Chapter - 9 Bishop

K-means Clustering, GMM, EM

Chetan Arora

Two kinds of people in a group

Indian | European

170 cm, σ, =

180 cm, σ =

Person = 180 cm

=0 Ind
=1 Eur

Bays Rule

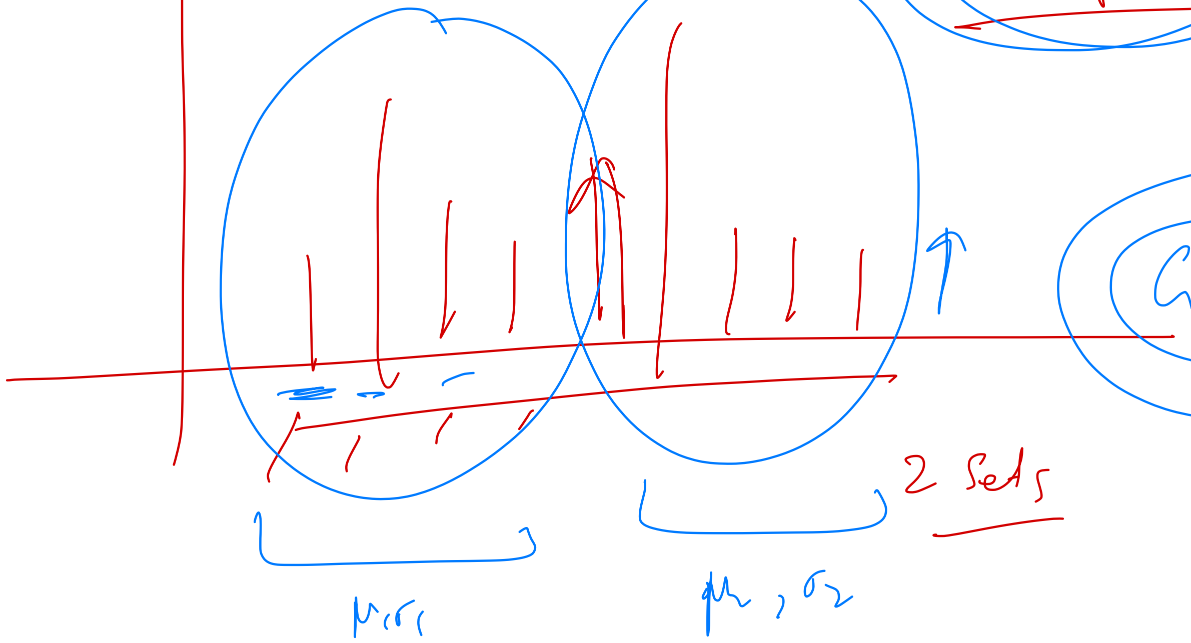
$$P(y/x) = \frac{P(x|y)\,P(y)}{P(x)}$$

$P(x|y=0)$ $P(x|y=1)$

Clustering

Unsupervised

Generative

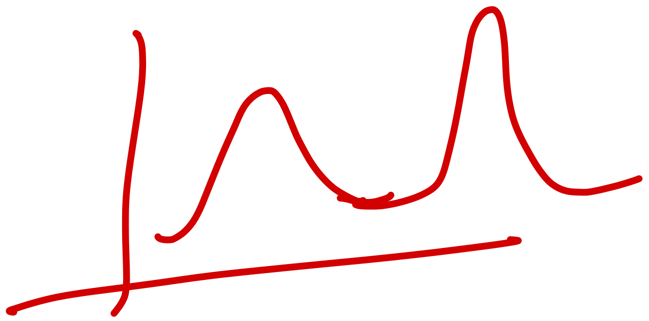


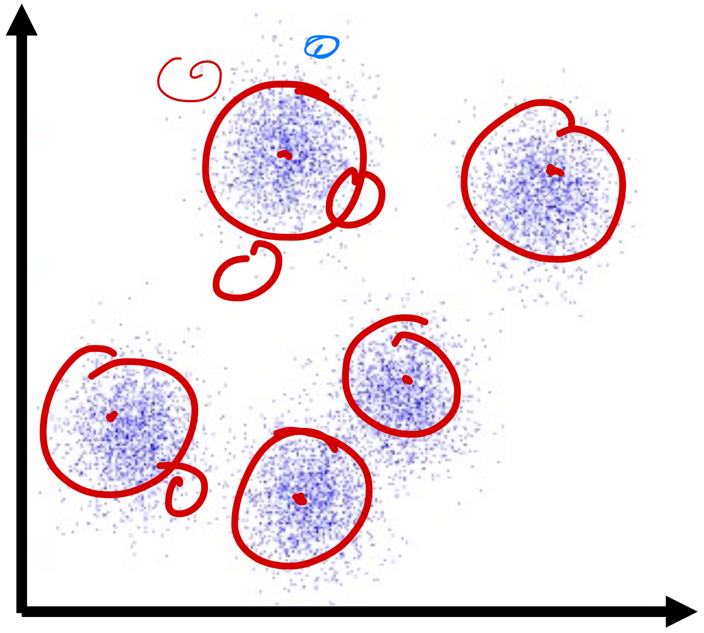
Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Clustering

- Sometimes the samples points form clusters, where samples within a cluster are similar to each other, and samples in different clusters are dissimilar.

- Such a distribution is multimodal, since it has multiple modes, or regions of high probability mass.





$P(x)$ $\hookrightarrow \mathbb{R}^d$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

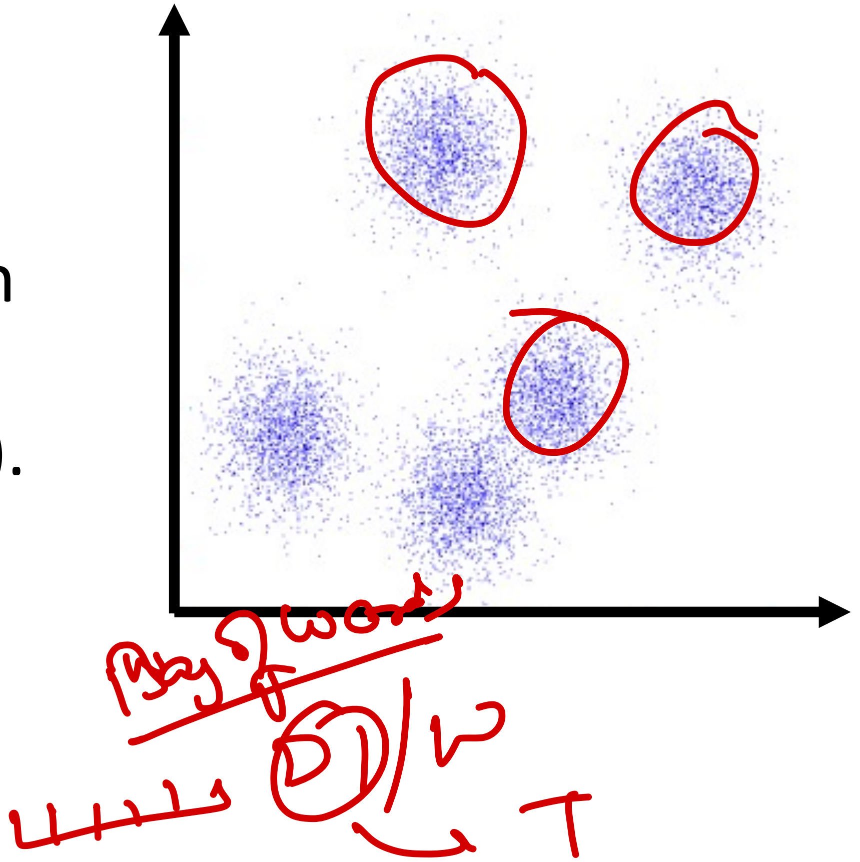
Clustering

- Grouping data points into clusters, with no observed labels, is called clustering. It is an unsupervised learning technique.

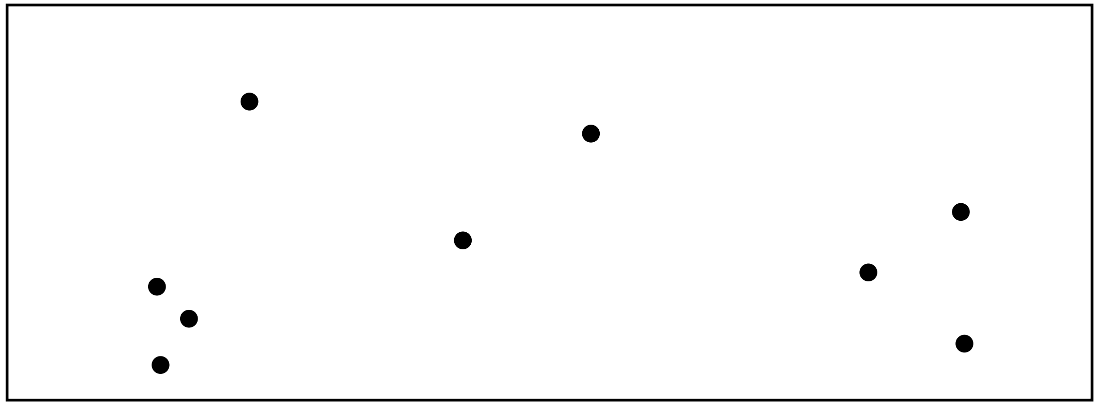
LDA

- E.g. clustering machine learning papers based on the topic (deep learning, Bayesian models, etc.)
  - But topics are never observed (unsupervised).

Topic models



Clustering Problem

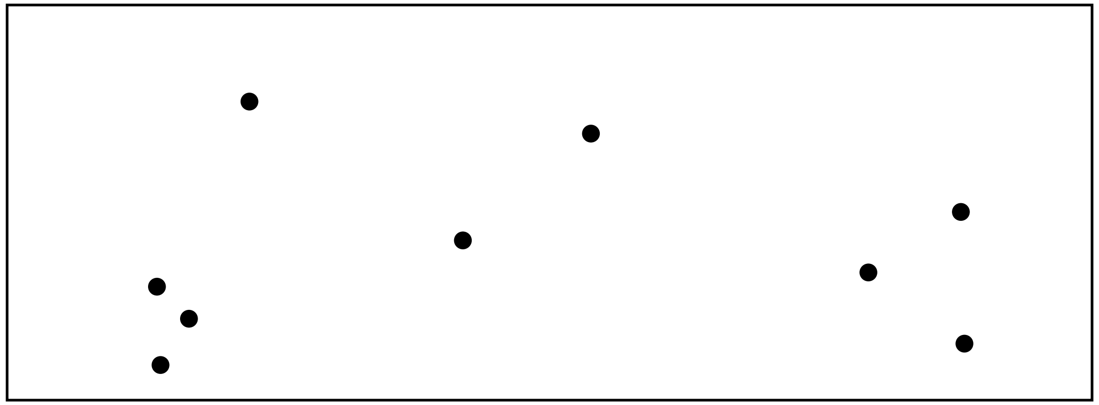


Assumptions:

- Data $\{\underline{x^{(1)}}, \ldots, \underline{x^{(N)}}\}$ lives in a Euclidean space, $x^{(n)} \in R^D$.
- Each data point belongs to one of K clusters
- Data points from same cluster are similar, i.e. close in Euclidean distance.

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Clustering Problem



- How can we identify the clusters?
  - data points that belong to each cluster.

- Formulate the problem as an optimization problem.

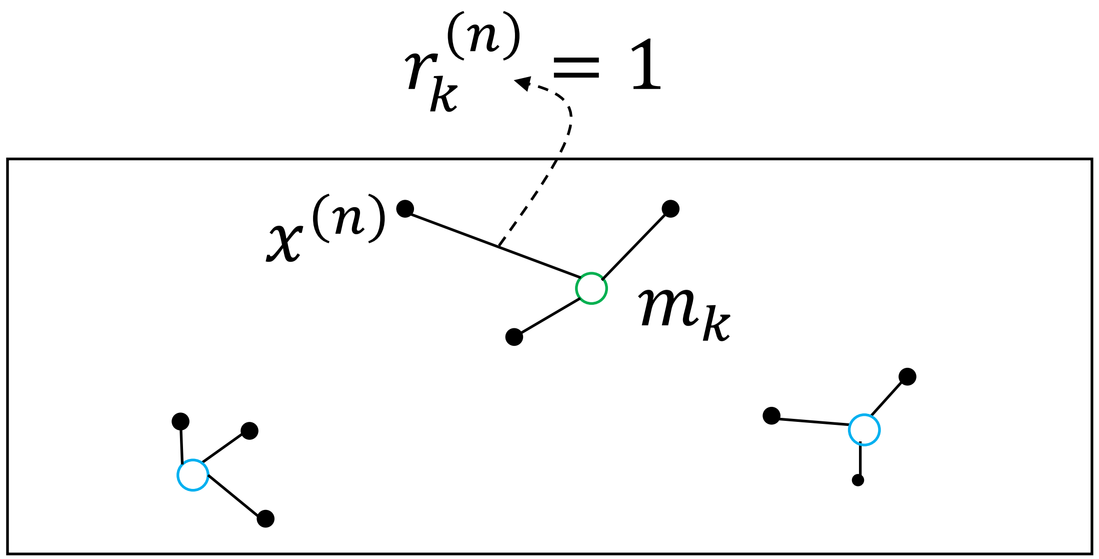
Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

K-means Objective

$\rightarrow \mu$

Find cluster centers $\{m_k\}_{k=1}^{K}$ and assignments $\{r^{(n)}\}_{n=1}^{N}$ to minimize the sum of squared distances of data points $\{x^{(n)}\}$ to their assigned centers.

responsibility



- Data sample $n = 1,..,N$: $x^{(n)} \in R^D$ (observed),
- Cluster center $k = 1,..,K$: $m_k \in R^D$ (not observed),
- Output: Cluster assignment for sample n: $r(n) \in R^K$
  - $r(n)$: 1-of-K encoding

$r^{(n)}$

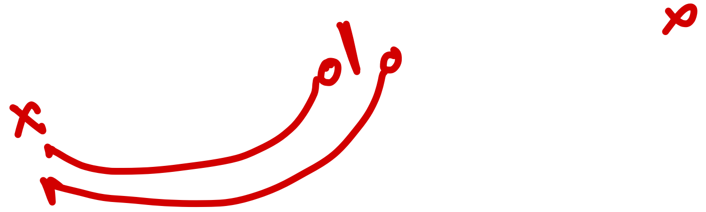
$r^{(n)}$

$r^{(n)}$

$\underbrace{0\,0\,0\,1\,0\,0\,0}_{K}$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

K-means Objective

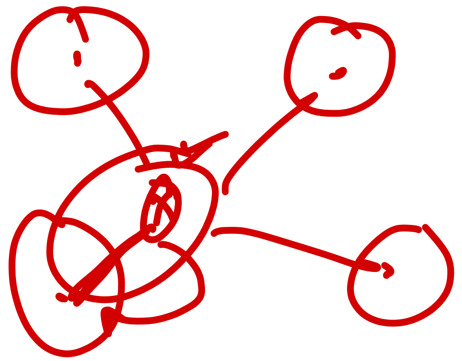


- Mathematically:

$$\min_{\{m_k\},\{r^n\}} J(\{m_k\},\{r^n\}) = \min_{\{m_k\},\{r^n\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} \left\|m_k - x^{(n)}\right\|^2$$

- Where $r_k^{(n)} = \mathbb{I}[x^{(n)}$ is assigned to cluster $k]$, i.e., $r^n = [0,..,1,..,0]^T$

- Finding an optimal solution is an NP-hard problem!



K-means Objective

- Optimization problem:

$$\min_{\{m_k\},\{r^n\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \left\| m_k - x^{(n)} \right\|^2$$

- Since $r_k^{(n)} = \mathbb{I}[x^{(n)}$ is assigned to cluster $k]$, i.e., $r^n = [0, . . , 1, . . , 0]^T$
- Hence, inner sum is over K terms but only one of them is non-zero.
- E.g. say sample $x^{(n)}$ is assigned to cluster $k = 3$, then $r^n = [0, 0, 1, 0, \dots]$

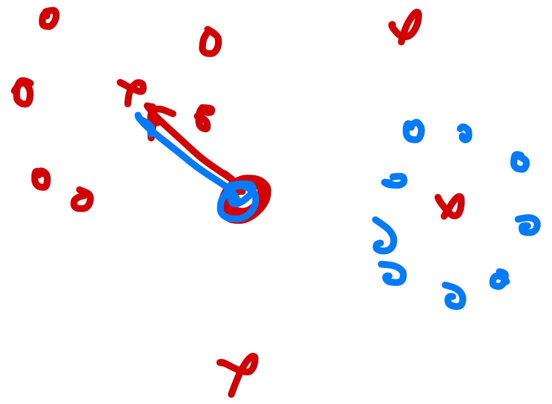
$$\sum_{k=1}^{K} r_k^{(n)} \left\| m_k - x^{(n)} \right\|^2 = \left\| m_3 - x^{(n)} \right\|^2$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

How to optimize?: Alternating Minimization

- Optimization problem:

$$\min_{\{m_k\},\{r^n\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \left\|m_k - x^{(n)}\right\|^2$$



- Problem is hard when minimizing jointly over the parameters $\{m_k\}, \{r^{(n)}\}$
- But if we fix one and minimize over the other, then it becomes easy.
- Doesn't guarantee the same/optimal solution!

Local optimum

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Alternating Minimization: Finding Assignment

- Optimization problem:

$$\min_{\{m_k\},\{r^n\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \left\| m_k - x^{(n)} \right\|^2$$

- If we fix the centers $\underline{\{m_k\}}$ then we can easily find the optimal assignments $\underline{\{r^{(n)}\}}$

- For each sample n

$$\min_{r^{(n)}} \sum_{k=1}^{K} r_k^{(n)} \left\| m_k - x^{(n)} \right\|^2$$

Alternating Minimization : Finding Assignment

- For each sample n

$$\min_{r^{(n)}} \sum_{k=1}^{K} r_k^{(n)} \left|\left|m_k - x^{(n)}\right|\right|^2$$

- Assign each point to the cluster with the nearest center

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

- E.g. if $x(n)$ is assigned to cluster $\hat{k}$: $r(n) \; = \; [0, 0, \dots, 1, \dots, 0]^T$

Alternating Minimization: Finding Centroids

- Optimization problem:

$$\min_{\{m_k\},\{r^n\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \left\| m_k - x^{(n)} \right\|^2$$

- Likewise, if we fix the assignments $\{r^{(n)}\}$ then can easily find optimal centers $\{m_k\}$ by solving for the following for $l = 1, 2, \ldots, K$:

$$\frac{\partial}{\partial m_l} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \, 2 \left\| m_k - x^{(n)} \right\|^{\cancel{2}} = 0$$

Alternating Minimization: Finding Centroids

- For $l = 1, 2, \ldots, K$

$$\frac{\partial}{\partial m_l} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \left\| m_k - x^{(n)} \right\|^2 = 0$$

$$\Rightarrow 2 \sum_{n=1}^{N} r_l^{(n)} \left(m_l - x^{(n)} \right) = 0 \qquad \Rightarrow m_l = \underline{\frac{\sum_n r_l^{(n)} x^{(n)}}{\sum_n r_l^{(n)}}}$$

<u>Centroid</u>

- Set each cluster's center to the average of its assigned data points

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Alternating Minimization: Overall Optimization

- Alternate between minimizing $J(\{m_k\}, \{r^{(n)}\})$ with respect to $\{m_k\}$ and with respect to $\{r^{(n)}\}$

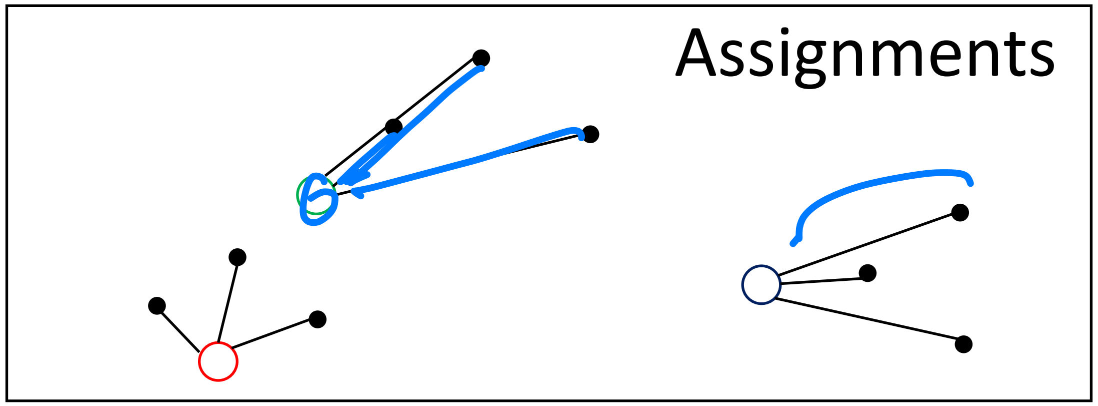
- This is called **alternating minimization**

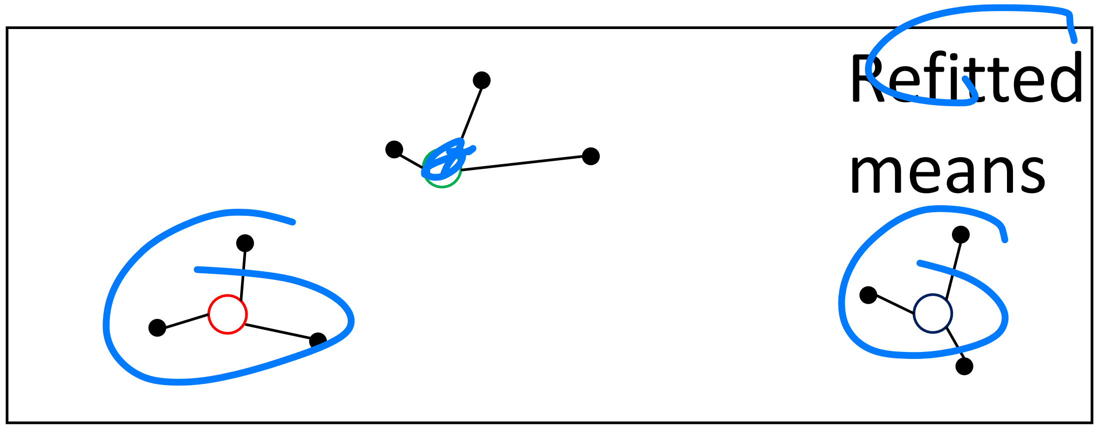
$$\begin{tikzcd}\underline{m} \arrow[r, phantom, "|"] & \underline{r} \arrow[l, bend left]\end{tikzcd}$$

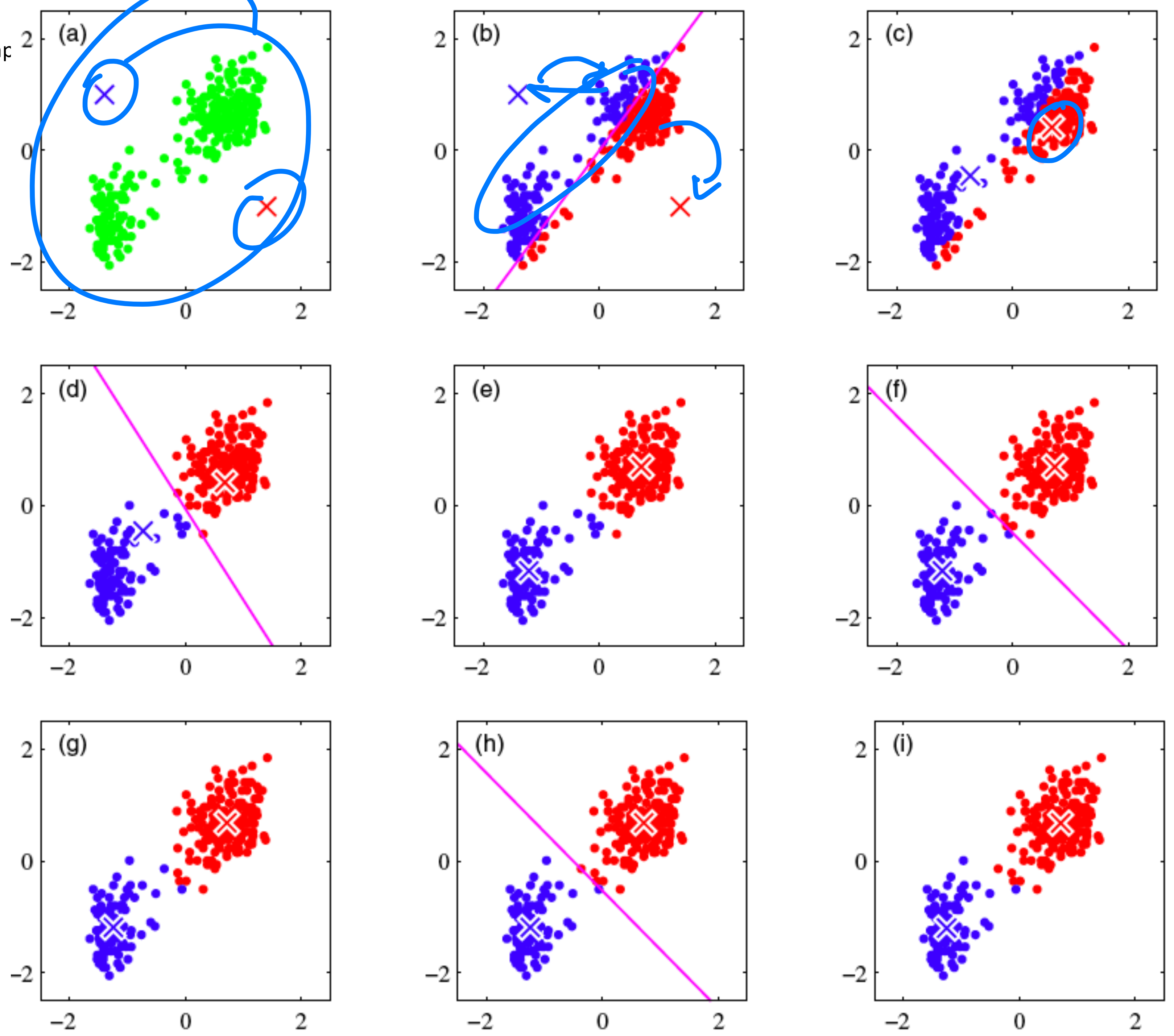
Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

K-means Algorithm

- **Initialization:** randomly initialize cluster centers K
- Alternate between two steps: r
  - **Assignment step:** Assign each data point to the closest cluster
  - **Refitting step**: Move each cluster center to the mean of the data assigned to it.






(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

The K-means Algorithm

- Initialization: Set K cluster means $m_1, \ldots, m_K$ to random values
- Repeat until convergence (until assignments do not change):
    - **Assignment:** Optimize $J\ w.r.t. \{r\}$
        - Each data point $x^{(n)}$ assigned to nearest center

$$\hat{k}^{(n)} = \underset{k}{\operatorname{argmin}} \left|\left|m_k - x^{(n)}\right|\right|^2$$

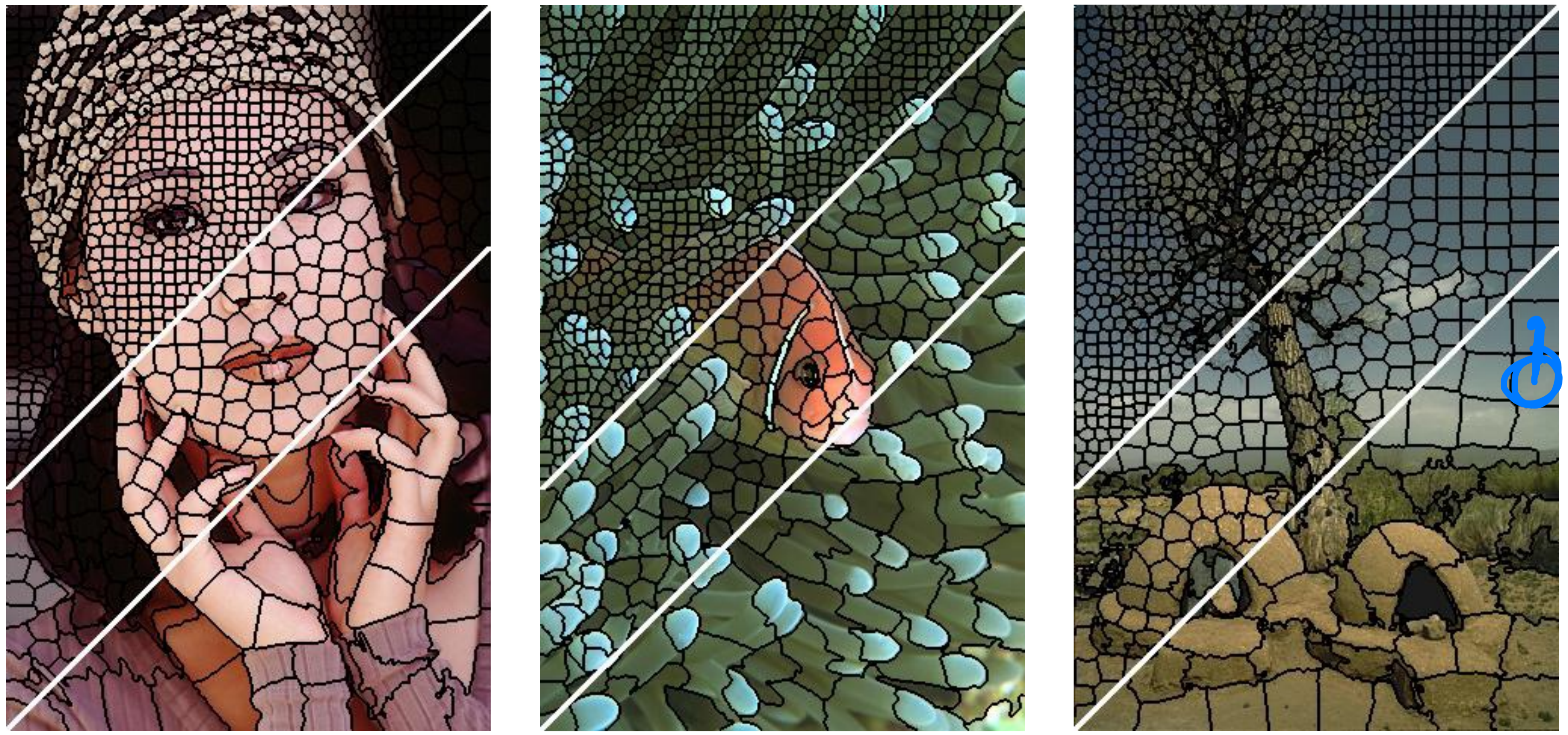
        - And responsibilities (1-hot or 1-of-K encoding) assigned to

$$r_k^{(n)} = \mathbb{I}\left[\hat{k}^{(n)} = k\right] \text{ for } k = 1, \ldots, K$$

    - **Refitting:** Optimize $J\ w.r.t. \{m\}$: Each center is set to mean of data assigned to it

$$m_k = \frac{\sum_n r_k^{(n)} x^{(n)}}{\sum_n r_k^{(n)}}$$

K-means for Image Segmentation



$\boxed{RGB}$

$\boxed{x, y}$

- Given image, construct "dataset" of pixels, represented by their RGB pixel intensities and grid locations
- Run k-means (with some modifications) to get superpixels

Why K-means Converges

- K-means algorithm reduces the cost at each iteration.

- Whenever an assignment is changed, the sum squared distances J of data points from their assigned cluster centers is reduced.

- Whenever a cluster center is moved, J is reduced.

Why K-means Converges

- **Test for convergence:** If the assignments do not change in the assignment step, we have converged (to at least a local minimum).

- This will always happen after a finite number of iterations, since the number of possible cluster assignments is finite

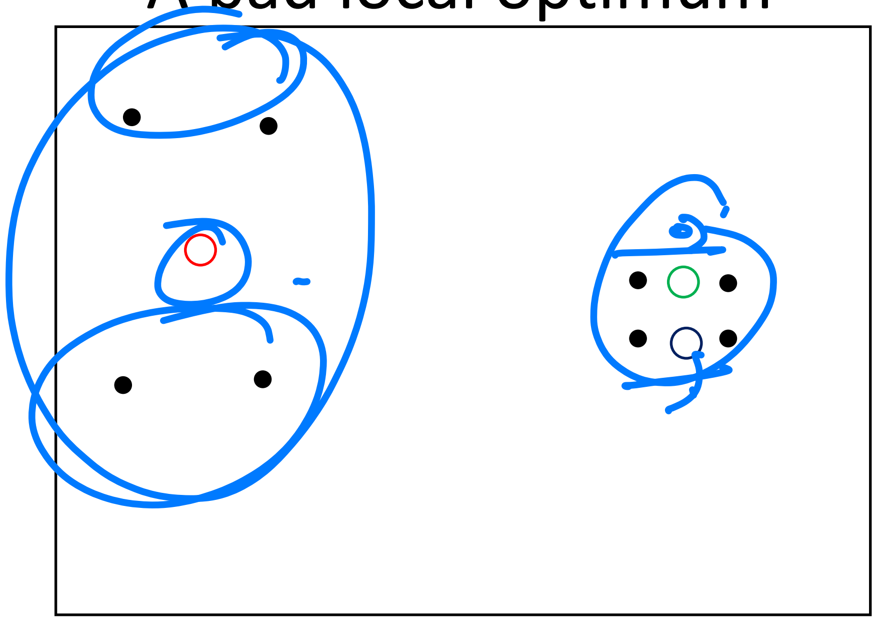
- K-means cost function after each assignment step (blue) and refitting step (red). The algorithm has converged after the third refitting step.

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Local Minima

- The objective J is non-convex (so coordinate descent on J is not guaranteed to converge to the global minimum)
- There is nothing to prevent k-means getting stuck at local minima.
- We could try many random starting points

A bad local optimum

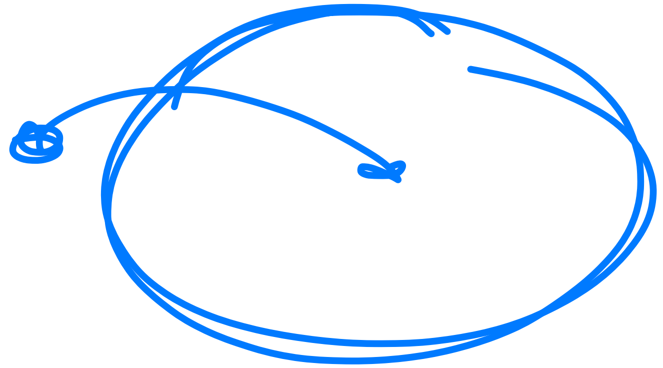


Other Distance Measures

- Assign each data point $x^{(n)}$ to nearest cluster k^* as per Mahalanobis distance:

$$k^* = \arg\min_k \underline{\left(x^{(n)} - m_k\right)^T \boxed{\mathbf{\Sigma}_k^{-1}} \left(x^{(n)} - m_k\right)}$$

- High variance indices get lower importance in distance computation



Soft K-means

- Instead of making hard assignments of data points to clusters, we can **make soft assignments**. One cluster may have a responsibility of 0.7 for a datapoint and another may have a responsibility of 0.3.

- Allows a cluster to use more information about the data in the refitting step.

- How do we decide on the soft assignments?

- We already saw this in multi-class classification: 1-of-K encoding vs softmax assignments

Soft K-means Algorithm

- **Initialization:** Set K means $\{m_k\}$ to random values
- Repeat until convergence (measured by how much J changes):
    - Assignment
    - Refitting

Soft K-means Algorithm

- Assignment: Each data point n given soft "degree of assignment" to each cluster mean k, based on responsibilities:

$$r_k^{(n)} = \frac{\exp\left[-\beta \left|\left|m_k - x^{(n)}\right|\right|^2\right]}{\sum_j \exp\left[-\beta \left|\left|m_j - x^{(n)}\right|\right|^2\right]}$$

$$r^{(n)} = \text{softmax}\left(-\beta \left\{\left|\left|m_k - x^{(n)}\right|\right|^2\right\}_{k=1}^{K}\right)$$

Note: As $\beta \rightarrow \infty$, **soft k-Means becomes k-Means!**

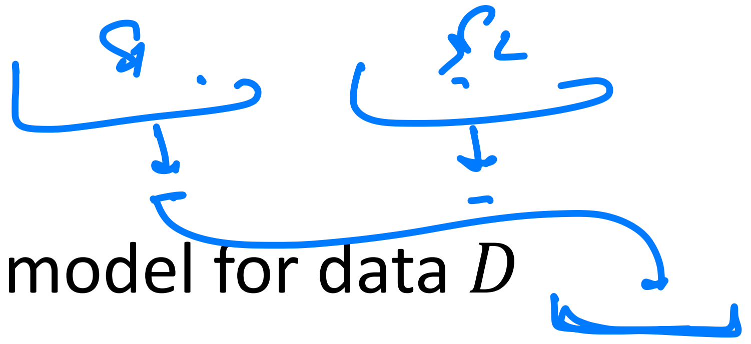
Soft K-means Algorithm

- Refitting: Model parameters, means, are adjusted to match sample means of datapoints they are responsible for:

$$m_k = \frac{\sum_n r_k^{(n)} x^{(n)}}{\sum_n r_k^{(n)}}$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

The Generative Model



- We'll be working with the following generative model for data D

- Assume a datapoint x is generated as follows:
  - Choose a cluster z from $\{1, \dots, K\}$ such that $p(z = k) = \pi_k$
  - Given z, sample x from a Gaussian distribution $N(x \mid \mu_z, I)$

- Can also be written:

$$\rightarrow \quad p(z = k) = \pi_k$$

$$p(x|z = k) = N(x \mid \mu_k, I)$$

Clusters from Generative Model

- This defines joint distribution $\underline{p(z,x)} = \underbrace{p(z)}_{\pi_k}\underline{p(x|z)}$ with parameters $\{\pi_k, \mu_k\}_{k=1}^{K}$ N

- The marginal of x is given by $\underline{p(x)} = \underline{\sum_z} \underline{p(z,x)}$

- $p(\underline{z = k} \mid \overset{\downarrow}{x})$ can be computed using Bayes rule:

$$p(\underline{z = \overset{\downarrow}{k} \mid} x) = \frac{p(x \mid z = k)\, p(z = k)}{\underline{p(x)}}$$

- $p(z = k \mid x)$ tells us the $\underline{\text{probability } x \text{ came from the } k^{th} \text{ cluster}}$

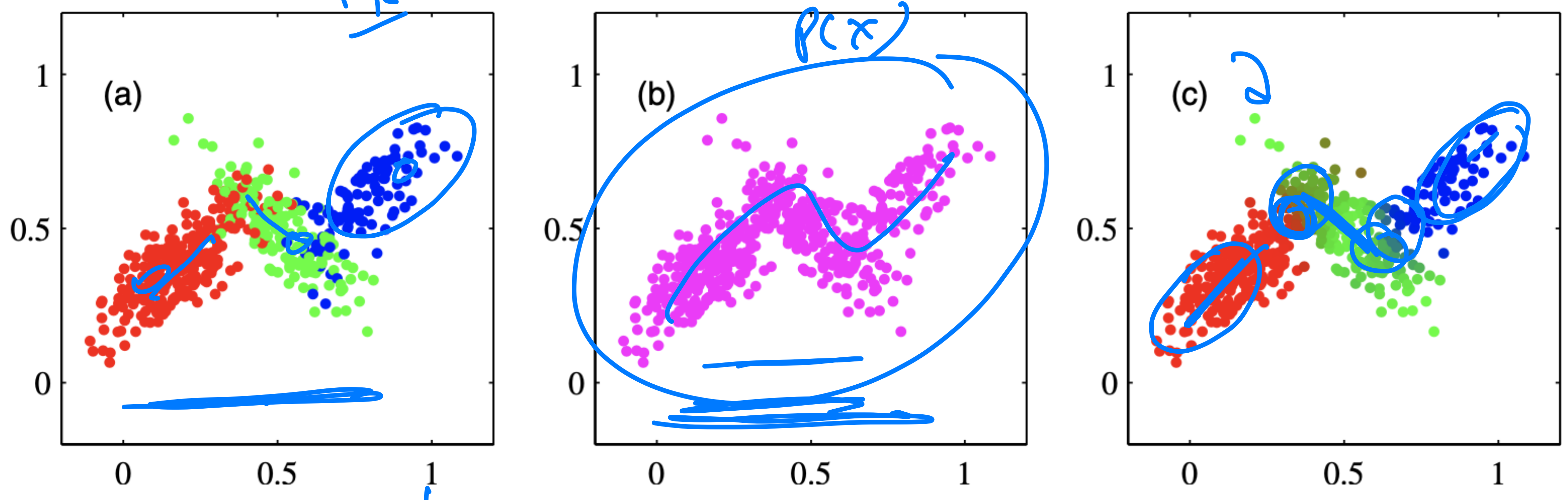
Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

The Generative Model

- 500 points drawn from a mixture of 3 Gaussians

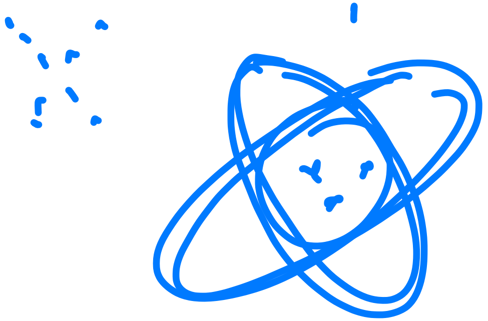
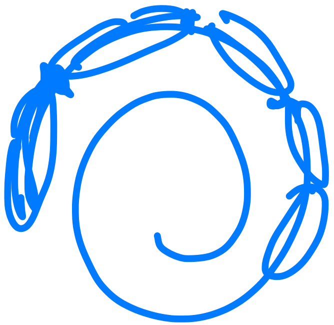
π_k

RGB



a) Samples from $p(x|z)$ b) Samples from the marginal $p(x)$ c) Responsibilities $p(z|x)$

$$p(x) = \sum_z p(x|z)\boxed{p(z)}$$

Estimation
error

Modelling
Error

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Maximum Likelihood with Latent Variables

- How should we choose the parameters $\underline{\{\pi_k, \mu_k\}}_{k=1}^{K}$?

- Maximum likelihood principle: choose parameters to maximize likelihood of the **observed data**

- We don’t observe the cluster assignments z, we only see the data x

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Maximum Likelihood with Latent Variables

- Given data $D = \{x^{(n)}\}_{n=1}^{N}$ choose parameters to maximize:

$$\log p(\overset{1}{D}) = \sum_{n=1}^{N} \log p(x^{(n)})$$

$p(n^1)\ p(n^2)$

$\sum_{i} p(x^{(i)})$

- We can find $p(x)$ by marginalizing out z:

$$p(x) = \sum_{k=1}^{K} p(z = k, x) = \sum_{k=1}^{K} p(z = k)\, p(x \mid z = k)$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

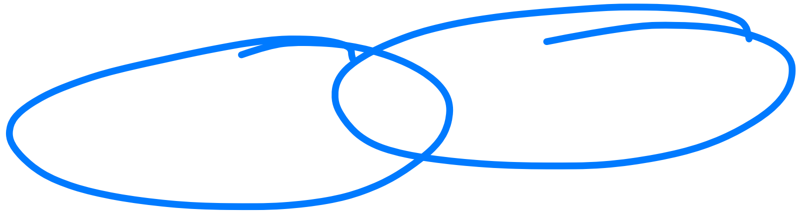
Gaussian Mixture Model (GMM)

- What is $p(x)$?

$$p(x) = \sum_{k=1}^{K} p(z = k)\, p(x \mid z = k) = \sum_{k=1}^{K} \pi_k N(x \mid \mu_k, I)$$

- This distribution is an example of a **Gaussian Mixture Model (GMM)**

- π_k are known as the **mixing coefficients.**



Gaussian Mixture Model (GMM)

$$p(x) = \sum_{k=1}^{K} p(z=k)\, p(x \mid z=k) = \sum_{k=1}^{\boxed{K}} \pi_k\, N(x \mid \mu_k, \boxed{I}) \xleftarrow{} \Sigma$$

- In general, we can have different covariance for each cluster, i.e.,
$$p(x \mid z = k) = N(x \mid \mu_k, \Sigma_k)$$

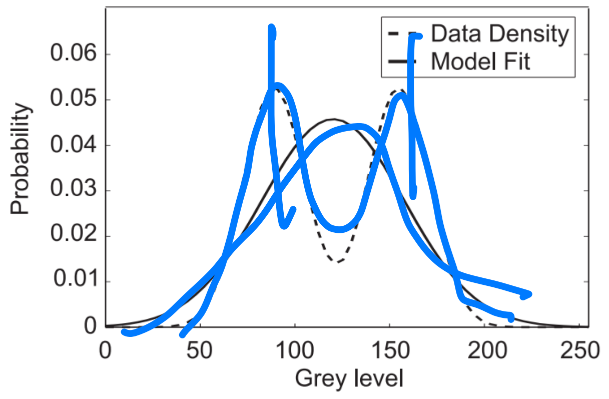
- For this lecture, we assume $\Sigma_k = I$ for simplicity.

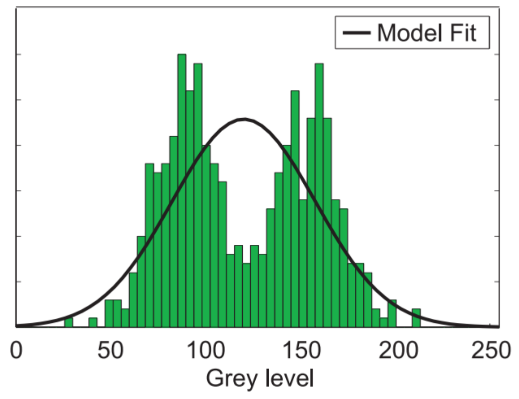
- If we allow arbitrary covariance matrices, GMMs are **universal approximators of probability densities** (if you have enough Gaussians).

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

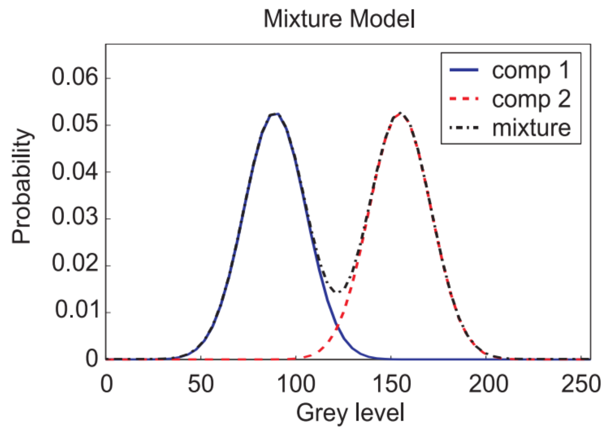
Visualizing a Mixture of Gaussians

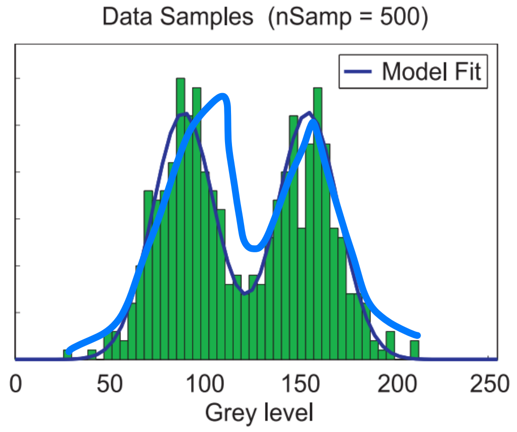
- If we fit single 1D Gaussian to the data:





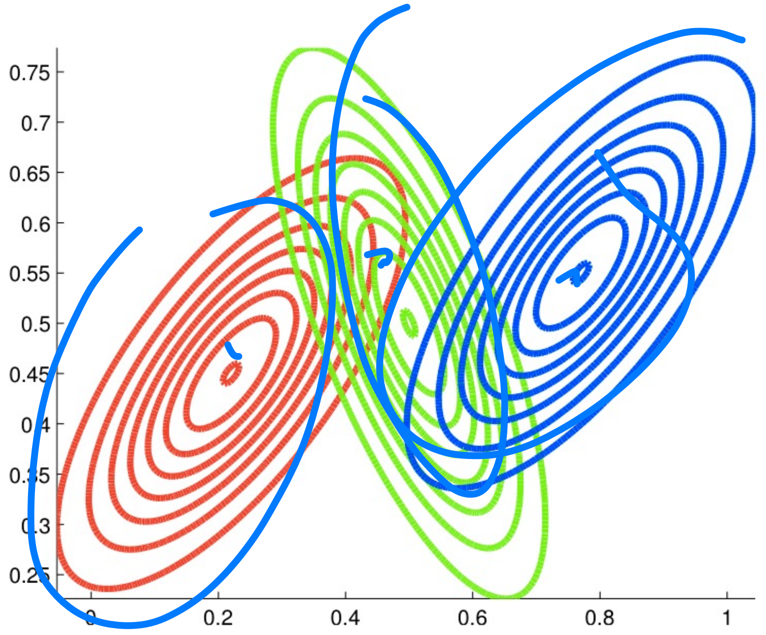
- If we fit a GMM with $K = 2$:

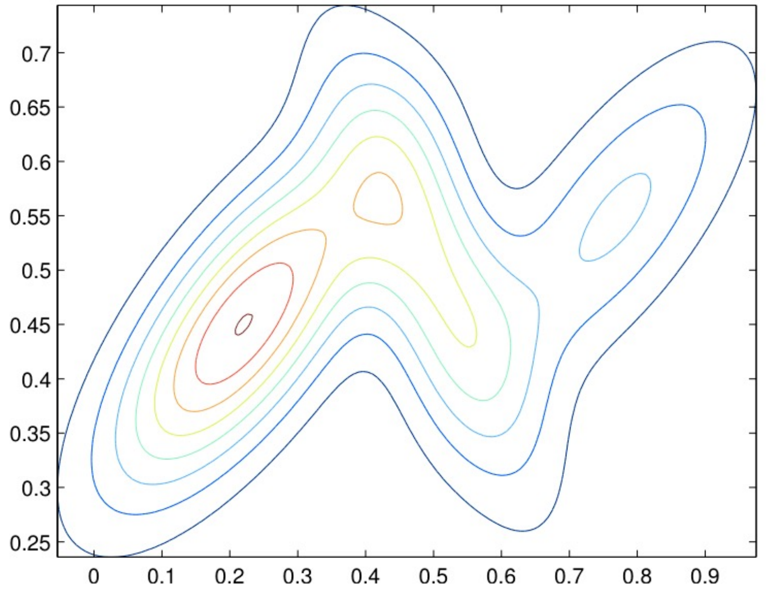


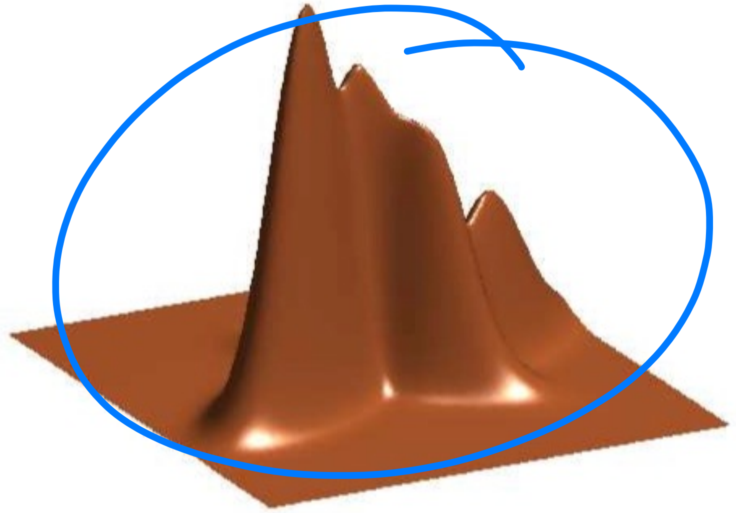


Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Visualizing a Mixture of Gaussians – 2D Gaussians







Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Fitting GMMs: Maximum Likelihood

- Maximum likelihood objective:

$$\log p(D) = \sum_{n=1}^{N} \log p\left(x^{(n)}\right) = \sum_{n=1}^{N} \overset{\log p(x)}{\log} \left(\overbrace{\sum_{k=1}^{K} \overset{p(z)}{\pi_K} \, \overset{p(x|z)}{N\left(x^{(n)} \mid \mu_k, I\right)}}^{p(x)} \right)$$

marginalizing

- How can we optimize the objective w.r.t. parameters $\{\pi_k, \mu_k\}$?
  - No closed form solution when we set derivatives to 0
  - Difficult because sum inside the log
- One option: gradient ascent (Not easy in this case). Can we do better?
- Can we have a closed form update?

Maximum Likelihood

Observation

- **If we knew** $\underline{z(n)}$ (cluster ID) for every $x(n)$, i.e. our dataset was complete:

$$D_{\text{complete}} = \{(\underline{z^{(n)}}, \underline{x^{(n)}})\}_{n=1}^{N}$$

- Then, the maximum likelihood problem is easy:

$$\underline{\log p(D_{\text{complete}})} = \sum_{n=1}^{N} \log \underline{p(z^{(n)}, x^{(n)})}$$

Maximum Likelihood

$$\log p\left(D_{\text{complete}}\right) = \sum_{n=1}^{N} \log p\left(z^{(n)}, x^{(n)}\right)$$

$$= \sum_{n=1}^{N} \underline{\log p\left(x^{(n)} \mid z^{(n)}\right)} + \underline{\log p\left(z^{(n)}\right)}$$

$$= \underline{\sum_{n=1}^{N} \sum_{k=1}^{K} \underline{\mathbb{I}\left[z^{(n)} = k\right]} \left(\underline{\log N\left(x^{(n)} \mid \acute{\mu}_k, I\right)} + \log \underline{\acute{\pi}_k}\right)}$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Maximum Likelihood

$$\log p(D_{\text{complete}}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \underbrace{\mathbb{I}[z^{(n)} = k] \left(\log N(x^{(n)} \mid \mu_k, I)\right.}\, + \underbrace{\log \pi_k}\left.\right)$$

- By maximizing $\log p(D_{\text{complete}})$, we would get this:

$f(x,y) = g(x) + h(y)$

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} \mathbb{I}[z^{(n)} = k]\, x^{(n)}}{\sum_{n=1}^{N} \mathbb{I}[z^{(n)} = k]} = \text{class means}$$

$$\hat{\pi}_k = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[z^{(n)} = k] = \text{class proportions}$$

Maximum Likelihood

- In real problem, we haven't observed the cluster assignments $\underline{z}^{(n)}$, but we can compute $p\left(z^{(n)}\middle|x^{(n)}\right)$ using Bayes rule
- Conditional probability (using Bayes rule) of z given x

$$\underline{p(z = k \mid x)} = \frac{p(z = k)\, p(x \mid z = k)}{p(x)}$$

$$= \frac{p(z = k)\, p(x \mid z = k)}{\sum_{j=1}^{K} p(z = j)\, p(x \mid z = j)} = \frac{\pi_k\, N(x \mid \mu_k, I)}{\underline{\sum_{j=1}^{K} N(x \mid \mu_j, I)}}$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

Maximum Likelihood

estimated

$$\log p(D_{\text{complete}}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{I}[z^{(n)} = k]\left(\log N(x^{(n)}|\mu_k, I) + \log \pi_k\right)$$

- We don't know the cluster assignments $\mathbb{I}[z^{(n)} = k]$, but we know their expectation $\mathbb{E}[\mathbb{I}[z^{(n)} = k] \mid x^{(n)}] = p(z^{(n)} = k|x^{(n)})$.

- If we plug in $r_k^{(n)} = p(z^{(n)} = k|x^{(n)})$ for $\mathbb{I}[z^{(n)} = k]$, we get:

$$\sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)}\left(\log N(x^{(n)}|\mu_k, I) + \log \pi_k\right)$$

Maximum Likelihood

$$\underline{\sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)}\left(\log N\left(x^{(n)} \middle| \mu_k, I\right) + \underline{\log \pi_k}\right)}$$

- This is still easy to optimize! Solution is similar to what we have seen:

$$\hat{\mu}_k = \underline{\frac{\sum_{n=1}^{N} r_k^{(n)} x^{(n)}}{\sum_{n=1}^{N} r_k^{(n)}}} \qquad \hat{\pi}_k = \underline{\frac{\sum_{n=1}^{N} r_k^{(n)}}{N}}$$

- This only works if we treat $r_k^{(n)} = \underline{\frac{\pi_k \, N\left(x^{(n)} \mid \mu_k, I\right)}{\sum_{j=1}^{K} \pi_j \, N\left(x^n \mid \mu_j, I\right)}}$ as fixed

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

How Can We Fit a Mixture of Gaussians?

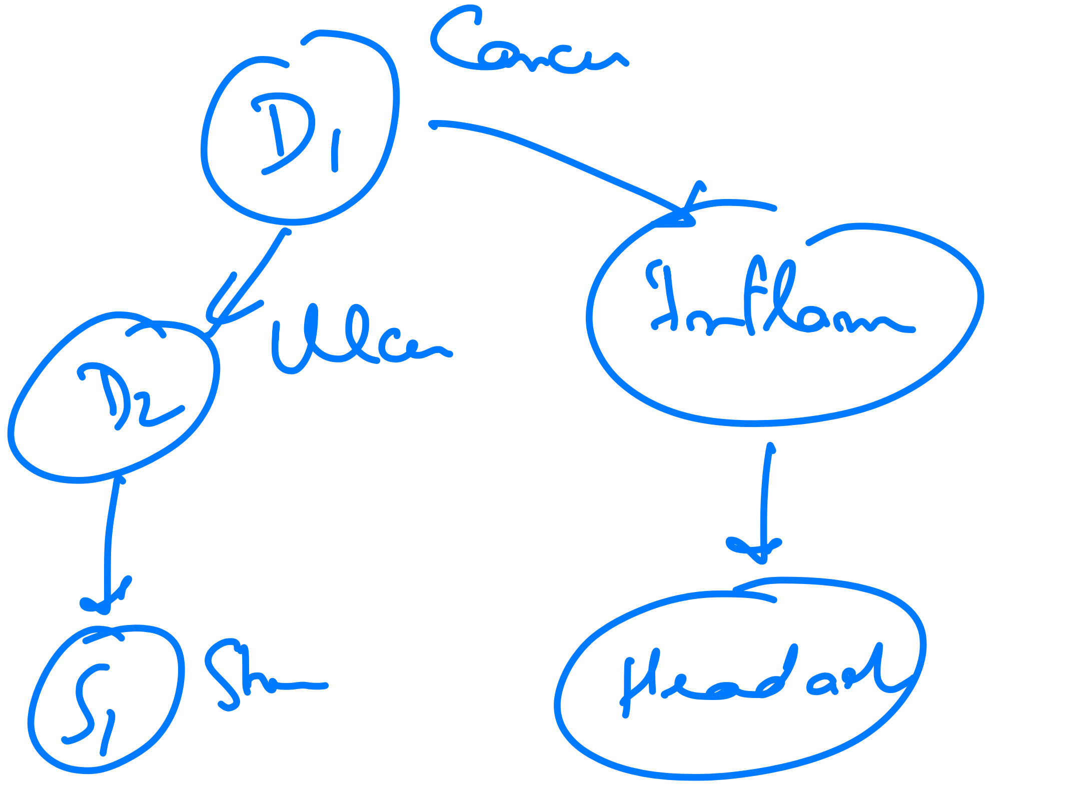
- This motivates the Expectation-Maximization algorithm, which alternates between two steps.

Expectation

- **E-step:** Compute the posterior probabilities $r_k^n = p\left(z^{(n)} = k \middle| x^{(n)}\right)$ given our current model:
  - i.e. how much do we think a cluster is responsible for generating a datapoint.

$\rightarrow \mu_k$

- **M-step:** Use the equations on the last slide to update the parameters, assuming $r_k^{(n)}$ are held fixed:
  - change the parameters of each Gaussian to maximize the probability that it would generate the data it is currently responsible for.



$C = 1$, $I = 0$, $H = 1$, $SA = 0$

EM Algorithm for GMM

- Initialize the means $\hat{\mu}_k$ and mixing coefficients $\hat{\pi}_k$
- Iterate until convergence:
  - E-step: Evaluate the responsibilities $r_k^{(n)}$ given current parameters

$$r_k^{(n)} = p\left(z^{(n)} = k \middle| x^{(n)}\right) = \frac{\hat{\pi}_k N(x^{(n)}|\mu_k, I)}{\sum_{j=1}^{K} \hat{\pi}_j N(x^n|\mu_j, I)} = \frac{\hat{\pi}_k \exp\left\{-\frac{1}{2}\left\|x^{(n)} - \hat{\mu}_k\right\|^2\right\}}{\sum_{j=1}^{K} \hat{\pi}_j \exp\left\{-\frac{1}{2}\left\|x^{(n)} - \hat{\mu}_j\right\|^2\right\}}$$

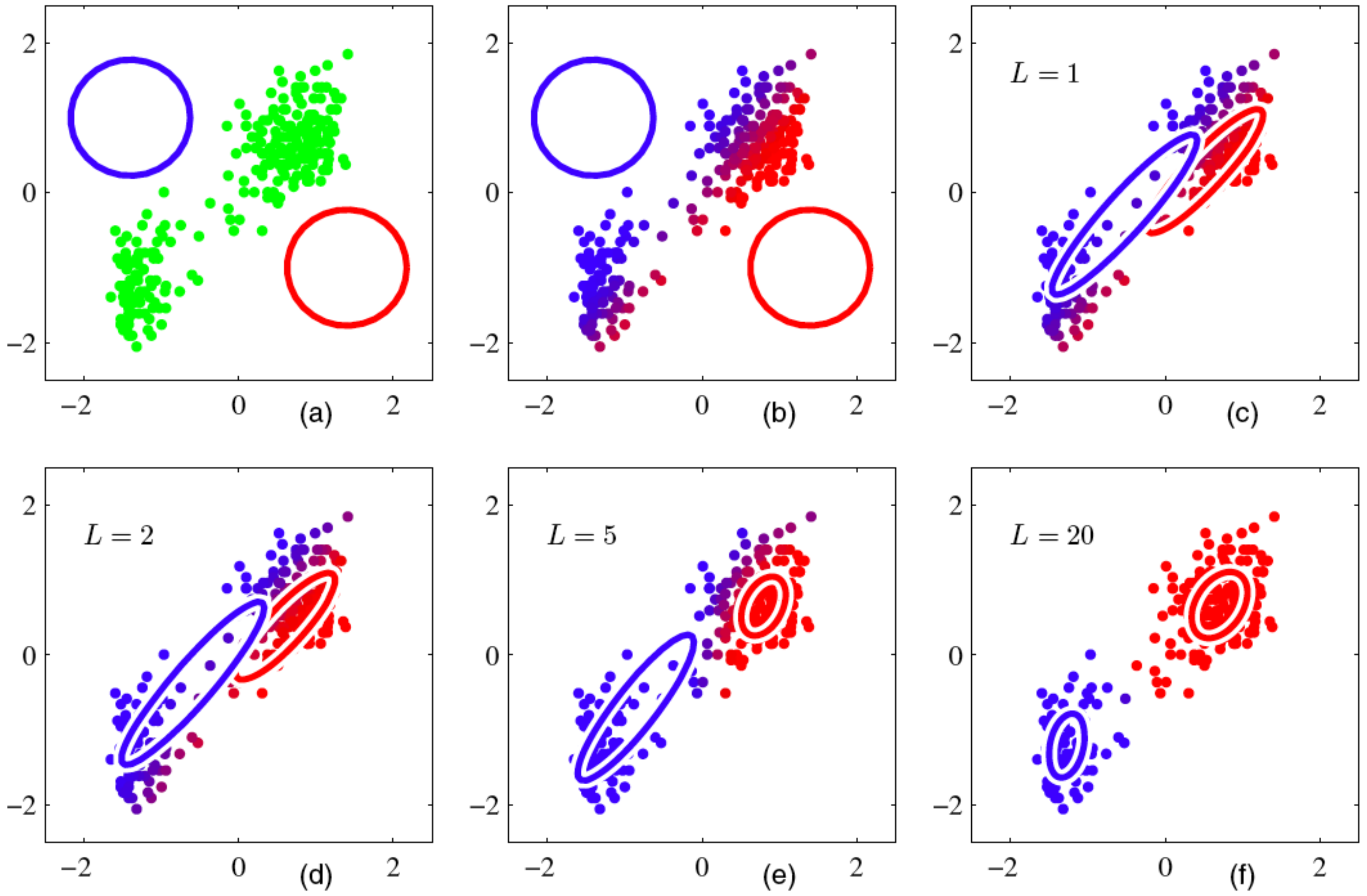
  - M-step: Re-estimate the parameters given current responsibilities

$$\hat{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_k^{(n)} x^{(n)} \qquad \hat{\pi}_k = \frac{N_k}{N}\ with\ \ N_k = \sum_{n=1}^{N} r_k^{(n)}$$

  - Evaluate log likelihood and check for convergence

$$\log p(D) = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{\cancel{N}\ K} \hat{\pi}_k N(x^{(n)}|\mu_k, I)\right)$$

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi



What just happened: A review

- The maximum likelihood objective $\sum_{n=1}^{N} \log p\left(x^{(n)}\right)$ was hard to optimize

- The complete data likelihood objective was easy to optimize:

$$\sum_{n=1}^{N} \log p\left(z^{(n)}, x^{(n)}\right) = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}\left[z^{(n)} = k\right]\left(\log N\left(x^{(n)} \middle| \mu_k, I\right) + \log \pi_k\right)$$

- We don't know $z^{(n)}$'s (they are latent), so we replaced $\mathbb{I}\left[z^{(n)} = k\right]$ with responsibilities $r_k^{(n)} = p\left(z^{(n)} = k \middle| x^{(n)}\right)$
- That is: we replaced $\mathbb{I}\left[z^{(n)} = k\right]$ with its expectation under $p\left(z^{(n)} \middle| x^{(n)}\right)$ (E-step).

What just happened: A review

- We ended up with the expected complete data log-likelihood:

$$\sum_{n=1}^{N} E_{p\left(z^{(n)}|x^{(n)}\right)}\left[\log p\left(z^{(n)}, x^{(n)}\right)\right] = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{I}\left[z^{(n)} = k\right]\left(\log N\left(x^{(n)}|\mu_k, I\right) + \log \pi_k\right)$$

which we maximized over parameters $\{\pi_k, \mu_k\}_k$ (M-step)

- The EM algorithm alternates between:
    - **The E-step:** computing the $r_k^{(n)} = p(z(n) = k|x^{(n)})$ (i.e. **expectations** $\mathbb{E}\left(\left[\mathbb{I}\left[z^{(n)} = k\right]\right] | \, x(n)\right)$ given the current model parameters π_k, μ_k
    - **The M-step:** update the model parameters π_k, μ_k to optimize the expected complete data log-likelihood

Relation to k-Means

- The K-Means Algorithm:
  1. **Assignment step:** Assign each data point to the closest cluster
  2. **Refitting step:** Move each cluster center to the average of the data assigned to it

- The EM Algorithm:
  1. **E-step:** Compute the posterior probability over z given our current model
  2. **M-step:** Maximize the probability that it would generate the data it is currently responsible for.

Chetan Arora
Department of Computer Science and Engineering, IIT Delhi

GMM Recap

- A probabilistic view of clustering - Each cluster corresponds to a different Gaussian.
- Model using **latent variables**.
- General approach, can replace Gaussian with other distributions (continuous or discrete)
- More generally, mixture models are very powerful models, i.e. **universal distribution approximators**
- Optimization is done using the **EM** algorithm.