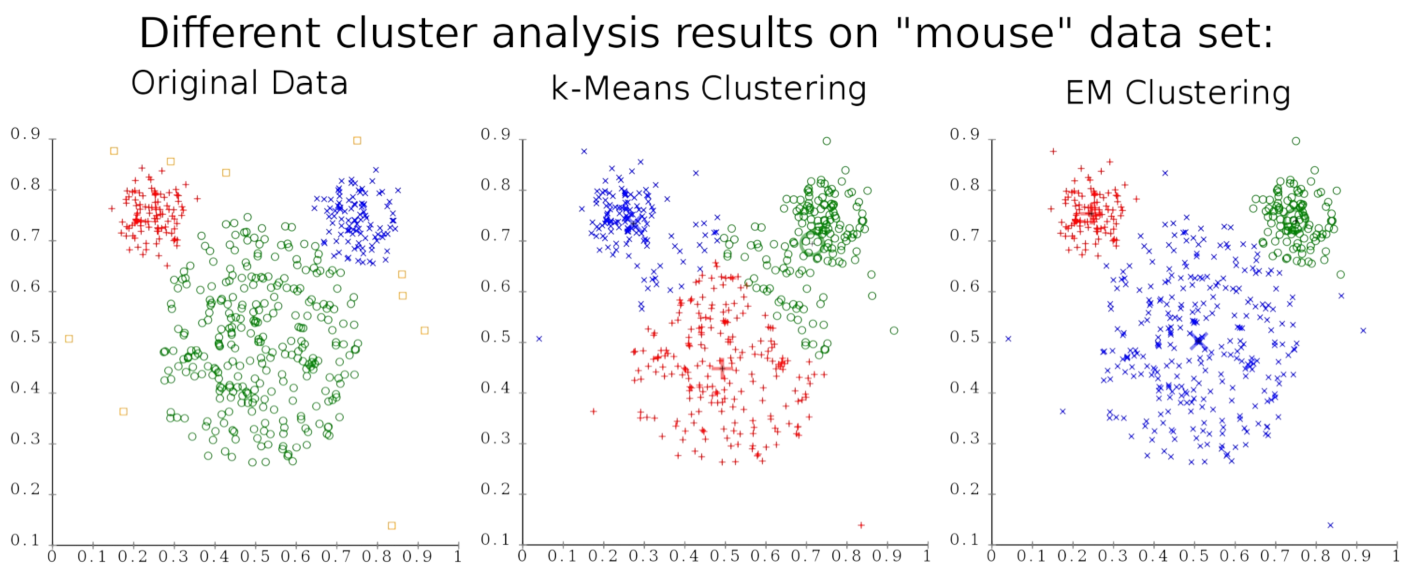
Lecture 17 – The Expectation-Maximization Algorithm

In some settings, the k-means algorithm does not give desirable results:



The issue is that k-means does not account for the size of the different clusters.

Alternative approach: assume $\underline{x}_i$'s are drawn at random from a mixture of Gaussians distribution and cluster using the Expectation Maximization (EM) algorithm.

A Gaussian distribution is a probability distribution that characterizes how likely different values of a random variable are.

$$f(\underline{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\underline{x}-\underline{\mu})^T \Sigma^{-1}(\underline{x}-\underline{\mu})\right\} \quad \left(\text{shorthand } \mathcal{N}(\underline{\mu}, \Sigma)\right)$$

$|\Sigma|$ = matrix determinant

= product of singular values

$\underline{\mu}$ = mean = $\mathbb{E}\,\underline{X}$ = expected (average) value of $\underline{X}$

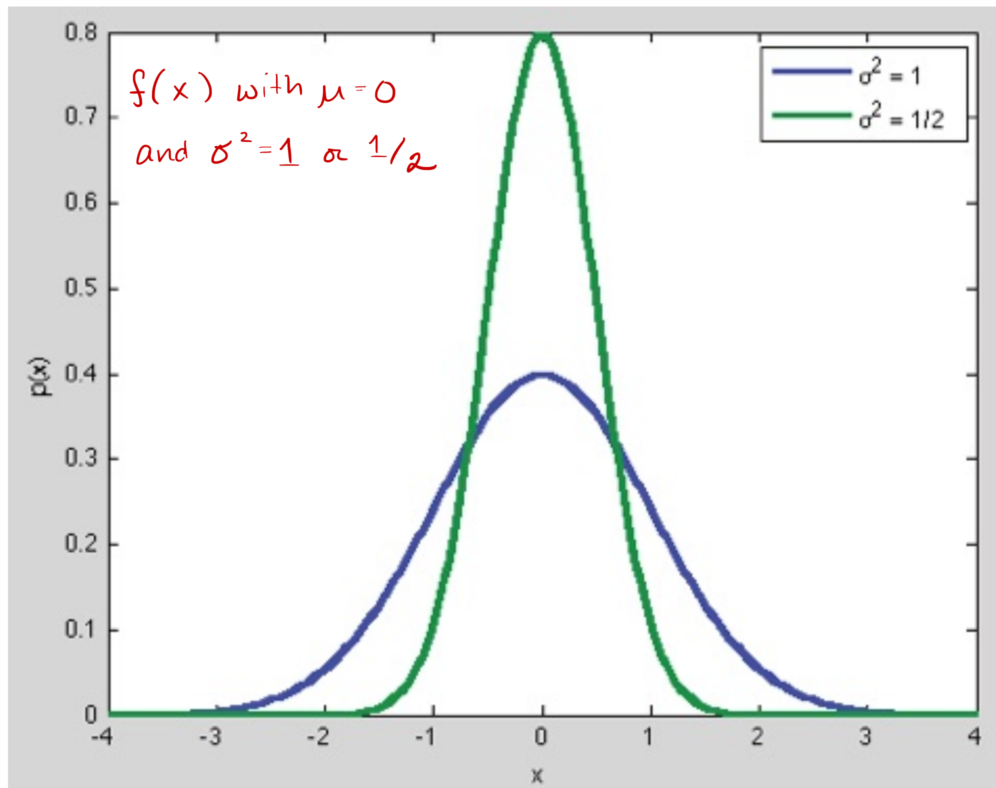
Σ = covariance = $\mathbb{E}\left[(\underline{x}-\underline{\mu})(\underline{x}-\underline{\mu})^T\right] \iff \Sigma_{ij} = \mathbb{E}\left[(x_i-\mu_i)(x_j-\mu_j)\right]$

Ex 1 : $p = 1$

$x \sim \mathcal{N}(\mu, \sigma^2)$

(i.e. $\Sigma = \sigma^2$)

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$



Ex. $p = 2$

Σ

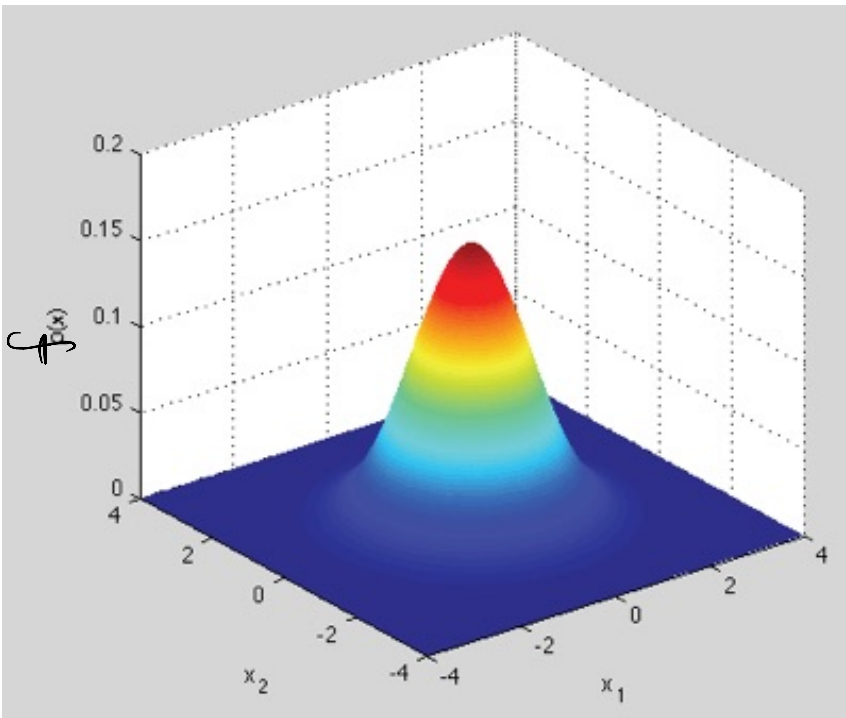
$\underline{x} \sim \mathcal{N}(\underline{\mu}, \sigma^2 I)$

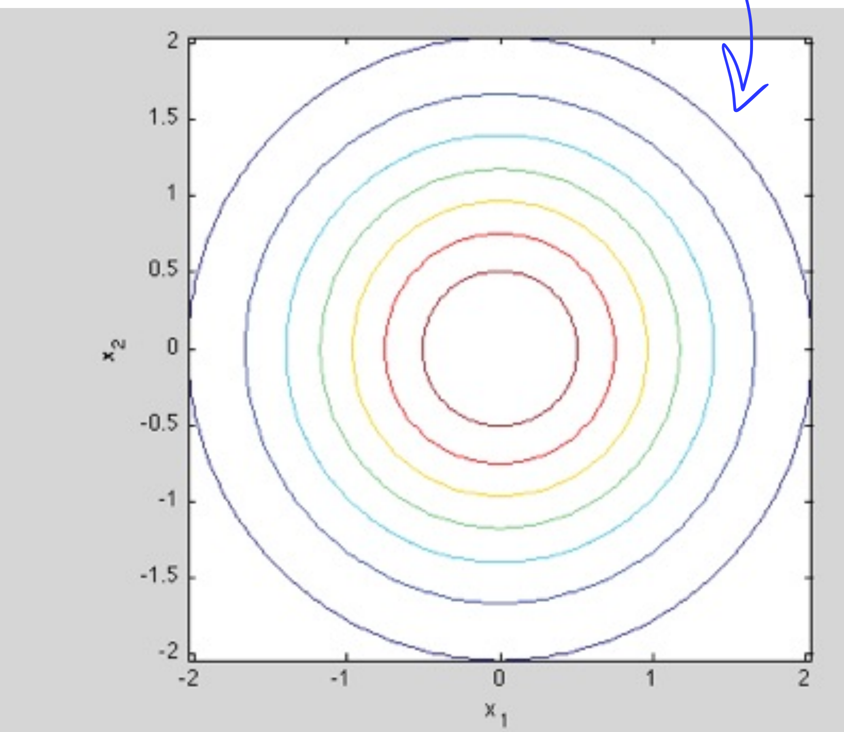
$\begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$

$$f(\underline{x}) = \gamma \iff \frac{(x_1 - \mu_1)^2}{\sigma^2} + \frac{(x_2 - \mu_2)^2}{\sigma^2} = \frac{\|\underline{x} - \underline{\mu}\|_2^2}{\sigma^2} = \gamma'$$

Contour plot. each circle is set of all $\underline{x}$ such that $f(\underline{x}) = \gamma$ for some γ

$=$ set of $\underline{x}$ such that $\|\underline{x} - \underline{\mu}\|^2 = \gamma'$





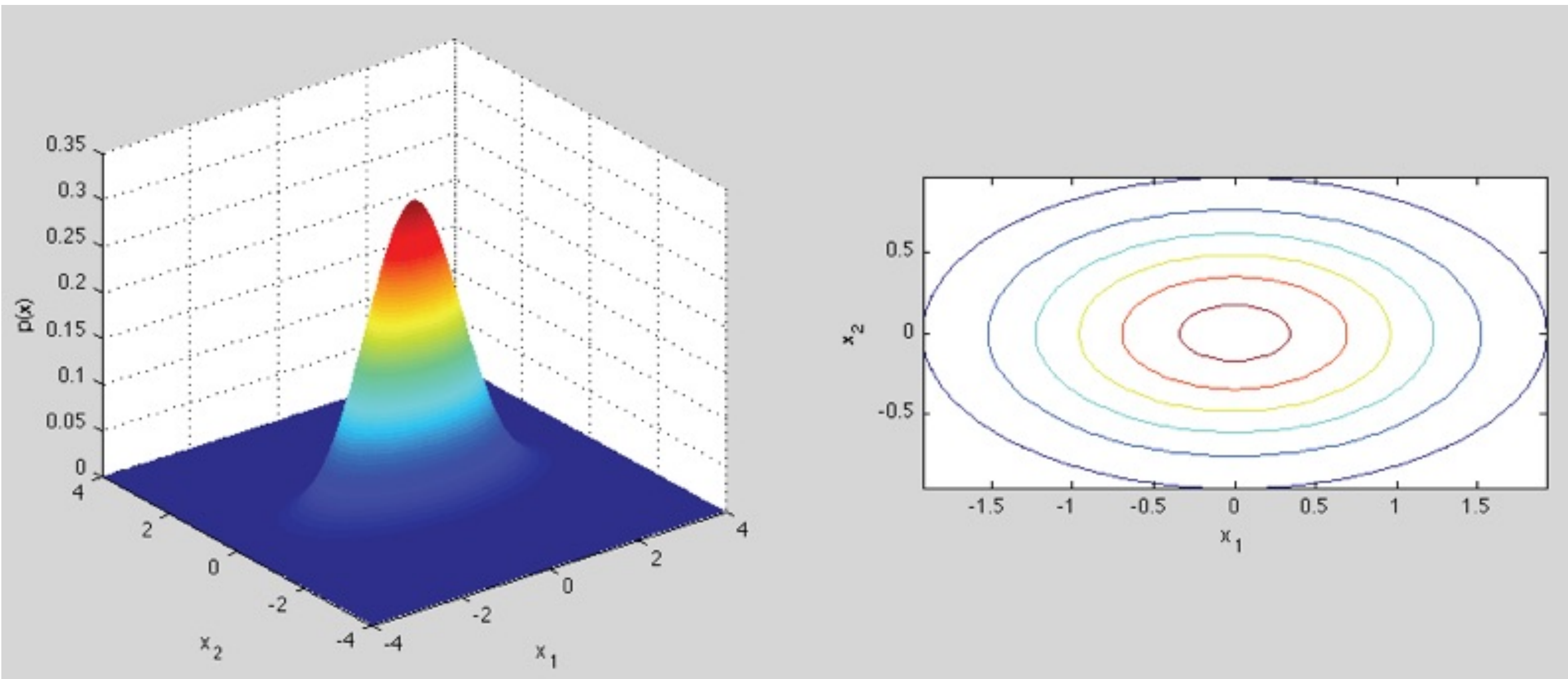
Ex. $p=2$.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

$$\underline{x} \sim N(\underline{\mu}, \Sigma)$$

Here density contours are ellipses whose axes align with the coordinate axes. Note:

$$f(\underline{x}) = \gamma \iff \frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} = \gamma'$$



Ex. Σ is an arbitrary positive-definite matrix.

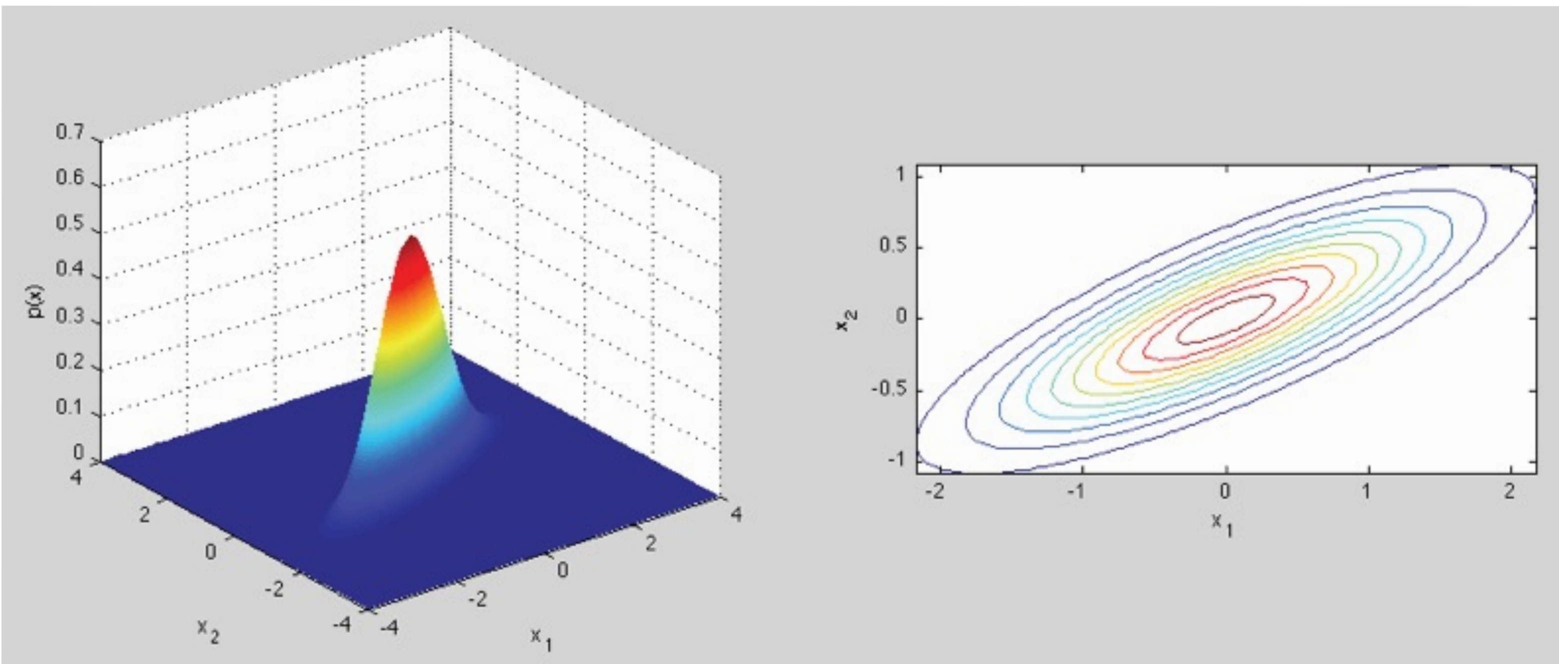
Let $\Sigma = V \Lambda V^T$ (eigenvalue decomposition)

Let $\underline{x}' = V^T \underline{x}$ and $\underline{\mu}' = V^T \underline{\mu}$. $\Leftarrow$ $\underline{x}' = x$ in rotated coordinate system defined by colums of V.

Then contour = set of all $\underline{x}$ s.t. $(\underline{x} - \underline{\mu})' \Sigma^{-1} (\underline{x} - \underline{\mu}) = \gamma$ for some γ

$$(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) = (\underline{x} - \underline{\mu})^T V \Lambda^{-1} V^T (\underline{x} - \underline{\mu}) = (\underline{x}' - \underline{\mu}')^T \Lambda^{-1} (\underline{x}' - \underline{\mu}') = \frac{(x_1' - \mu_1')^2}{\lambda_1} + \frac{(x_2' - \mu_2')^2}{\lambda_2}$$

= elipse in rotated coordinate system, where V defines rotation.



The SVD of the covariance matrix Σ tells us how the distribution is "oriented" and spread out in different directions

Maximum likelihood estimation

Given $\underline{x}_i \sim \mathcal{N}(\mu, \Sigma)$ for $i = 1, 2, \dots, n$, we may wish to estimate $\underline{\mu}$ and Σ.

Maximum likelihood strategy: Choose $\hat{\underline{\mu}}$ and $\hat{\Sigma}$ to maximize the likelihood of the $\underline{x}_i$'s

$$(\hat{\underline{\mu}}, \hat{\Sigma}) = \underset{\underline{\mu}, \Sigma}{\operatorname{argmax}} \prod_{i=1}^{n} f(\underline{x}_i ; \mu, \Sigma)$$

$$= \underset{\underline{\mu}, \Sigma}{\operatorname{argmax}} \log\left(\prod_{i=1}^{n} f(\underline{x}_i ; \underline{\mu}, \Sigma)\right)$$

$$= \underset{\underline{\mu}, \Sigma}{\operatorname{argmin}} -\log\left(\prod_{i=1}^{n} f(\underline{x}_i ; \underline{\mu}, \Sigma)\right)$$

$$= \underset{\underline{\mu}, \Sigma}{\operatorname{argmin}} \sum_{i=1}^{n} -\log f(\underline{x}_i ; \underline{\mu}, \Sigma)$$

$$= \underset{\underline{\mu}, \Sigma}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{1}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(\underline{x}_i - \underline{\mu})^T \Sigma^{-1}(\underline{x} - \underline{\mu})$$

Compute gradients, set to zero $\Longrightarrow$

$$\hat{\underline{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \underline{x}_i \, , \qquad \hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} (\underline{x}_i - \hat{\underline{\mu}})(\underline{x}_i - \hat{\underline{\mu}})^T$$

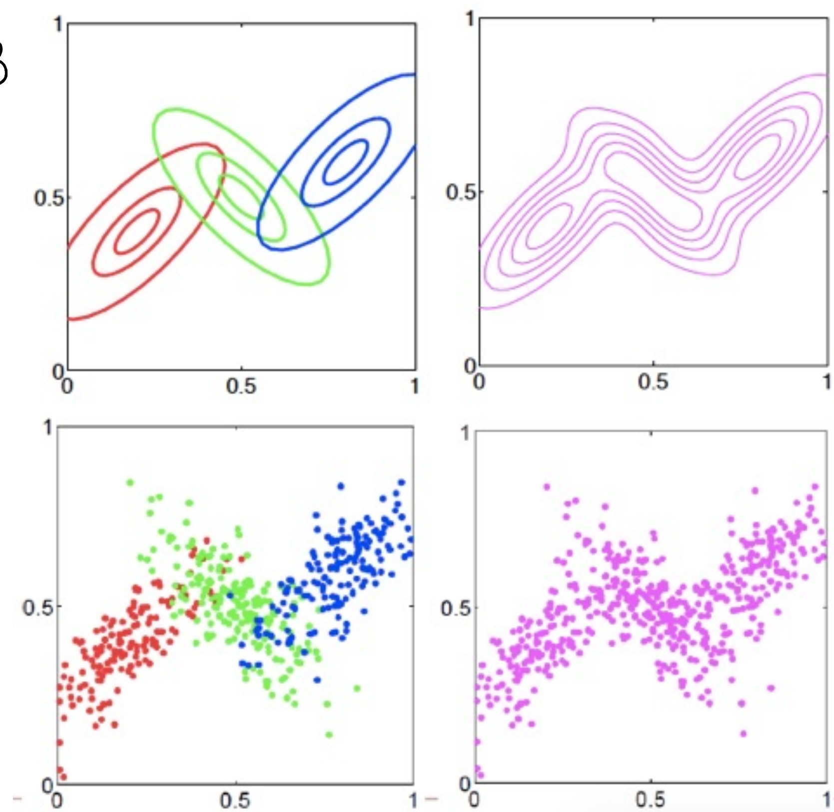
Back to clustering

Assume there are K Gaussians (each will correspond to a different cluster) with means $\underline{\mu}_k$ and covariances Σ_k for $k = 1, 2, \dots, K$.

We model the observed $\underline{x}_i$'s drawn from a mixture of these Gaussians as follows.

- Choose one of the K Gaussians — ie. the k^{th} Gausian is chosen with probability π_k, where $\sum_{k=1}^{K} \pi_k = 1$; denote k_i
- draw $\underline{x}_i \sim \mathcal{N}(\underline{\mu}_{k_i}, \Sigma_{k_i})$

$K=3$



$$\Rightarrow f(\underline{x}_i) = \sum_{k=1}^{K} \pi_k \, f(\underline{x} ; \underline{\mu}_k, \Sigma_k)$$

Given $\underline{x}_i$'s, we want to cluster them <u>without</u> knowing $\underline{\mu}_k$'s or Σ_k's or π_k's.

if we knew cluster membership k_i for each $\underline{x}_i$, then we could use maximum likelihood estimation to compute $\underline{\mu}_k$'s, Σ_k's, and π_k's. <u>Without</u> k_i's, maximum likelihood estimation is hard.

Expectation-Maximization Algorithm

- Initialize means $\hat{\mu}_k$, covariances $\hat{\Sigma}_k$, and mixture weights $\hat{\pi}_k$ for $k = 1, 2, \ldots, K$

- E-step: compute $p_k(\underline{x}_i)$ = Probability that $\underline{x}_i$ was drawn from k^{th} Gaussian given value of $\underline{x}_i$

$$= \Pr(k_i = k \mid \underline{x}_i) = \frac{\Pr(k_i = k)\, f(\underline{x} \mid k_i = k)}{f(\underline{x})} \qquad \text{(Bayes rule)}$$

$$\hat{p}_k(\underline{x}_i) = \frac{\hat{\pi}_k\, f(\underline{x}_i \mid \hat{\underline{\mu}}_k, \hat{\Sigma}_k)}{\sum_{j=1}^{K} \hat{\pi}_j\, f(\underline{x}_i \mid \hat{\underline{\mu}}_j, \hat{\Sigma}_j)} \qquad \text{for } k = 1, 2, \ldots, K; \quad i = 1, 2, \ldots, n$$

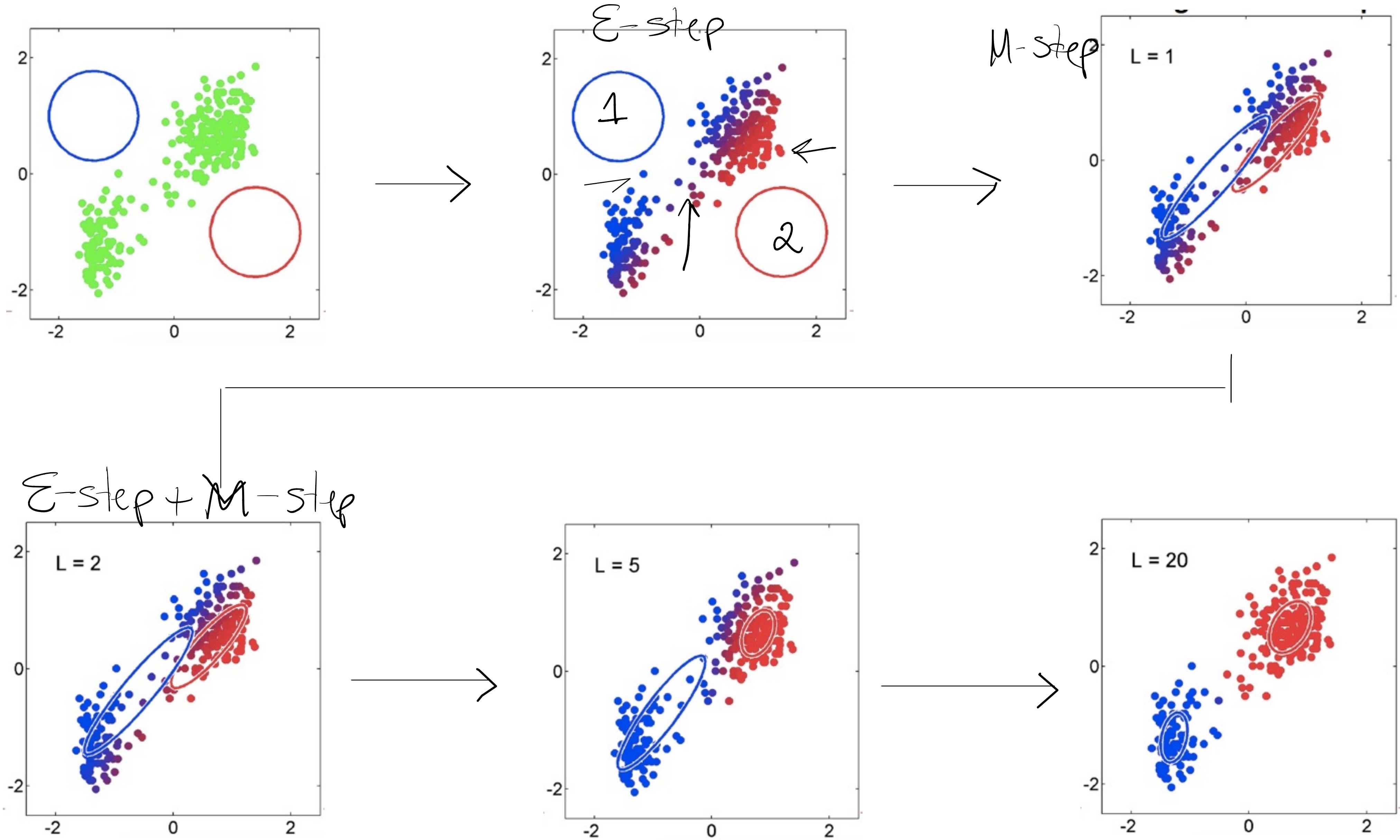
- M-step: using $\hat{p}_k(\underline{x}_i)$'s, update estimates of $\hat{\mu}_k$, $\hat{\Sigma}_k$, $\hat{\pi}_k$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{n} \hat{p}_k(\underline{x}_i)\, \underline{x}_i}{\sum_{j=1}^{n} \hat{p}_k(\underline{x}_j)}$$

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \hat{p}_k(\underline{x}_i)(\underline{x}_i - \hat{\underline{\mu}}_k)(\underline{x}_i - \hat{\underline{\mu}}_k)^T}{\sum_{j=1}^{n} \hat{p}_k(\underline{x}_j)}$$

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_k(\underline{x}_i)$$

- if not converged, return to E-step.



E-step
M-step
1
2
L = 1
E-step + M-step
L = 2
L = 5
L = 20