TFIP-AI – Advanced Machine Learning

Unit 2 Clustering (Gaussian Mixture)

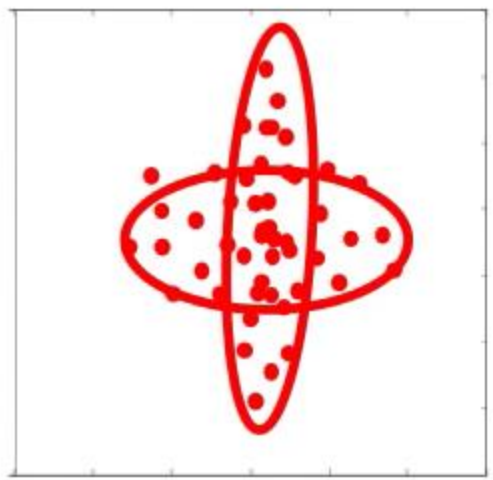
Iterative Schema + Gaussian → EM

EM algorithm

Mixtures of Gaussians

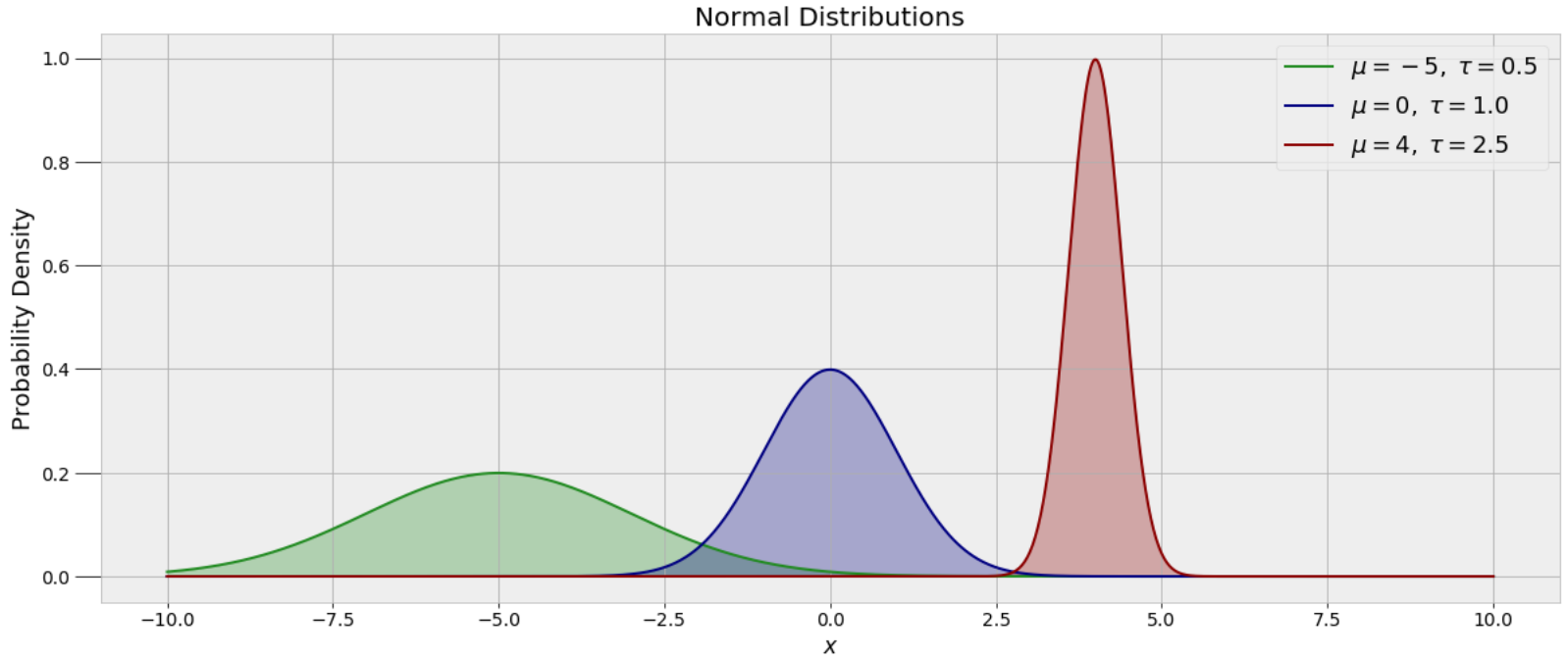
- K-means algorithm
  - Assigned each example to exactly one cluster
  - What if clusters are overlapping?
    - Hard to tell which cluster is right
    - Maybe we should try to remain uncertain
  - Used Euclidean distance
  - What if cluster has a non-circular shape?

- Gaussian mixture models
  - Clusters modeled as Gaussians
    - Not just by their mean
  - EM algorithm: assign data to cluster with some *probability*
  - Gives probability model of x! ("generative")



Gaussian/Normal Distribution – univariate

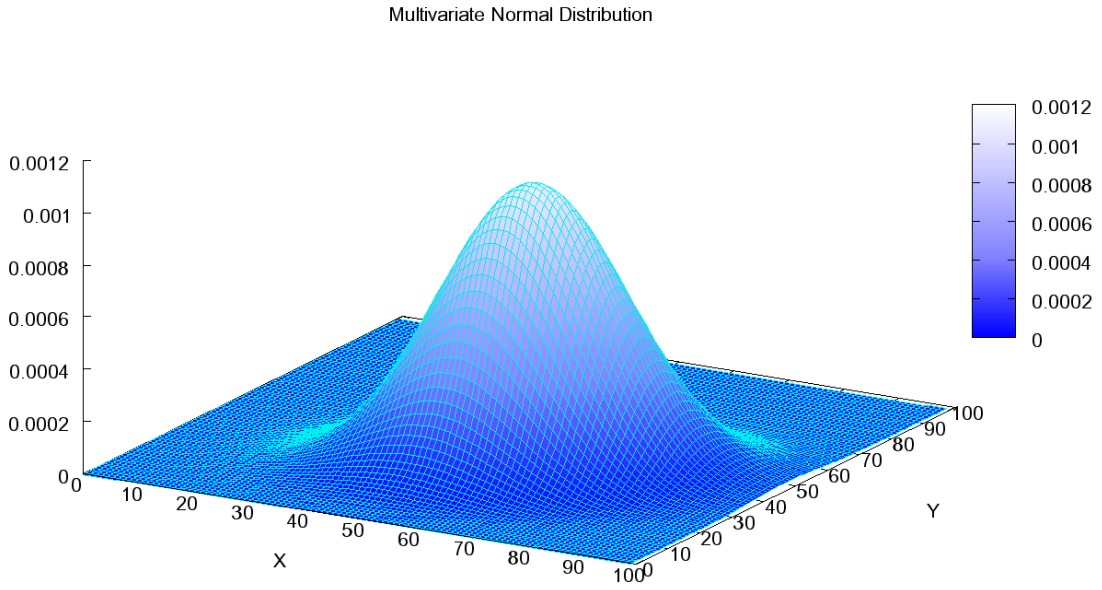
$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



- Parameter: $p(XY|C=1)$
- $p(XY|C=1) = p(X|C=1)p(Y|C=1)$

Gaussian/Normal Distribution - Multivariate

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$



μ: d dimensional mean vector
Σ: k×k covariance matrix
|Σ|: determinant of Σ

μ: (50,50)
Σ: $\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$

- Parameter: $p(XY|C=1)$

Gaussian Learning - univariate

X
1
3
4
5
6
7
9

Assuming that the dataset follow a normal distribution,

Dataset is described by the normal distribution PDF

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

Objective of Learning: estimate parameters (μ, σ)

ML Estimation method

X
1
3
4
5
6
7
9

data set X is i.i.d

$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu,\sigma^2\right)$$

Taking log

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

Partial derivation

Maximizing it with respect to μ

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Maximizing it with respect to σ^2

$$\sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{\mathrm{ML}})^2$$

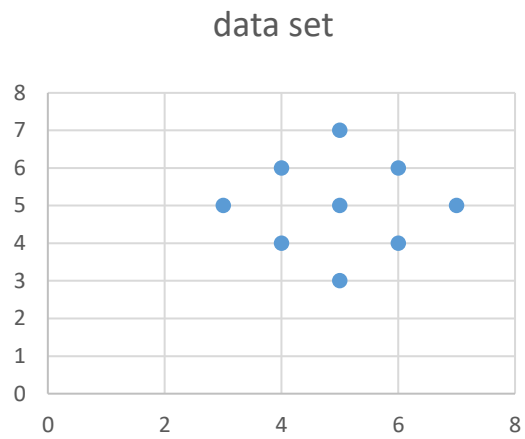
(μ, σ^2) =

Gaussian Learning - multivariate

X1	X2	X1-μ1	X2-μ2
6	6	1	1
3	5	-2	0
4	4	-1	-1
5	5	0	0
6	4	1	-1
7	5	2	0
4	6	-1	1
5	7	0	2
5	3	0	-2

MLE

μ: (5,5) Σ: $\begin{bmatrix} 1.33 & 0 \\ 0 & 1.33 \end{bmatrix}$



covariance: $\text{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - E(X))(y_i - E(Y))$

	X1	X2
X1	cov(X1,X1)	cov(X1,X2)
X2	cov(X2,X1)	cov(X2,X2)

	X1	X2
X1	1.33	0
X2	0	1.33

X	Y	p(c=1|(x,y))	p(c=0|(x,y))
5	8		
4	7		
8	9		
6	8		
8	2		
7	1		
5	2		

K-Means like scheme

1. Randomly select $u_k, \sigma_k, \ \pi_k \ (k = 1..K)$
2. Calculate $p(c = k|(xy))$
3. Update $u_k, \sigma_k, \ \pi_k$
4. Repeat step 2 until convergence

$$p(c = 1|(5,8)) \propto p((5,8)|c = 1) \times p(c = 1) = N((5,8)|u_1, \sigma_1) \times \pi_1$$

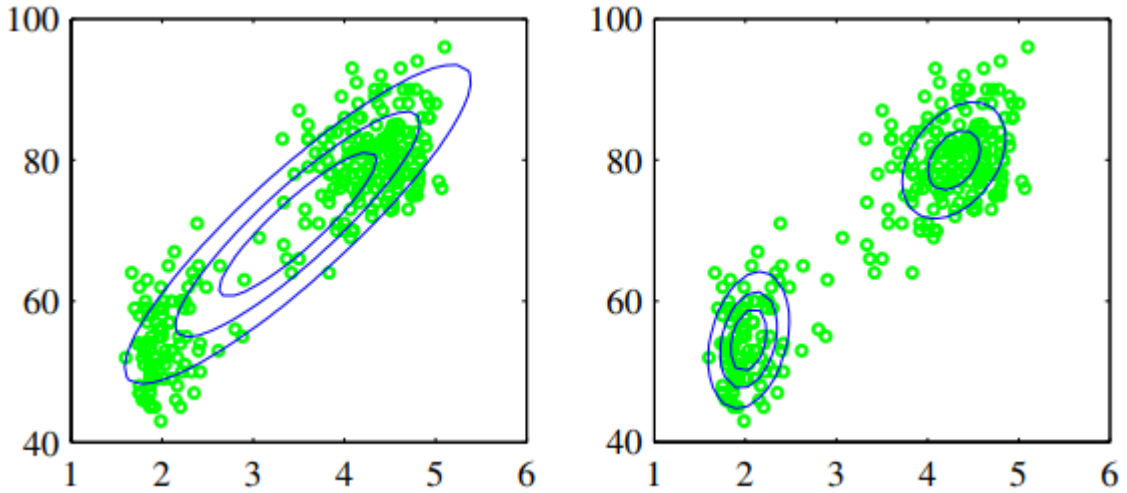
GMM → EM

MLE

- Review MLE Method
  - Step1: likelihood function $\mathrm{p}(\mathrm{dataset}|\theta)$
  - Step2: taking log of $\mathrm{p}(\mathrm{dataset}|\theta)$
  - Step3: taking partial derivative wrt. θ, and equate to zero

  - Question
    - For a supervised problem, $\mathrm{p}(\mathrm{x}|\theta 1)$ if x is in $\mathrm{C1}$ (the class label is known)
    - For a unsupervised problem: ?

Gaussian Mixture Model – Motivation



Single Gaussian distribution which has been fitted to (learnt from) the data using maximum likelihood.

fails to capture the two clusters in the data

The distribution is given by a linear combination of two Gaussians

Partitioning Algorithms

- **K-means**
  - **hard assignment**: each object belongs to only one cluster

$$\theta_i \in \{\theta_1, \ldots, \theta_K\}$$

- **Mixture modeling**
  - **soft assignment**: probability that an object belongs to a cluster

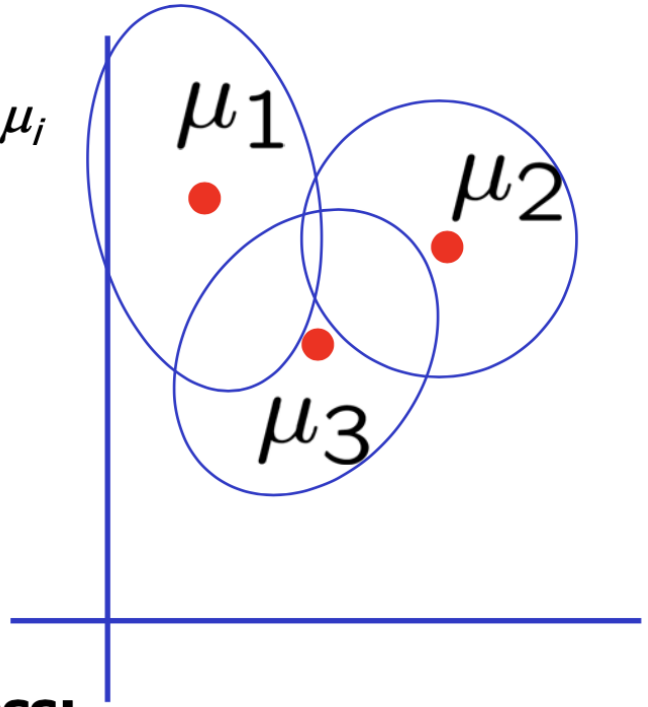
$$(\pi_1, \ldots, \pi_K),\ \pi_i \geq 0,\ \sum_{i=1}^{K} \pi_i = 1$$

Gaussian Mixture Model

Mixture of K Gaussians distributions: (Multi-modal distribution)

- There are K components
- Component *i* has an associated mean vector μ_i

Component *i* generates data from $N(\mu_i, \Sigma_i)$



Each data point is generated using this process:

1) Choose component i with probability $\pi_i = P(y = i)$
2) Datapoint $x \sim N(\mu_i, \Sigma_i)$

Gaussian Mixture Model cont…

Mixture of K Gaussians distributions: (Multi-modal distribution)

Hidden variable

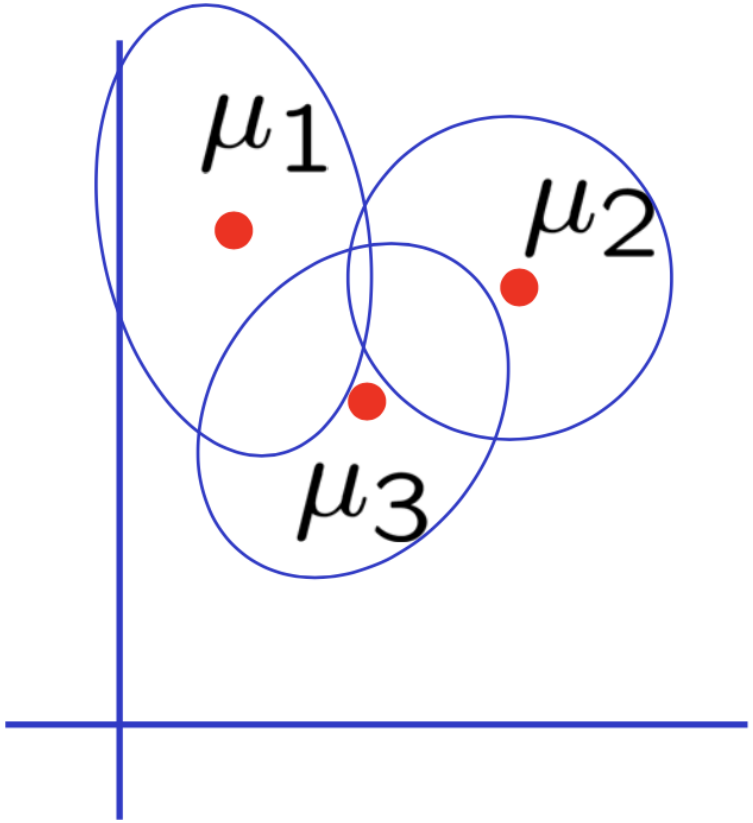
$$p(x|y=i) = N(\mu_i, \Sigma_i)$$

$$p(x) = \sum_{i=1}^{K} p(x|y=i)P(y=i)$$

Observed data — $p(x)$

Mixture component — $p(x|y=i)$

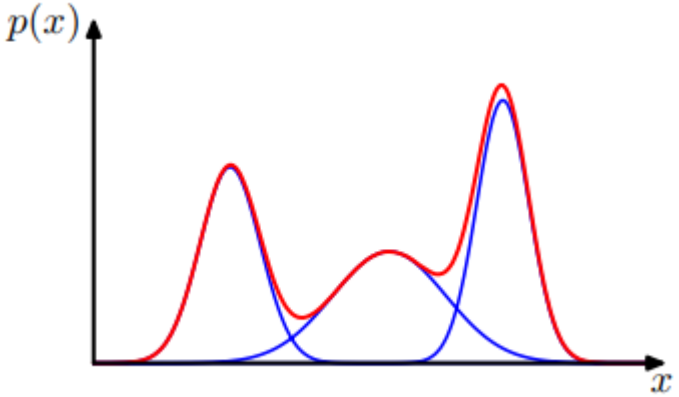
Mixture proportion — $P(y=i)$



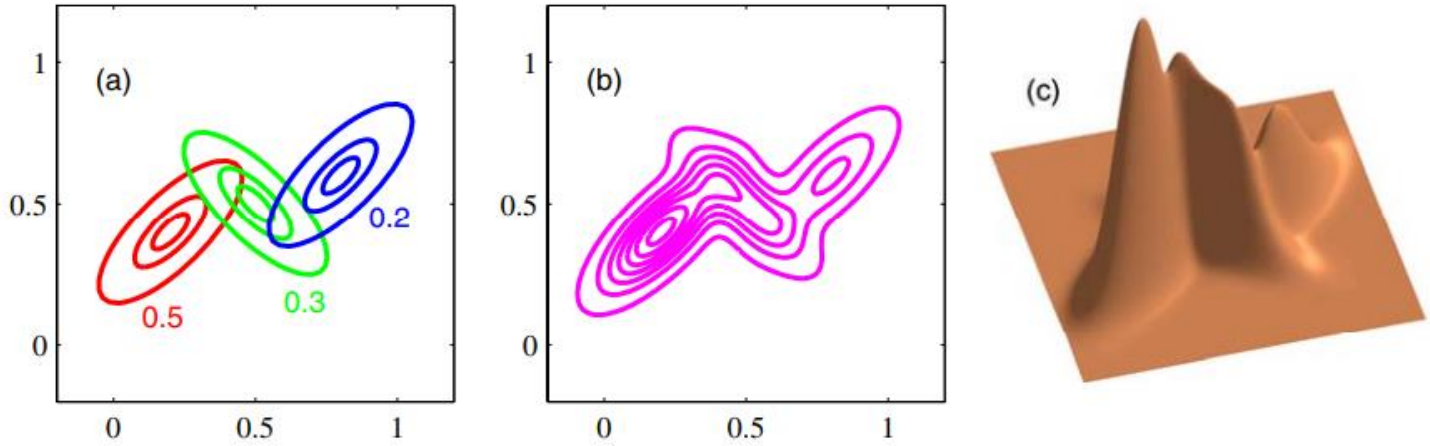
Gaussian Mixture Model

For a sample x, $p(x|\theta) = \sum_{k=1}^{K} p(x|c=k)p(c=k)$

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



one dimension GMM
three Gaussians (each scaled by a coefficient) in blue and their sum in red



two dimension GMM
three Gaussians with coefficient

Log-likelihood

- For each sample x which follow a GMM

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

- The log-likelihood for the dataset
  - K: number of classes
  - N: number of samples

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

Taking derivative w.r.t. (μ, Σ, π) and equate to zero

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

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

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

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

$P(C = k|X_n)$

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

However, it is not a closed-form solution

e.g. to calculate μ_k, we need to know the values of all other parameters

Solution: iterative scheme

- Try to find a correct solution
  - Maximizing the log likelihood function for a GMM turns out to be a more complex problem than for the case of a single Gaussian.
  - The difficulty arises from the presence of the summation over k that appears inside the logarithm.
- Go back and observe the result of setting derivatives of the log likelihood to zero
  - The results do suggest a simple iterative scheme for finding a solution

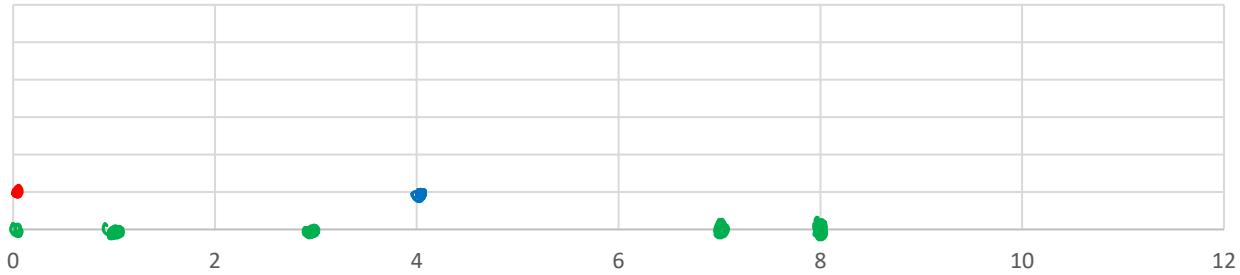
EM - 1-d Example - Estep

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

↑ 3.14

X	N(x|μ1,σ1)	N(x|μ2,σ2)	P(C=1|X)	P(C=2|X)
0	0.40	0.0001	1	0
1	0.24	0.0044	0.98 $= \frac{0.5 \times 0.24}{0.5 \times 0.24 + 0.5 \times 0.0044}$	0.02
3	0.004	0.242	0.02	0.98
7	0	0.0044	0	1
8	0	0.0001	0	1

$\mu1=0,\sigma1=1,\pi1=0.5$
$\mu2=4,\sigma2=1,\pi2=0.5$



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

EM - 1-d Example - Mstep

X	P(C=1|X)	P(C=2|X)
0	1	0
1	0.98	0.02
3	0.02	0.98
7	0	1
8	0	1
Nk	2	3

$\pi_1 = \frac{2}{5}$

$\mu1=1.04/2=0.52, \sigma1=1.16/2=0.58, \pi1=0.4$
$\mu2=17.96/3=5.99, \sigma2=14.32/3=4.77, \pi2=0.6$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} z_{nk}\ \mathbf{x}_n \qquad \pi_k = \frac{N_k}{N}$$

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

$\mu1=0, \sigma1=1, \pi1=0.5$
$\mu2=4, \sigma2=1, \pi2=0.5$

EM - 1-d Example - Mstep

$\pi_1 = \frac{2}{5}$

X	P(C=1|X)	P(C=2|X)
0	1	0
1	0.98	0.02
3	0.02	0.98
7	0	1
8	0	1
Nk	2	3

$\mu1=1.04/2=0.52, \sigma1=0.62/2=0.31, \pi1=0.4$
$\mu2=17.96/3=5.99, \sigma2=26.16/3=8.72, \pi2=0.6$

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} z_{nk}\ \mathbf{x}_n \qquad \pi_k = \frac{N_k}{N}$$

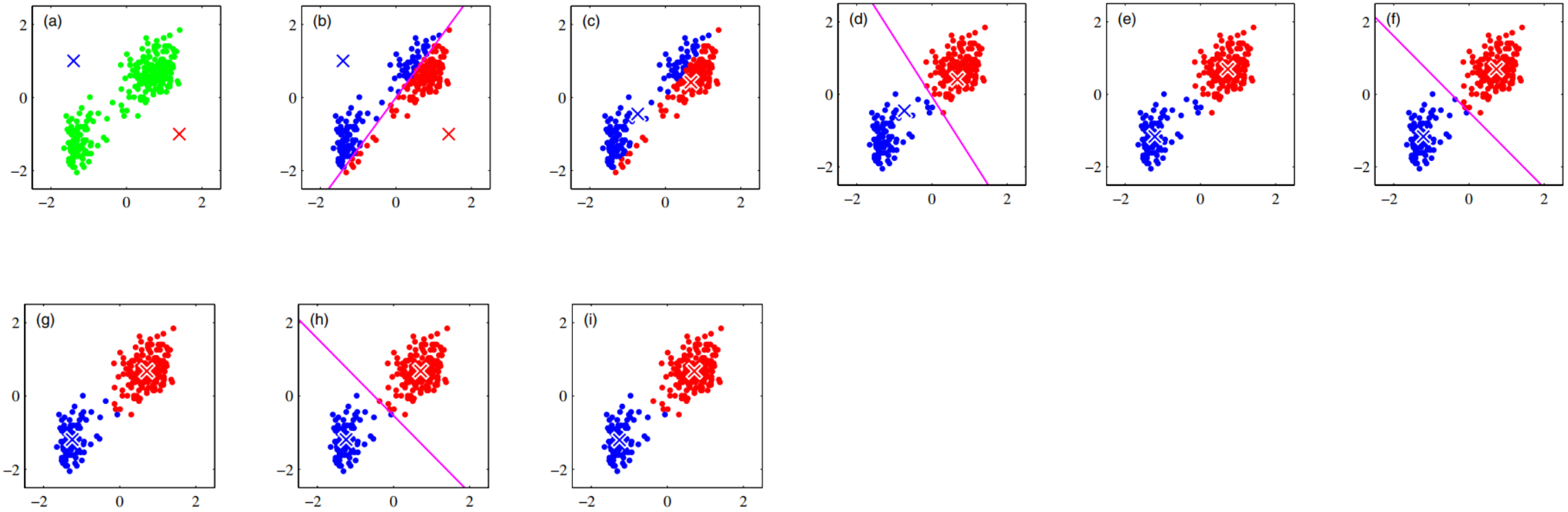
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N} z_{nk}\ (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

$\mu1=0, \sigma1=1, \pi1=0.5$
$\mu2=4, \sigma2=1, \pi2=0.5$

$$u_1 = \frac{0\times 1 + 1\times 0.98 + 3\times 0.02 + 7\times 0 + 8\times 0}{2} = 0.52$$

$$\sigma_2 = \frac{0\times(0-4)^2 + 0.02\times(1-4)^2 + 0.98\times(3-4)^2 + 1\times(7-4)^2 + 1\times(8-4)^2}{3}$$

K-means

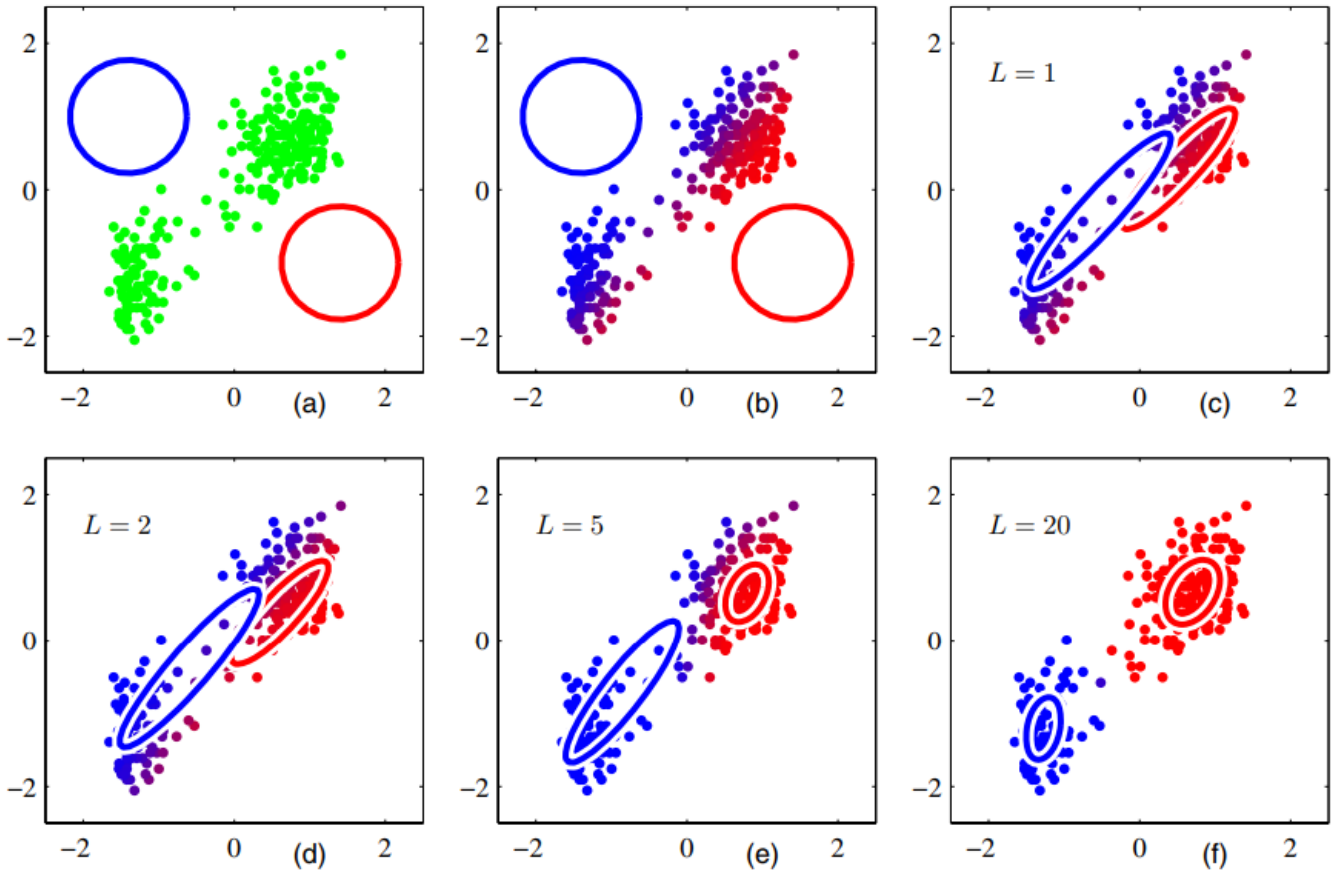


EM with K-means-like iteration

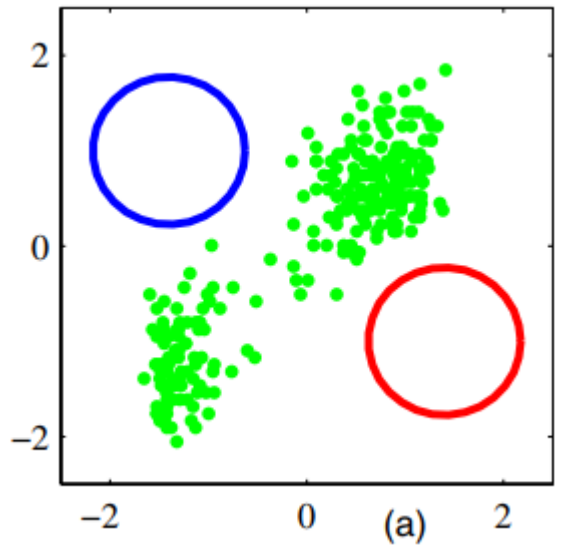
K=2

2 Gaussians

L: number of iteration



initialization

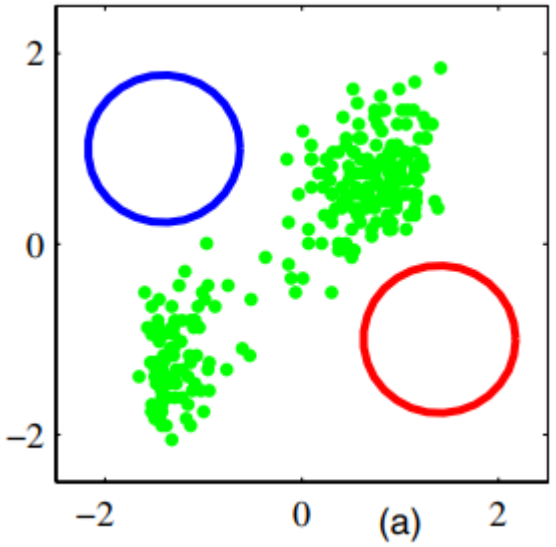


- How many parameters?
- K is set as 2 → 2 clusters
- Each cluster described by a single Gaussian
- Each Gaussian has two parameters μ and Σ
- Each object described by a GMM, and the prior of cluster is needed.

- 6 or 5 parameters

Initialization - example

- P(C=blue)=0.6, then P(C=red)=0.4
- Cluster Blue:
  - μ=(-1.5,1.5)
  - Σ: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Cluster Red:
  - μ=(1.5,-1)
  - Σ: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

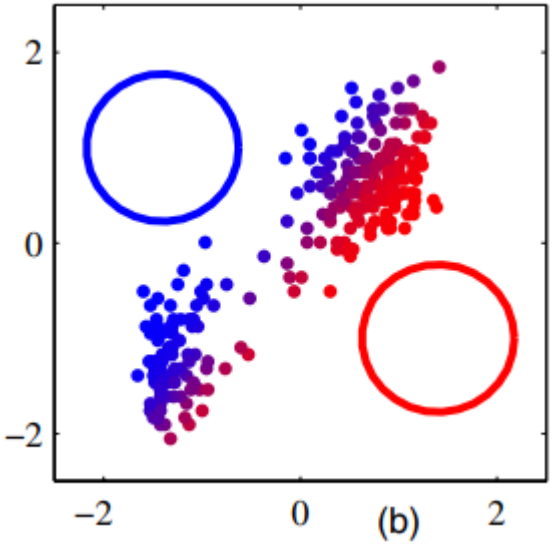


E-step

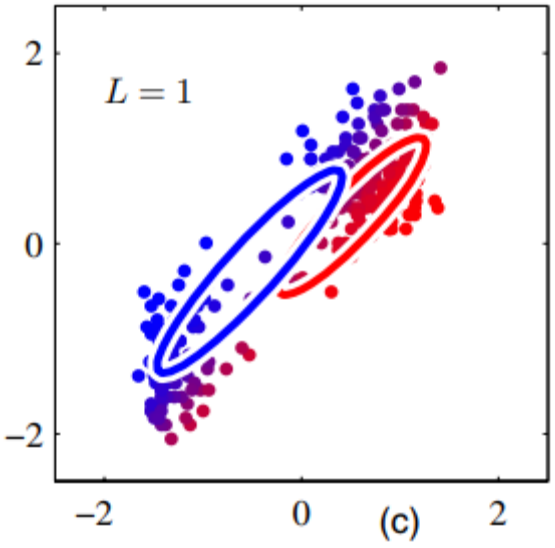
- For each object X(green dot), calculate p(X) using current values for the parameters
  - $p(C = Blue|X) = \frac{p(X|C=Blue)p(C=Blue)}{p(X|C=Blue)p(C=Blue)+p(X|C=Red)p(C=Red)}$

$$= \frac{\mathrm{N}(X|\mu_{blue},\Sigma_{blue})p(C=Blue)}{\mathrm{N}(X|\mu_{blue},\Sigma_{blue})p(C=Blue)+\mathrm{N}(X|\mu_{red},\Sigma_{red})p(C=Red)}$$

X1	X2	P(C=Blue|X)	P(C=Red|X)
0.6	1.6	0.8	0.2
-1.3	1.5	0.72	0.28
-0.44	0.4	0.1	0.9
1.5	-1.5	0.5	0.5
…	…		



M-step – Expectation Maximization



- Re-estimate the parameters by Expectation Maximization
  - $\mu_{\text{blue}}^{\text{new}} = \frac{1}{\text{N}_{blue}} \sum_{n=1}^{N} p(C = Blue|X_n) X_n$
  - $\Sigma_{\text{blue}}^{\text{new}} = \frac{1}{\text{N}_{blue}} \sum_{n=1}^{N} p(C = Blue|X_n)(X_n - \mu_{\text{blue}})\,(X_n - \mu_{\text{blue}})^{\text{T}}$
  - $p(c = blue) = \frac{N_{blue}}{N}$

X1	X2	P(C=Blue|X)	P(C=Red|X)
0.6	1.6	0.8	0.2
-1.3	1.5	0.72	0.28
-0.44	0.4	0.1	0.9
1.5	-1.5	0.5	0.5
…	…		

$$N_{blue} = \sum_{n=1}^{N} p(C = Blue|X_n)$$

Evaluate the log likelihood

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k)$$

$$\sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} p(C = k)p(X_n|C = k)\}$$

$$\sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} p(C = k)N(X_n|\mu_k, \Sigma_k)\}$$

X1	X2	P(C=Blue|X)	P(C=Red|X)
0.6	1.6	0.8	0.2
-1.3	1.5	0.72	0.28
-0.44	0.4	0.1	0.9
1.5	-1.5	0.5	0.5
…	…		

Termination: Check for convergence of either the parameters or the log likelihood

If the convergence criterion is not satisfied, go to E-step

EM and missing data

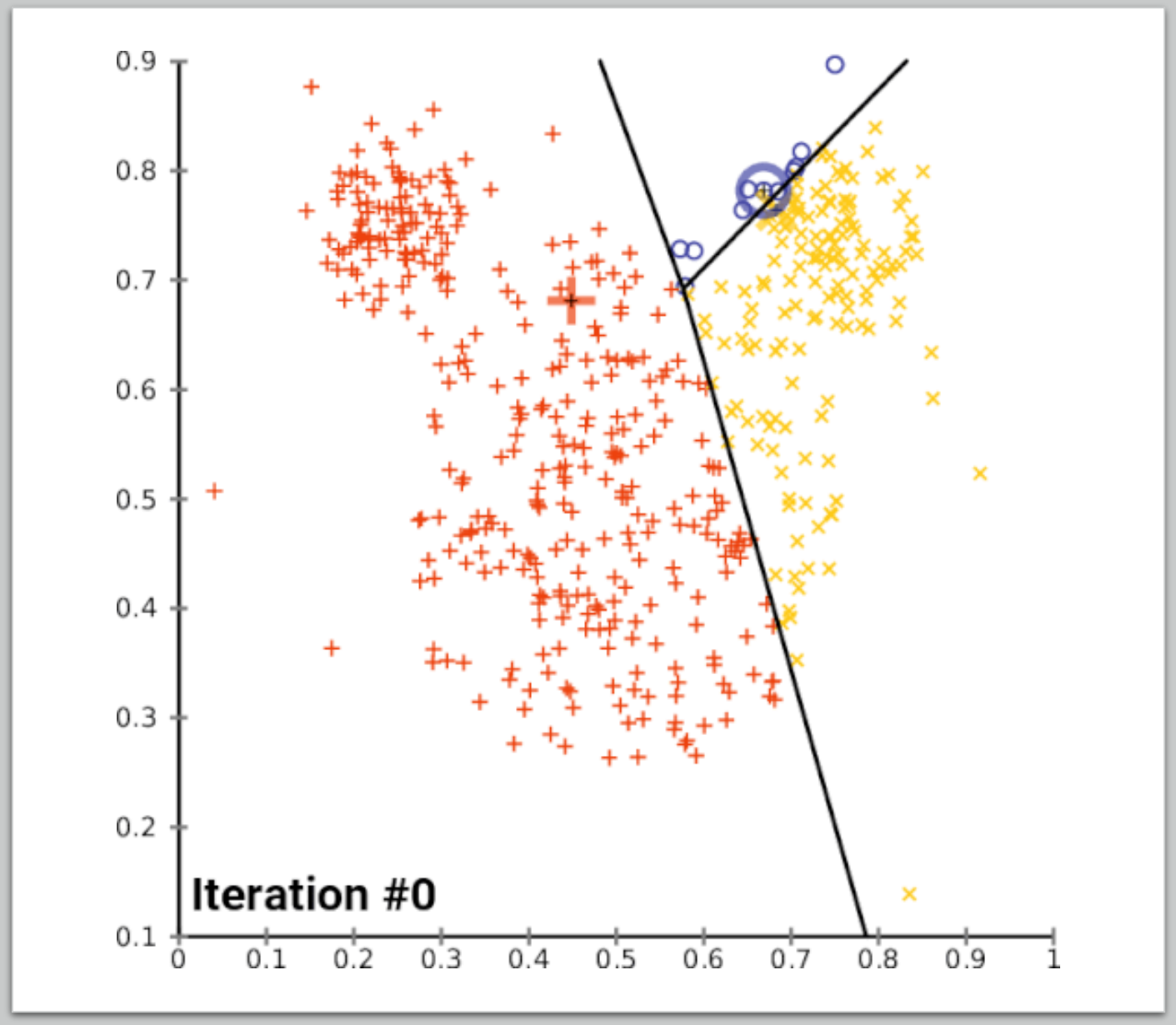
- EM is a general framework for partially observed data
  - "Complete data" xi, zi – features and assignments
  - Assignments zi are missing (unobserved)

- EM corresponds to
  - Computing the distribution over all zi given the parameters
  - Maximizing the "expected complete" log likelihood
  - GMMs = plug in "soft assignments", but not always so easy

- Alternatives: Stochastic EM, Hard EM
  - Instead of expectations, just sample the zi or choose best (often easier)
  - Called "imputing" the values of z
  - Hard EM: similar to EM, but less "smooth", more local minima
  - Stochastic EM: similar to EM, but with extra randomness
    - Not obvious when it has converged

GMM VS Kmeans
Kmeans

1. Choose the number of clusters *K*
2. Initialize the vector **μ_k** that defines a central point of each cluster
3. Assign each data point **x** to the closest cluster centre
4. Recalculate central points **μ_k** for each cluster
5. Repeat 3–4 until central points stop moving



GMM vs Kmeans
Kmeans

- The K-Means algorithm will converge but it might not be a global minimum. To avoid a situation where it converges to a local minimum, K-Means should be re-run a few times with different parameters.

- K-Means performs hard assignment which means that each datapoint has to belong to a certain class and there is no probability assigned to each datapoint.

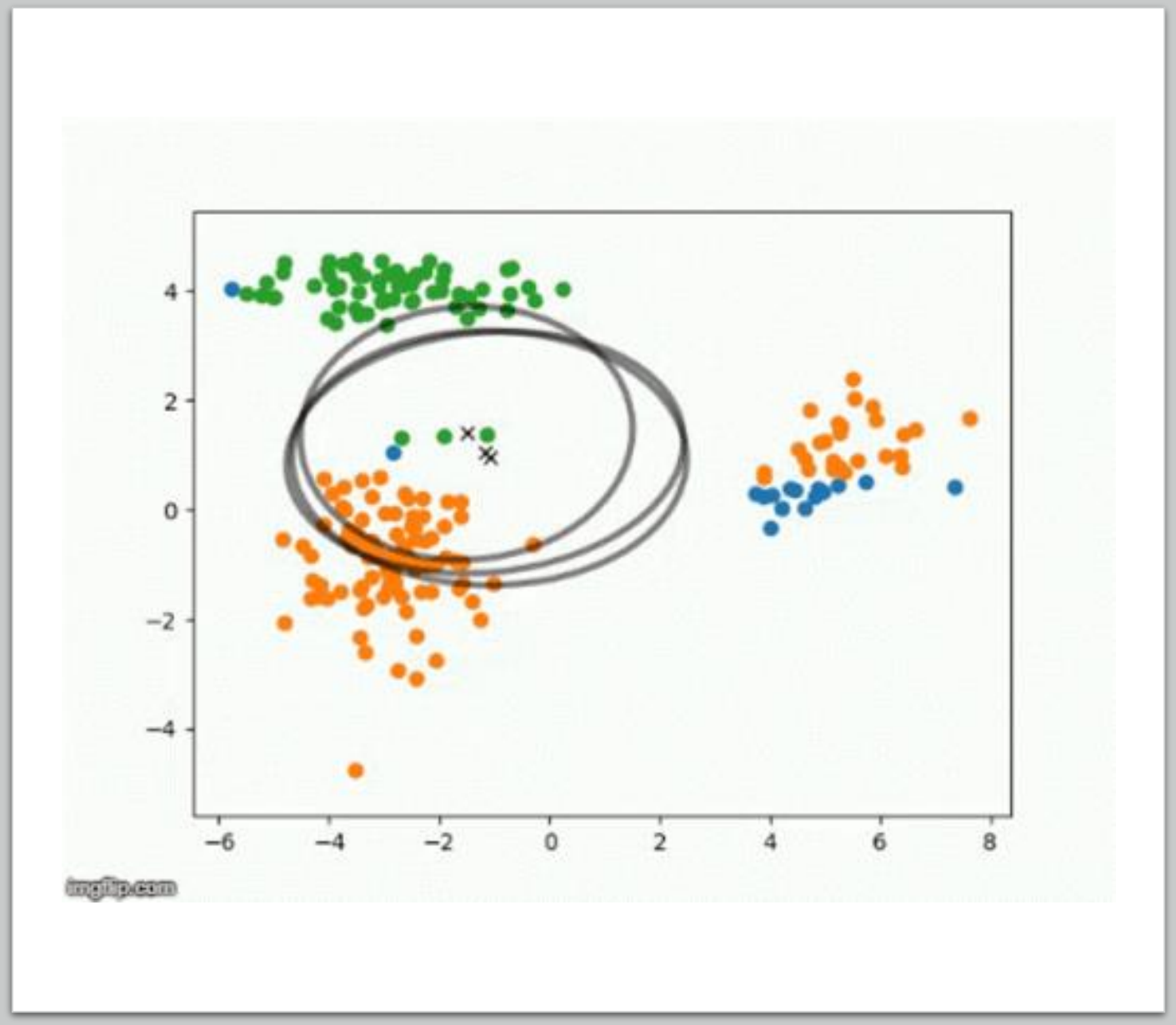
GMM vs Kmeans
GMM

- Gaussian Mixtures are based on *K* independent Gaussian distributions that are used to model *K* separate clusters. As a reminder, the multivariate Gaussian distribution is given as:

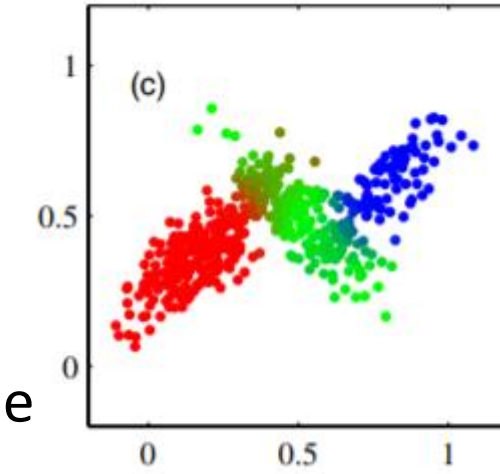
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$

GMM vs Kmeans GMM

1. Initialize **μ, ∑,** and mixing coefficient **π** and evaluate the initial value of the log likelihood L
2. Evaluate the responsibility function using current parameters
3. Obtain new **μ, ∑,** and **π** using newly obtained responsibilities
4. Compute the log likelihood L again. Repeat steps 2–3 until the convergence.



GMM vs Kmeans



- Decision Boundaries:
  - GMM: More flexible and with a covariance matrix, we can make the boundaries elliptical
  - Kmeans: Circular boundaries
- Probability:
  - GMM: show how strong is our belief that a given datapoint belongs to a specific cluster
  - Kmeans: 1 or 0

  GMs usually tend to be slower than K-Means because it takes more iterations of the EM algorithm to reach the convergence. They can also quickly converge to a local minimum that is not a very optimal solution.

  Solution: GMM with Kmeans Initializer

GMM vs Kmeans

- It is quite strange that a plain K-Means is slower than GM with a K-Means initializer. **Behind the hood**, Scikit-Learn seems to apply an optimized version of K-Means that takes fewer iterations to converge.
- If you look for robustness, GM with K-Means initializer seems to be the best option. K-Means should be theoretically faster if you experiment with different parameters, but as we can see from the computation plot above, GM with K-Means initializer is the fastest. GM on its own is not much of use because it converges too fast to a non-optimal solution for this dataset.

Different cluster analysis results on "mouse" data set: