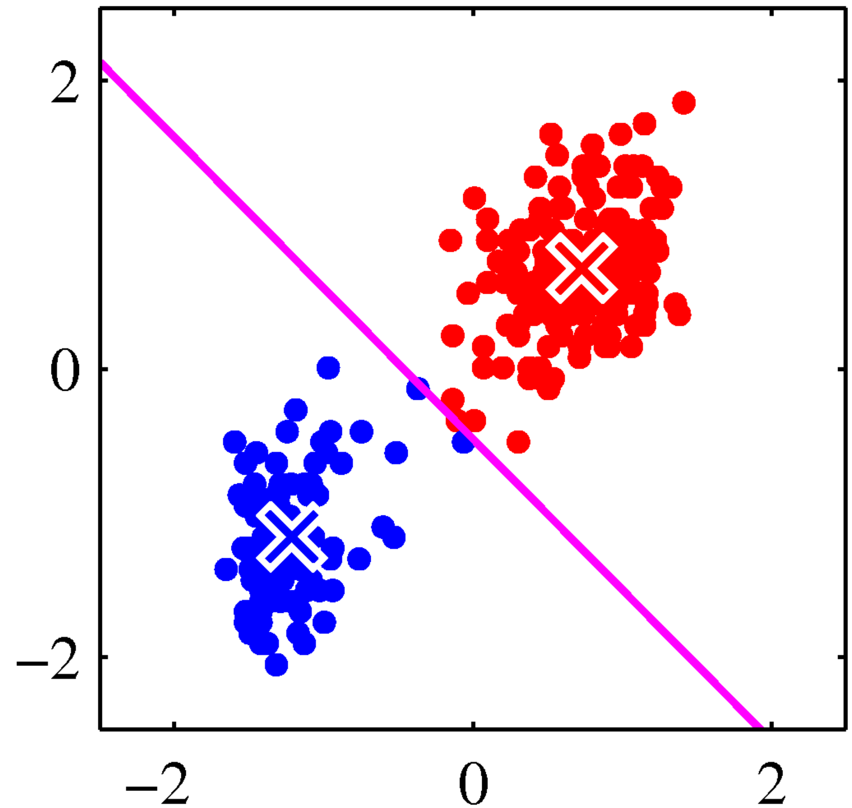
K-Means Algorithm

Step 1 — Initialize K "means" $\boldsymbol{\mu}_k$, one for each class

Step 2 — Assign each point to the closest mean $\boldsymbol{\mu}_k$

Step 3 — Update the means $\boldsymbol{\mu}_k$ of the new clusters

Step 4 — Terminate if means do not change anymore



K=2

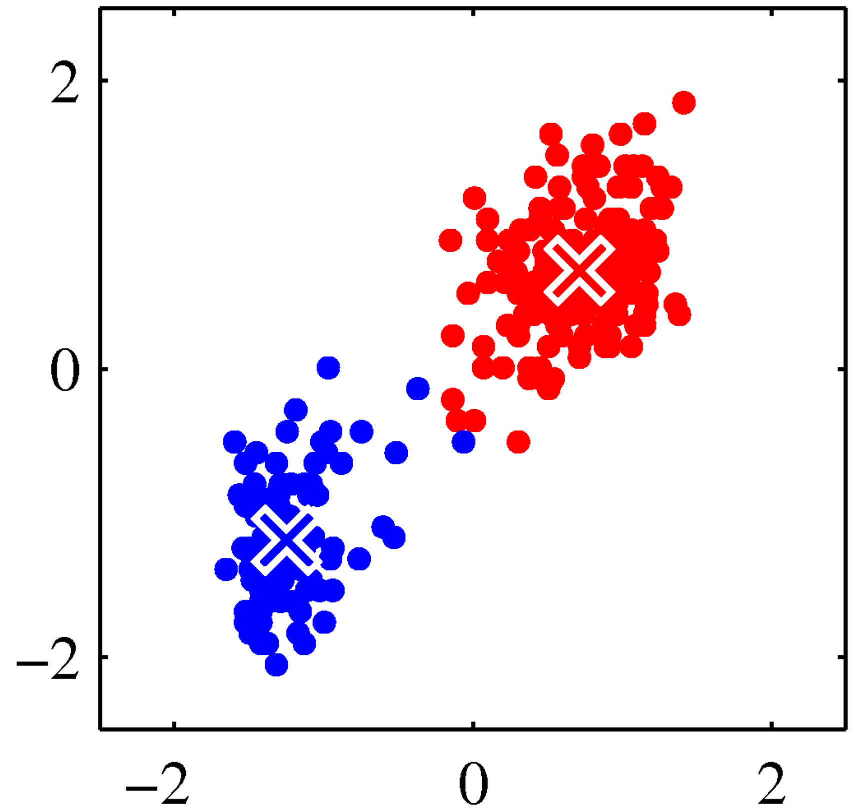
K-Means Algorithm

Step 1 Initialize *K* "means" $\boldsymbol{\mu}_k$, one for each class

Step 2 Assign each point to the closest mean $\boldsymbol{\mu}_k$

Step 3 Update the means $\boldsymbol{\mu}_k$ of the new clusters

Step 4 Terminate if means do not change anymore



K=2

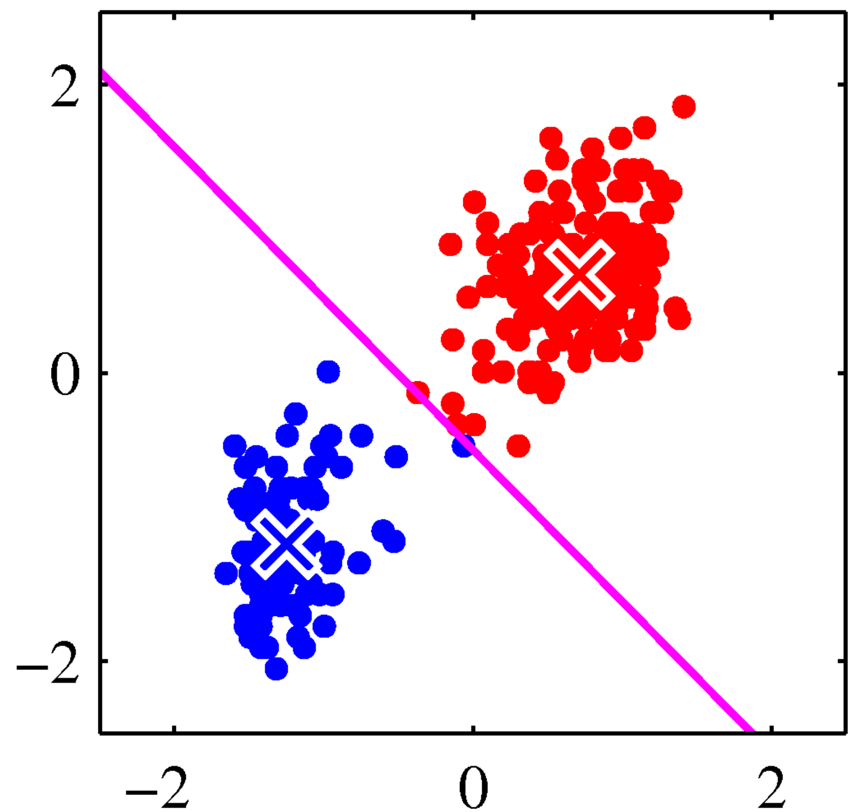
K-Means Algorithm

Step 1 — Initialize *K* "means" $\boldsymbol{\mu}_k$, one for each class

Step 2 — Assign each point to the closest mean $\boldsymbol{\mu}_k$

Step 3 — Update the means $\boldsymbol{\mu}_k$ of the new clusters

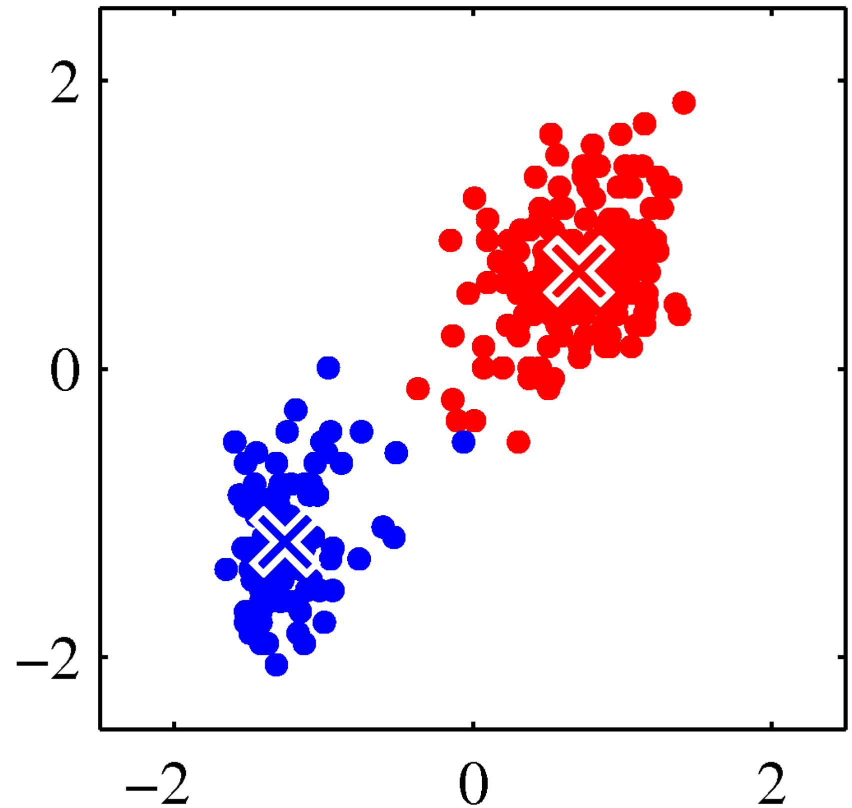
Step 4 — Terminate if means do not change anymore



K=2

K-Means Algorithm

Step	
Step 1	Initialize *K* “means” $\boldsymbol{\mu}_k$, one for each class
Step 2	Assign each point to the closest mean $\boldsymbol{\mu}_k$
Step 3	Update the means $\boldsymbol{\mu}_k$ of the new clusters
Step 4	If means do not change terminate, else back to step 2



K=2

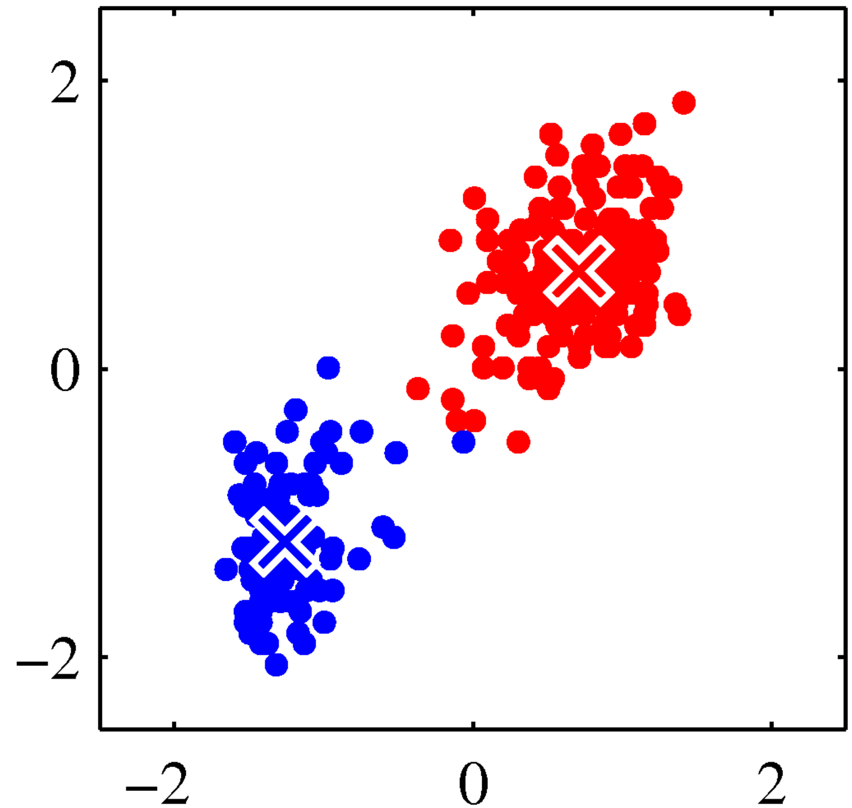
K-Means Algorithm

Step 1 Initialize K “means” $\boldsymbol{\mu}_k$, one for each class

Step 2 Assign each point to the closest mean $\boldsymbol{\mu}_k$

Step 3 Update the means $\boldsymbol{\mu}_k$ of the new clusters

Step 4 If means do not change terminate, else back to step 2



K=2

Optimization Objective

Let´s define some notation to describe the assignment of data points to clusters:

For each data point $x^{(n)}$ we introduce a corresponding set of binary indicator variables:

$$r_{nk} \in \{0,1\}$$

where *k = 1, …, K* describing which of the *K* clusters the data point $x^{(n)}$ is assigned to

We define an objective function (cost function, distortion measure) as:

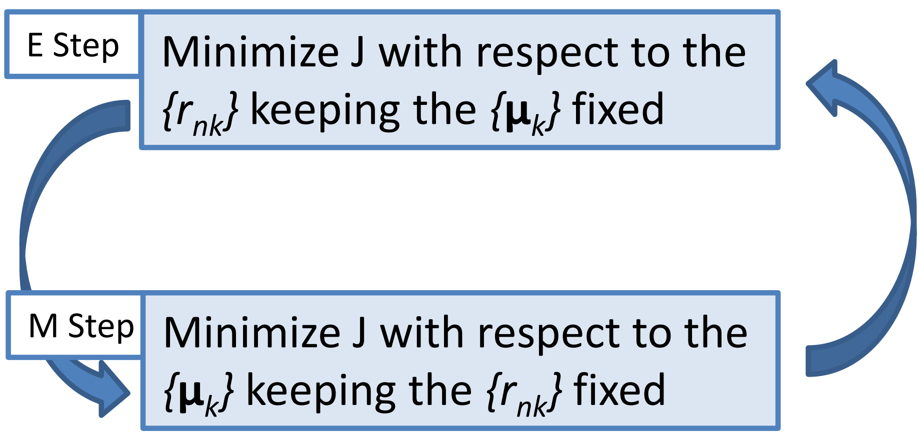
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| \mathbf{x}^{(n)} - \boldsymbol{\mu}_k \right\|^2$$

which represents the sum of the squares of distances of each data point to its assigned vector $\boldsymbol{\mu}_k$

Optimization Objective

Goal: find the values for the $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$ so as to minimize J

Intuition: we can do this through an iterative procedure in which each iteration involves two successive steps corresponding to successive optimizations with respect to the $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$



Note: We will see next that these two steps correspond to the E (expectation) and M (Maximization) steps of the EM algorithm

Expectation step

E Step — Minimize J with respect to the $\{r_{nk}\}$ keeping the $\{\boldsymbol{\mu}_k\}$ fixed

M Step — Minimize J with respect to the $\{\boldsymbol{\mu}_k\}$ keeping the $\{r_{nk}\}$ fixed

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

Observe:

- J is a linear function of r_{nk} -> **closed form** solution
- The terms involving different n are independent -> we can optimise for each n **separately** by choosing r_{nk} to be *1* for whichever value of k gives the minimum value of $\|\mathbf{x}^{(n)}-\boldsymbol{\mu}_k\|^2$

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \left\| x^{(n)} - \mu_j \right\|^2 \\ 0 & \text{otherwise} \end{cases}$$

Maximisation Step

E Step — Minimize J with respect to the $\{r_{nk}\}$ keeping the $\{\boldsymbol{\mu}_k\}$ fixed

M Step — Minimize J with respect to the $\{\boldsymbol{\mu}_k\}$ keeping the $\{r_{nk}\}$ fixed

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

Observe:

- J is a quadratic function of $\boldsymbol{\mu}_{nk}$ -> minimise setting the derivative with respect to $\boldsymbol{\mu}_{nk}$ to zero

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 2 \sum_{n=1}^{N} r_{nk} \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k \right) = 0$$

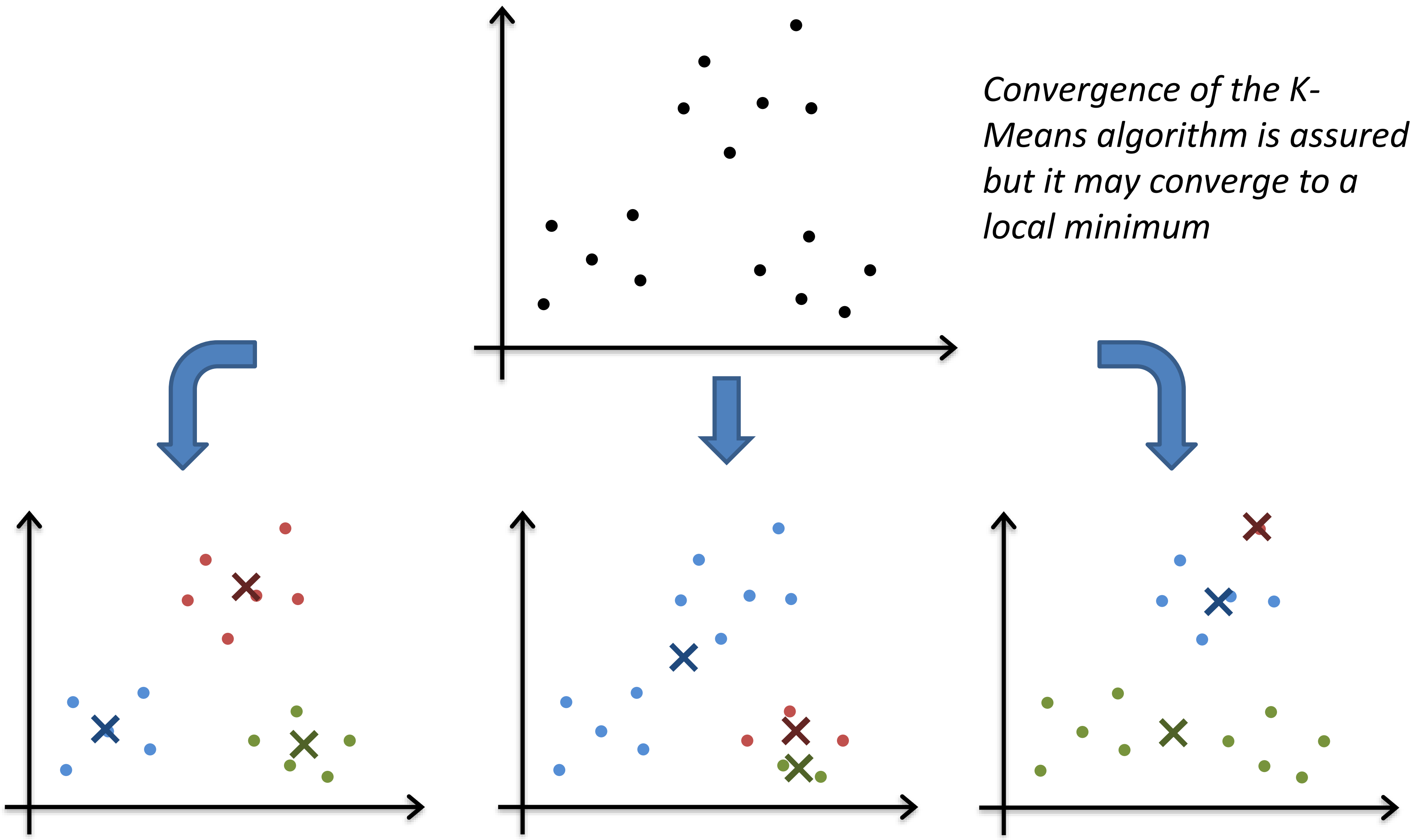
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}^{(n)}}{\sum_n r_{nk}}$$

K-Means Algorithm Revisited

Step 1	Initialize *K* "means" $\boldsymbol{\mu}_k$, one for each class
Step 2	Assign each point to the closest mean $\boldsymbol{\mu}_k$
Step 3	Update the means $\boldsymbol{\mu}_k$ of the new clusters
Step 4	If means do not change terminate, else back to step 2

Initialization	Initialize *K* "means" $\boldsymbol{\mu}_k$, one for each class
Expectation	Minimize *J* with respect to the $\{r_{nk}\}$ keeping the $\{\boldsymbol{\mu}_k\}$ fixed
Maximisation	Minimize *J* with respect to the $\{\boldsymbol{\mu}_k\}$ keeping the $\{r_{nk}\}$ fixed
Terminate	Terminate if *J has* converged, otherwise back to E-Step

Random Initialization



Convergence of the K-Means algorithm is assured but it may converge to a local minimum

Random Initialization

A possible solution: repeat many times and select the best fit.

for $i = 1$ to 100

{

  Randomly initialize K-means

  Run K-means, and get the $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$

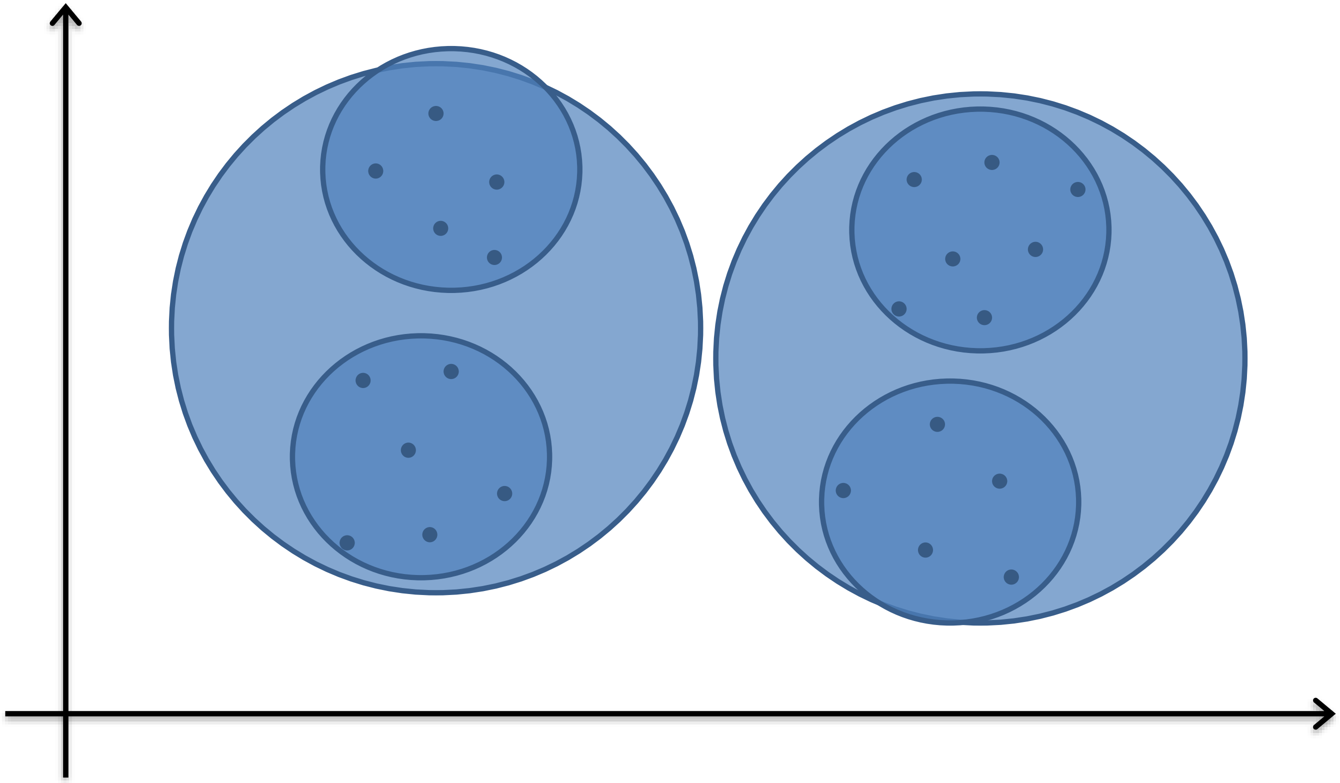
  Compute the cost function J

}

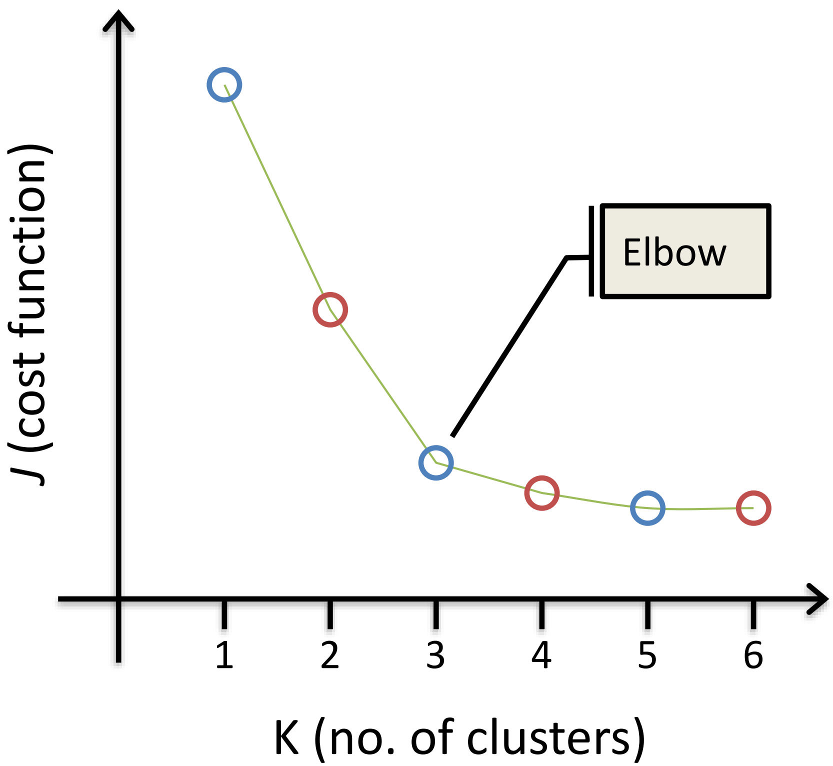
Pick clustering that gave lowest cost J

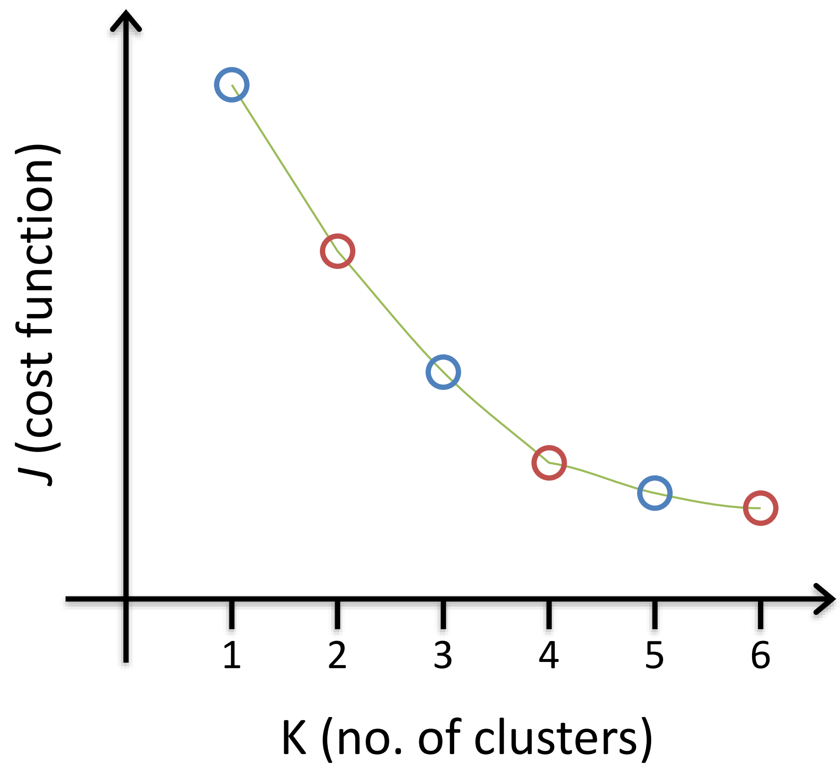
Choosing the number of Clusters

What is the right number (K) of classes?



Choosing the number of Clusters

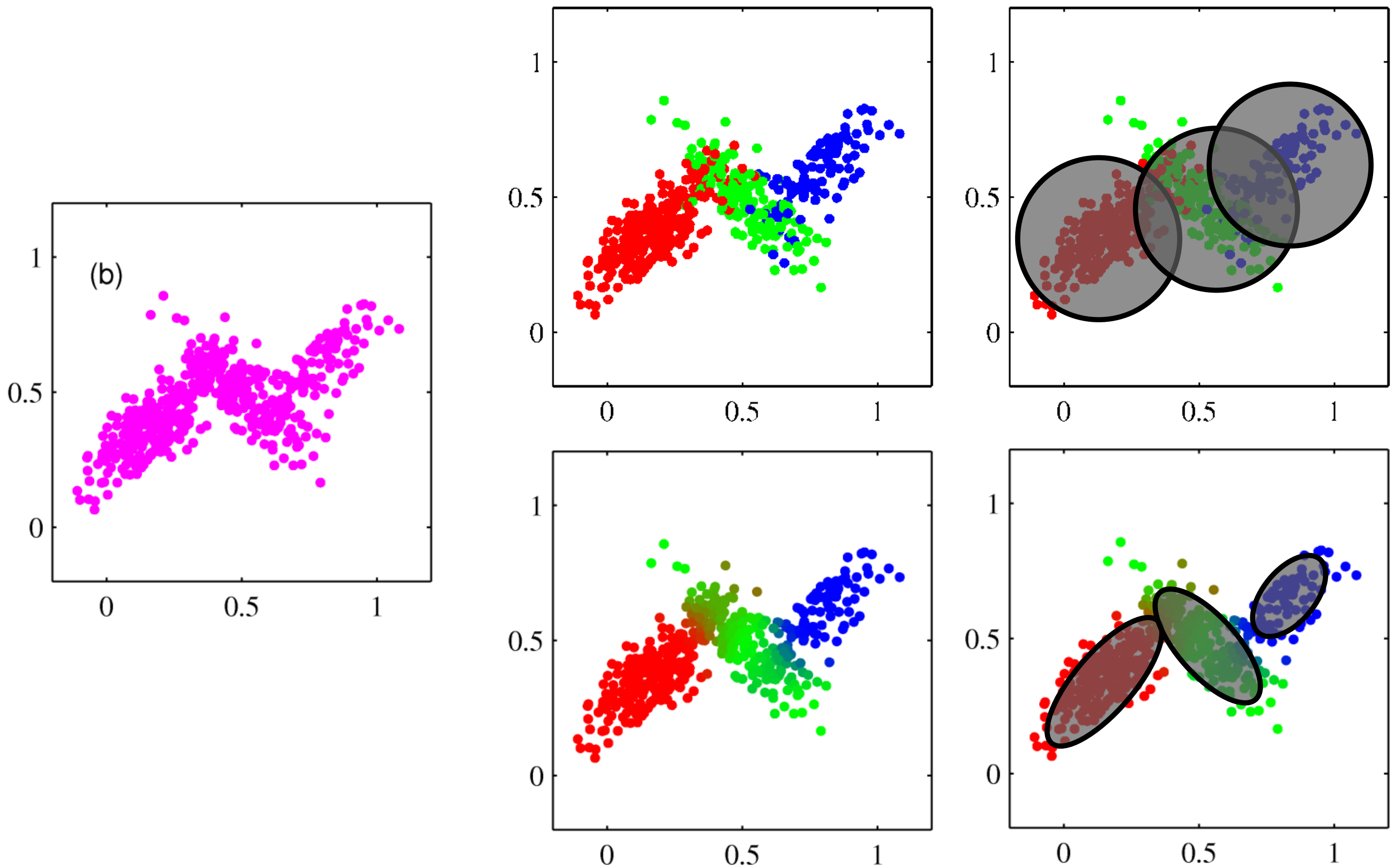




The first clusters will add much information (explain a lot of variance / reduce a lot the value of the cost function), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified.

Limitations of K-means

- A point can only pertain to one class (hard classification)
- Clusters have the same "shape" (only parameters are the means $\boldsymbol{\mu}_k$)



K-means

How do we go about this?

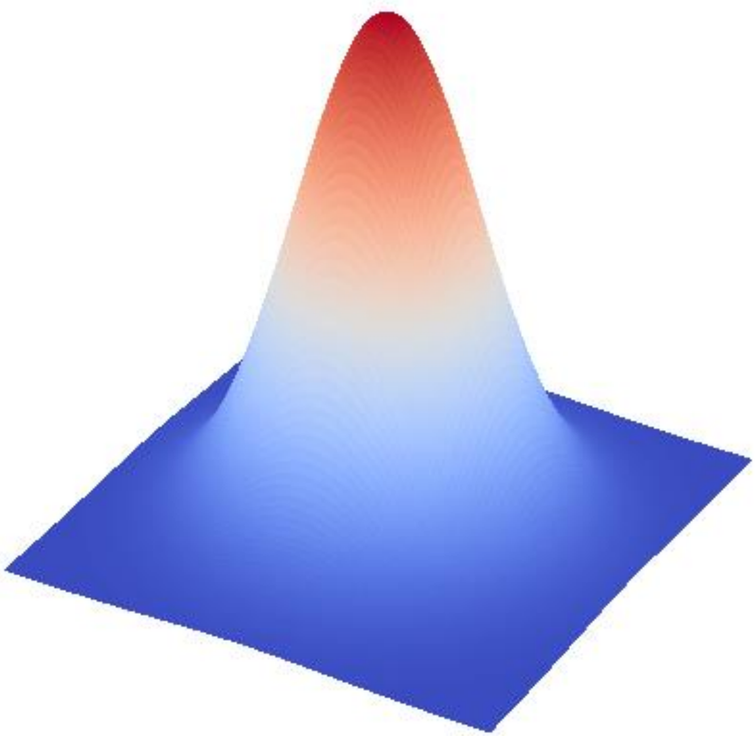
GAUSSIAN MIXTURE MODELS

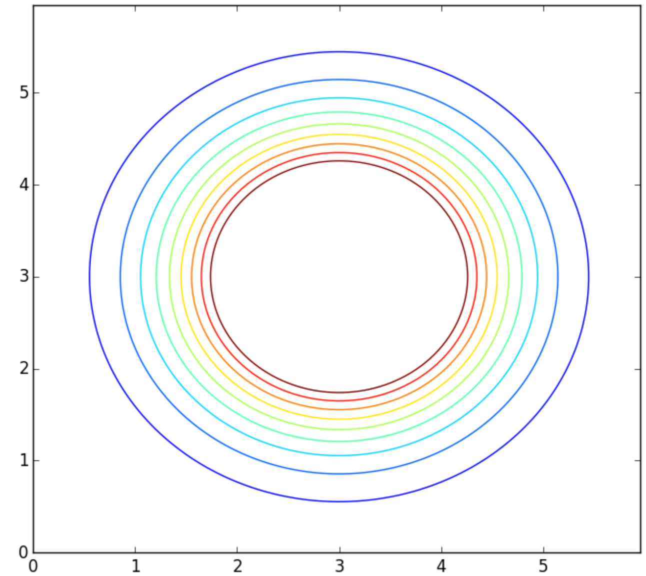
A Gaussian Density

$$p(\mathbf{x}) = N\left(\mathbf{x} \middle| \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

Mean

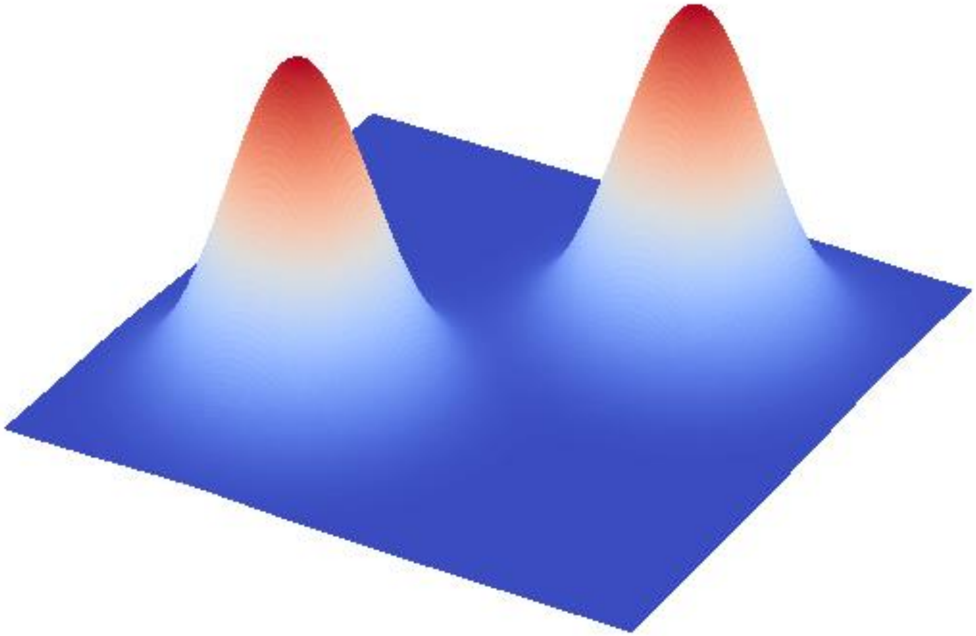
Variance

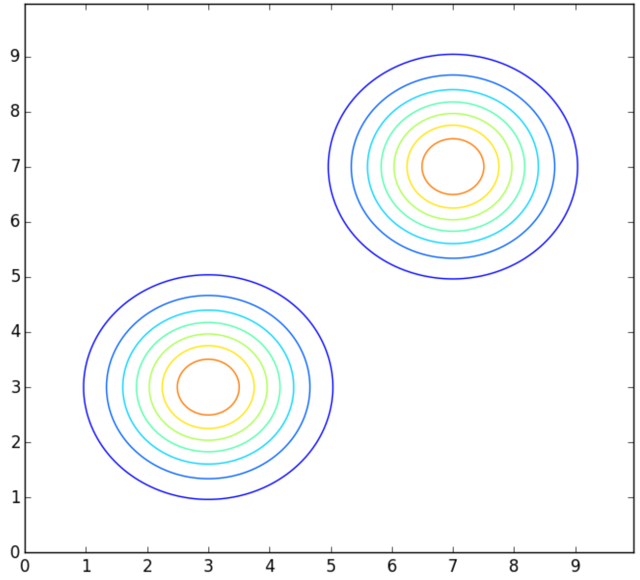




A Gaussian Mixture

$$N(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + N(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$



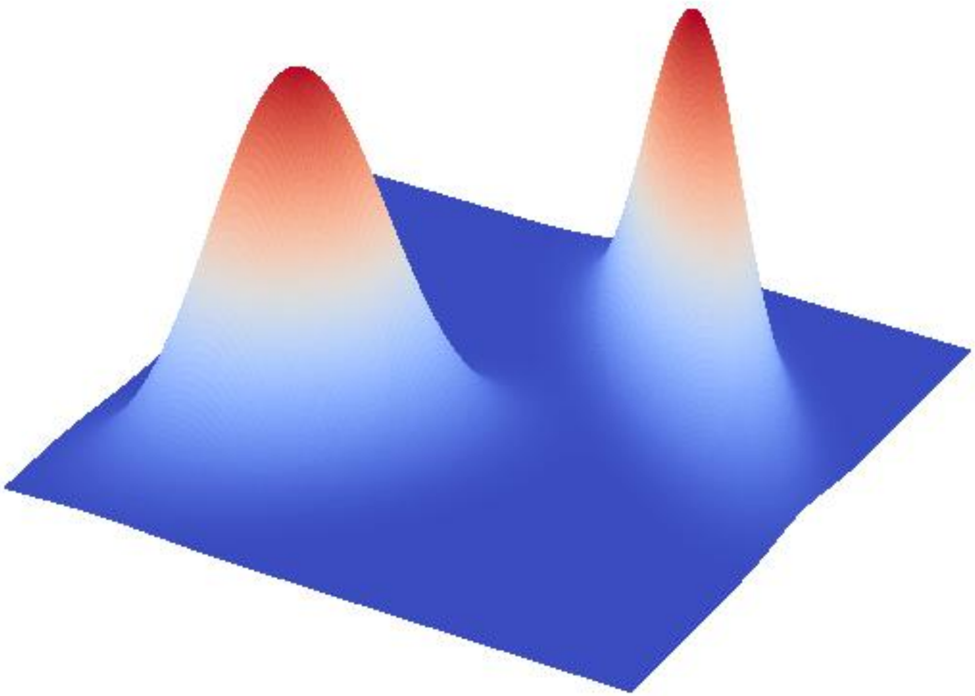


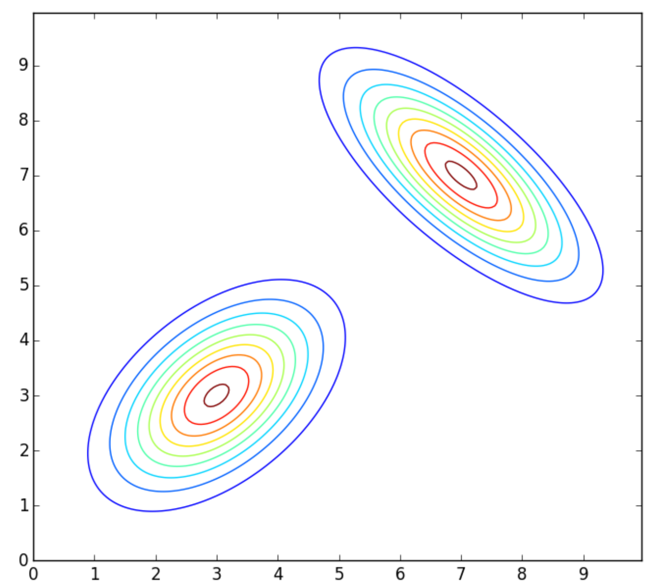
A Gaussian Mixture

Different Variances

$$N(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + N(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

Different Variances

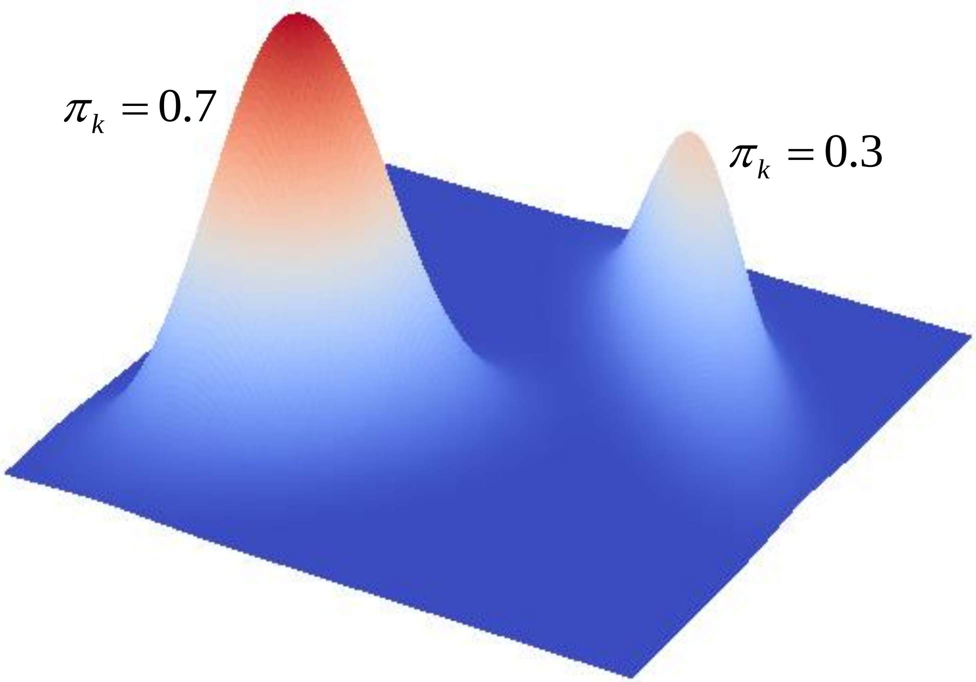


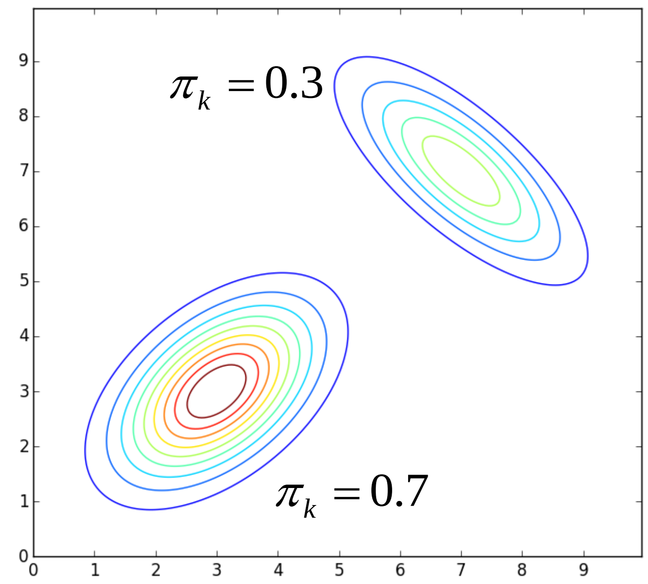


A Gaussian Mixture

$$p(\mathbf{x}) = \underbrace{\pi_1}_{\text{Mixing coefficient}} N\left(\mathbf{x} \middle| \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right) + \overbrace{\pi_2}^{\text{Mixing coefficient}} N\left(\mathbf{x} \middle| \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right)$$

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





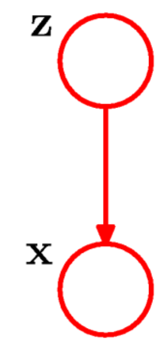
Gaussian Mixture Distribution

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

A Different Formulation

Given a mixture of Gaussians as an underlying probability distribution each sample is produced in two steps, by first choosing one of the Gaussian mixture components, and then according to the selected component's probability density

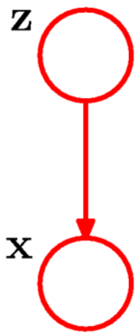
Intuition: We do not directly observe which component was responsible for each sample – this "responsible component" can be represented by a **latent (hidden) variable *z***



A Different Formulation

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

$$z_k = \begin{cases} 1 & \text{for a particular element, } k = i \\ 0 & \text{for all other elements, } k \neq i \end{cases}$$



$$z_k \in \{0,1\}$$

$$\sum_k z_k = 1$$

We define: $p(z_k = 1) = \pi_k$

A Different Formulation

As z uses a *1-of-K* representation we can write:

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

Similarly, for the conditional distribution of x given a particular value of z:

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

Finally, the marginal distribution of x over all possible states of z:

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

A Different Formulation

Using the Bayes theorem:

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

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

$\gamma(z_k)$ can be seen as the "responsibility" that component k takes for "explaining" the observation **x**

K-means vs Mixture of Gaussians

K-Means

- K-means is a ("hard") classifier
- Parameters to estimate
  - Means $\{\boldsymbol{\mu}_k\}$

Mixture of Gaussians

- Mixture of Gaussians is a probability model
- We can USE it as a "soft" classifier
- Parameters to estimate
  - Means $\{\boldsymbol{\mu}_k\}$
  - Covariances $\{\boldsymbol{\Sigma}_k\}$
  - Mixing coefficients $\{\pi_k\}$

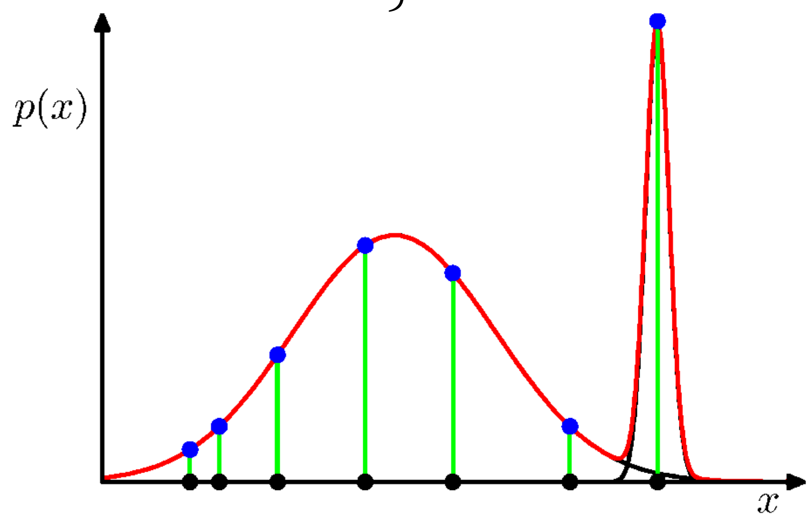
Maximum Likelihood

Goal: given a data set of observation {x(1), x(2), ..., x(N)) we wish to model the data using a mixture of Gaussians

Intuition: follow the same practice as before and maximise the (log) likelihood of the model parameters

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

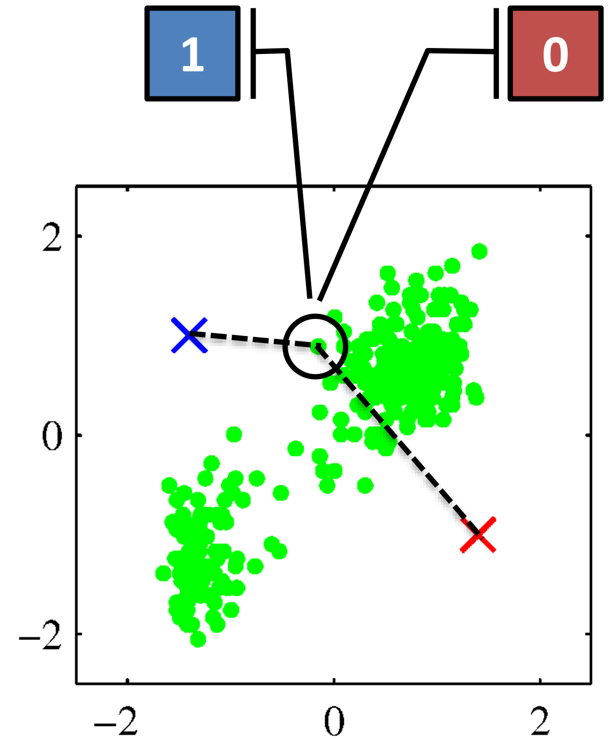
Problem: contrary to estimating the parameters of a single Gaussian, the presence of singularities makes this not a well posed problem



EXPECTATION MAXIMISATION FOR GAUSSIAN MIXTURES

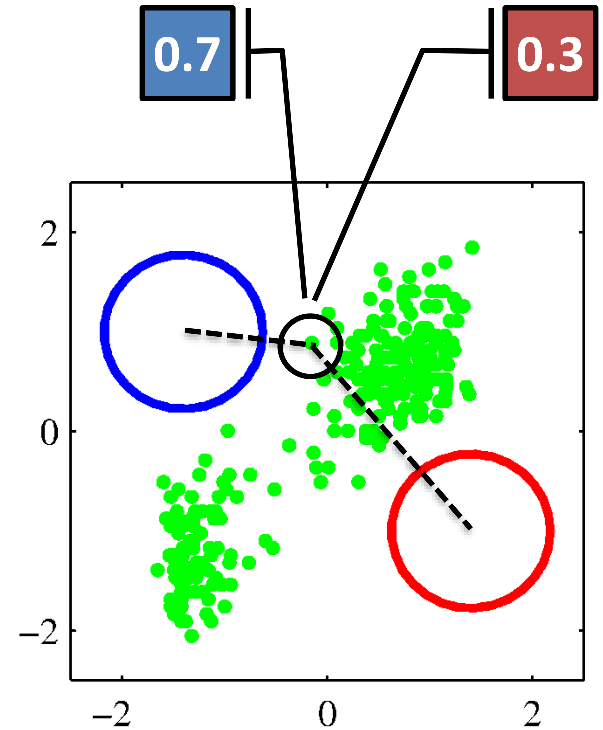
K-Means Algorithm Reminder

Initialization	Initialize K "means" $\boldsymbol{\mu}_k$, one for each class
Expectation	Minimize J with respect to the $\{r_{nk}\}$ keeping the $\{\boldsymbol{\mu}_k\}$ fixed
Maximisation	Minimize J with respect to the $\{\boldsymbol{\mu}_k\}$ keeping the $\{r_{nk}\}$ fixed
Terminate	Terminate if J *has* converged, otherwise back to E-Step



Expectation Maximization (EM) for Gaussian Mixtures

Initialization	Initialize the parameters of *K* Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster *k*
Maximisation	Given the scores, adjust $\{\boldsymbol{\mu}_k , \pi_k, \boldsymbol{\Sigma}_k\}$ for each cluster *k*
Terminate	Evaluate log likelihood, stop when converged, else back to E-step



Expectation Maximization (EM) for Gaussian Mixtures

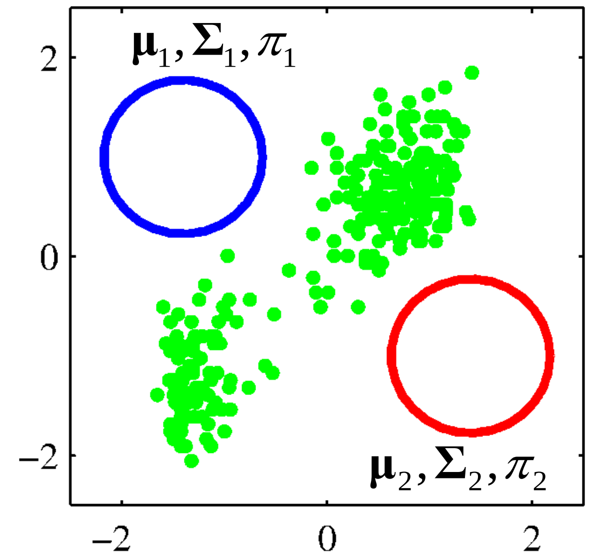
Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)

Hint: we can use the K-means result to initialize

$$\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k$$

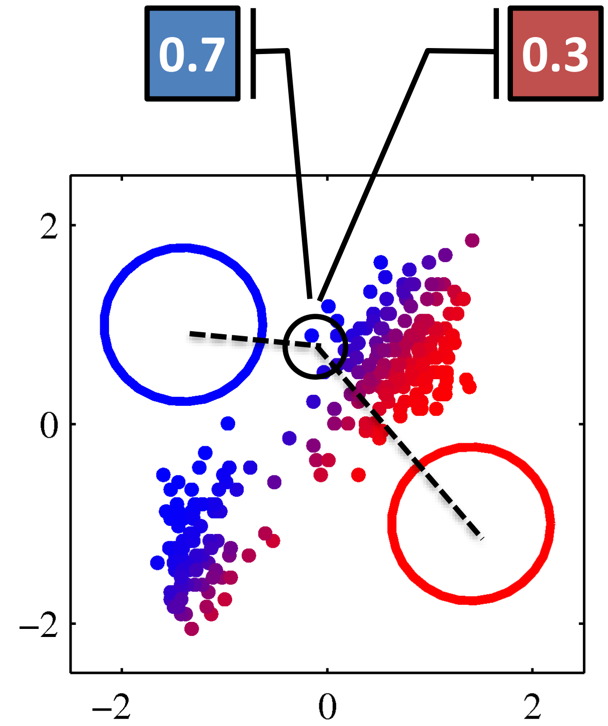
$$\boldsymbol{\Sigma}_k \leftarrow \mathrm{cov}\big(\mathrm{cluster}(k)\big)$$

$$\pi_k \leftarrow \frac{\#\,\text{points in cluster } k}{\text{Total number of points}}$$



Expectation Maximization (EM) for Gaussian Mixtures

Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster k

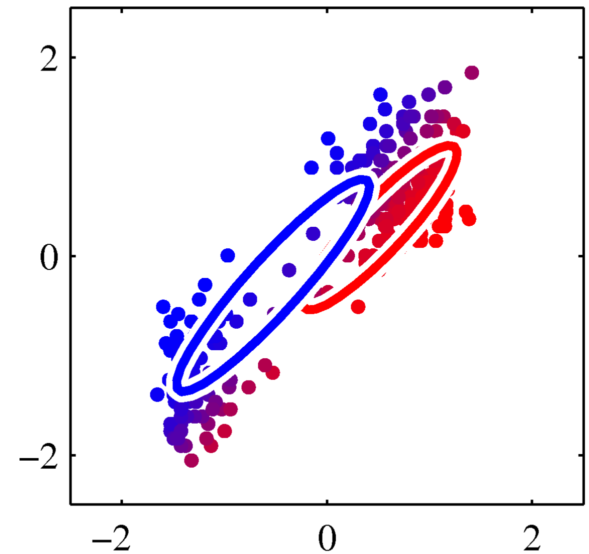


$$\gamma(z_{nk}) = \frac{\pi_k N\left(\mathbf{x}^{(n)} \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j N\left(\mathbf{x}^{(n)} \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$

$\gamma(z_{nk})$ is called "responsibility": how much is the Gaussian k responsible for the point $x^{(n)}$

Expectation Maximization (EM) for Gaussian Mixtures

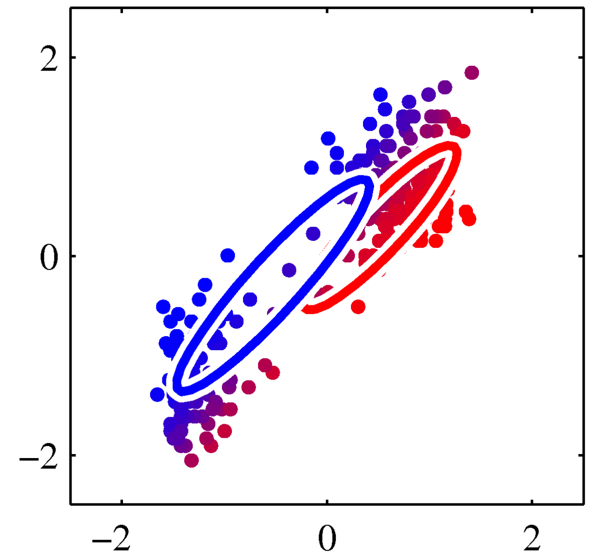
Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster k
Maximisation	Given the scores, adjust $\{\boldsymbol{\mu}_k, \pi_k, \boldsymbol{\Sigma}_k\}$ for each cluster k



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

Expectation Maximization (EM) for Gaussian Mixtures

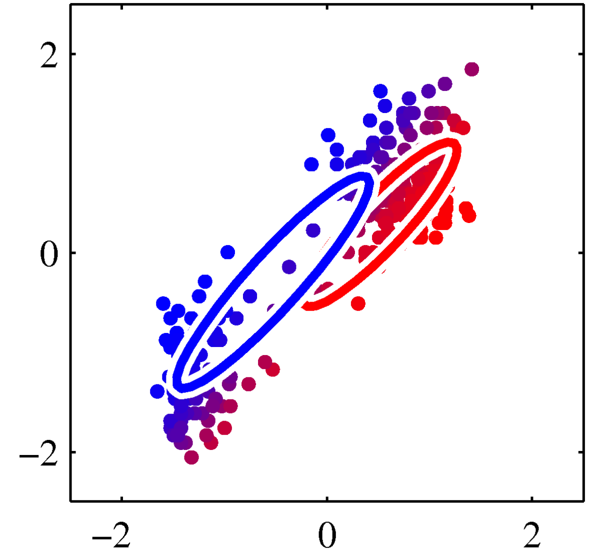
Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster k
Maximisation	Given the scores, adjust $\{\boldsymbol{\mu}_k, \pi_k, \boldsymbol{\Sigma}_k\}$ for each cluster k



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

Expectation Maximization (EM) for Gaussian Mixtures

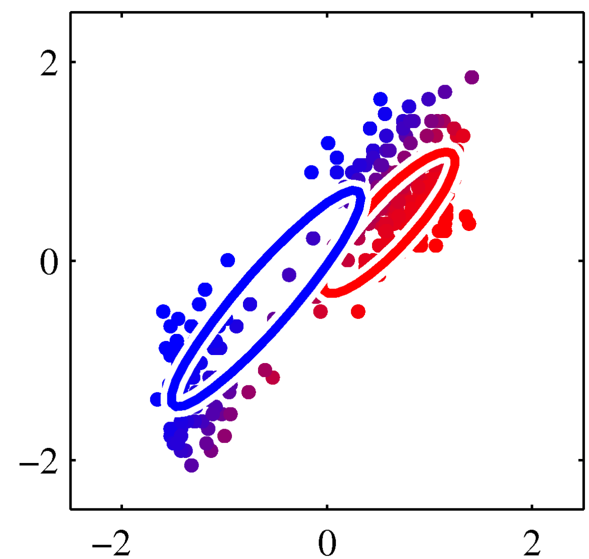
Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster k
Maximisation	Given the scores, adjust $\{\boldsymbol{\mu}_k , \pi_k, \boldsymbol{\Sigma}_k\}$ for each cluster k



$$\pi_k^{\text{new}} = \frac{N_k}{N} \qquad N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

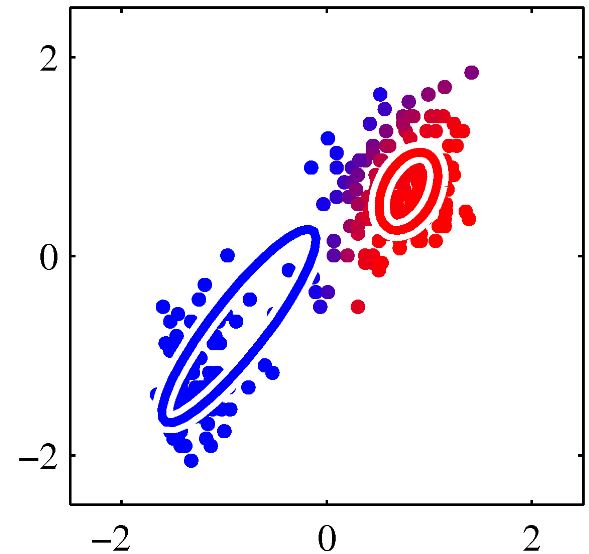
Expectation Maximization (EM) for Gaussian Mixtures

Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster k
Maximisation	Given the scores, adjust $\{\boldsymbol{\mu}_k, \pi_k, \boldsymbol{\Sigma}_k\}$ for each cluster k
Terminate	Evaluate log likelihood, stop when converged , else back to E-step



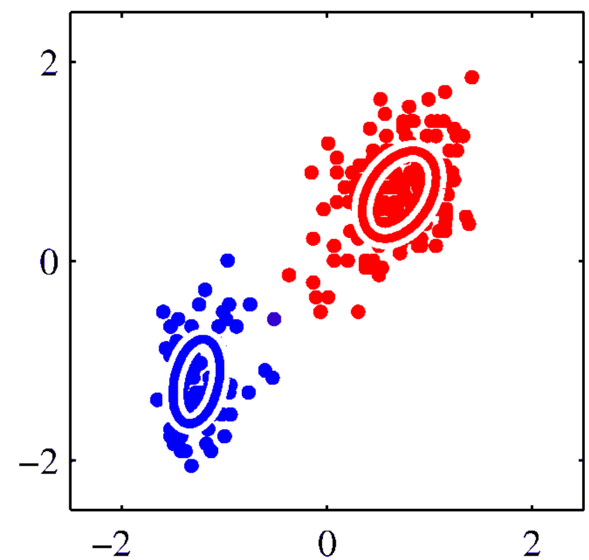
Expectation Maximization (EM) for Gaussian Mixtures

Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster k
Maximisation	Given the scores, adjust $\{\boldsymbol{\mu}_k, \pi_k, \boldsymbol{\Sigma}_k\}$ for each cluster k
Terminate	Evaluate log likelihood, stop when converged , else back to E-step



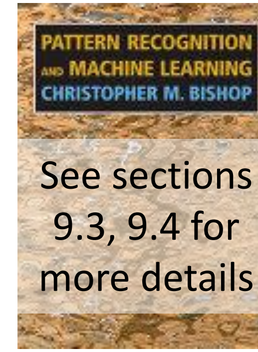
Expectation Maximization (EM) for Gaussian Mixtures

Initialization	Initialize the parameters of K Gaussians ($\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k)
Expectation	Give each point $x^{(n)}$ an assignment score $\gamma(z_{nk})$ for each cluster k
Maximisation	Given the scores, adjust $\{\boldsymbol{\mu}_k, \pi_k, \boldsymbol{\Sigma}_k\}$ for each cluster k
Terminate	Evaluate log likelihood, stop when converged , else back to E-step



THE GENERAL EXPECTATION MAXIMISATION ALGORITHM

General EM Algorithm



Initialization

Initialize parameters θ^{old}

Expectation

Evaluate $p(Z|X, \theta^{old})$

Latent (hidden) variables

Observed variables

Maximisation

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

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

Likelihood

Terminate

Evaluate log likelihood, if it converges, stop. Otherwise do $\theta^{old} \leftarrow \theta^{new}$ and go back to E-step

SUMMARY

Summary

- K-Means is an unsupervised method for data clustering
- K-means yields "hard" cluster assignments
- Gaussian Mixture Models are useful for modelling data with "soft" cluster assignments
- Expectation Maximization is a method used when we have a model with latent (hidden) variables
- K-Means can be seen as a special case of the EM algorithm $\left(\Sigma = \varepsilon \cdot I, \varepsilon \to 0\right)$

What's Next

		Mondays	**Tuesdays**				
		16:00 - 18:00	**15:00 - 17:00**				
Practical Sessions		**M**	**T**	**W**	**T**	**F**	**Lectures**
	Feb	8	9	10	11	12	Introduction and Linear Regression
P0. Introduction to Python, Linear Regression		15	16	17	18	19	Logistic Regression, Normalization
P1. Text non-text classification (Logistic Regression)		22	23	24	25	26	Regularization, Bias-variance decomposition
	Mar	29	1	2	3	4	Normalization and subspace methods (dimensionality reduction)
		7	8	9	10	11	Probabilities, Bayesian inference
Discussion of intermediate deliverables / project presentations		14	15	16	17	18	Parameter Estimation, Bayesian Classification
		21	22	23	24	25	Easter Week
	Apr	28	29	30	31	1	Clustering, Gausian Mixture Models, Expectation Maximisation
P2. Feature learning (k-means clustering, NN, bag of words)		4	5	6	7	8	Nearest Neighbour Classification
		11	12	13	14	15	
		18	19	20	21	22	Kernel methods
Discussion of intermediate deliverables / project presentations		25	26	27	28	29	Support Vector Machines, Support Vector Regression
P3. Text recognition (multi-class classification using SVMs)	**May**	2	3	4	5	6	Neural Networks
		9	10	11	12	13	Advanced Topics: Metric Learning, Preference Learning
		16	17	18	19	20	Advanced Topics: Deep Nets
Final Project Presentations		23	24	25	26	27	Advanced Topics: Structural Pattern Recognition
	Jun	30	31	1	2	3	Revision

LEGEND	
	Project Follow Up
	Project presentations
	Lectures
	Project Deliverable due date
	Vacation / No Class