

Clustering and expectation maximization

School of Electrical and Computer Engineering

University of Tehran

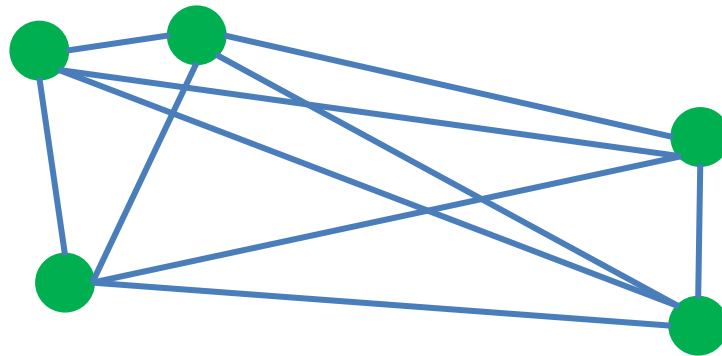
Erfan Darzi

erfandarzi@ut.ac.ir



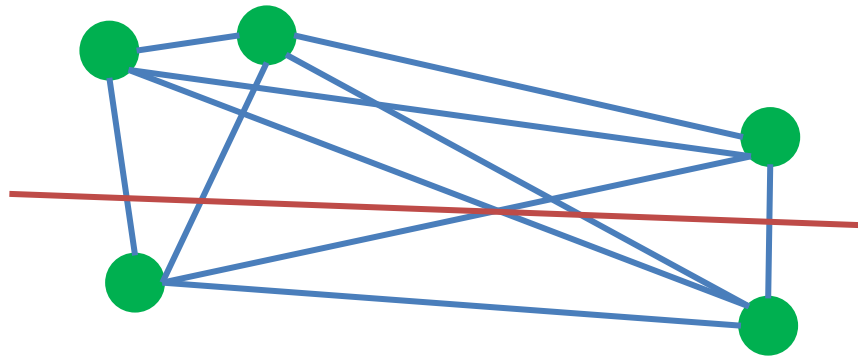
Clustering as graph cut

- Describe the pairwise distance via a graph



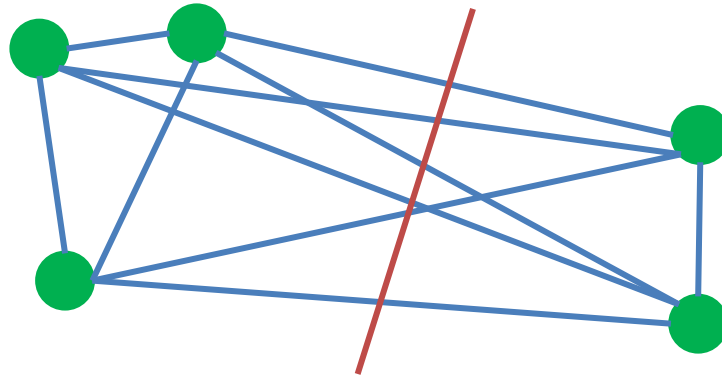
Clustering as graph cut

- Describe the pairwise distance via a graph
 - Clustering can be obtained via graph cut



Clustering as graph cut

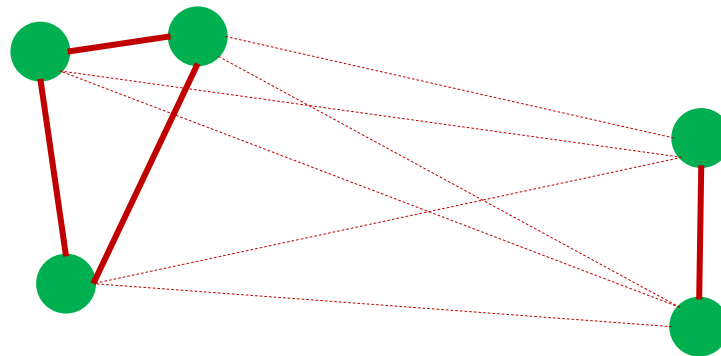
- Describe the pairwise distance via a graph
 - Clustering can be obtained via graph cut



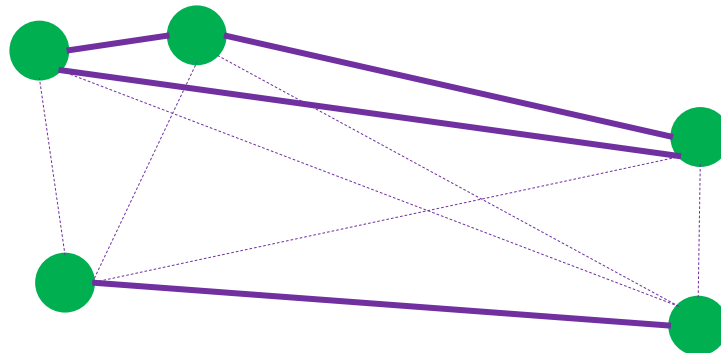
Clustering as graph cut

- Describe the pairwise distance via a graph
 - Clustering can be obtained via graph cut

Cut by class label



Cut by cluster label



Recap: external validation

- Given class label Ω on each instance
 - Rand index

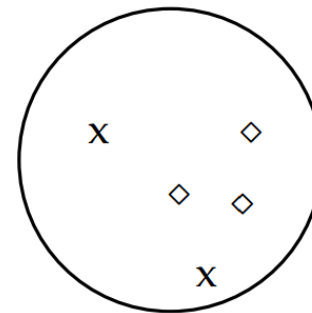
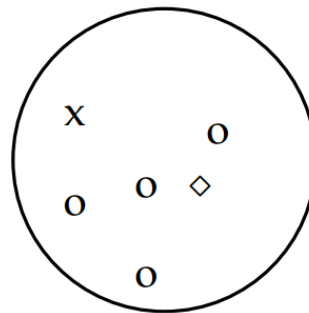
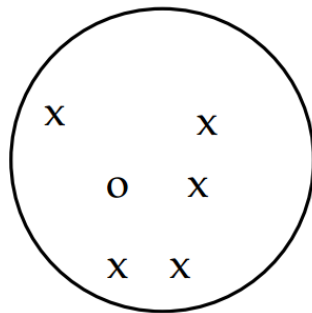
	$w_i = w_j$	$w_i \neq w_j$
$c_i = c_j$	20	20
$c_i \neq c_j$	24	72

$$TP + FP = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

cluster 1

$$TP = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

cluster 2 cluster 3



Today's lecture

- k -means clustering
 - A typical partitional clustering algorithm
 - Convergence property
 - Expectation Maximization algorithm
 - Gaussian mixture model



Partitional clustering algorithms

- Partition instances into exactly k non-overlapping clusters
 - Flat structure clustering
 - Users need to specify the cluster size k
 - Task: identify k non-overlapping clusters that optimize



Partitional clustering algorithms

- Partition instances into exactly k non-overlapping clusters
 - Typical criterion
 - $\max \sum_{i \neq j} d(c_i, c_j) - C \sum_i \sigma_i$
 - Optimal solution: enumerate every possible partition of size k and return the one maximizes the criterion

Optimize this in an alternative way

Inter-cluster distance

Intra-cluster distance

Let's approximate this!

Unfortunately, this is NP-hard!



k -means algorithm

Input: cluster size k , instances $\{x_i\}_{i=1}^N$, distance metric $d(\cdot, \cdot)$

Output: cluster membership assignments $\{z_i\}_{i=1}^N$

1. Initialize k cluster centroids $\{c_i\}_{i=1}^k$ (randomly if no domain knowledge is available)
2. Repeat until no instance changes its cluster membership:
 - Decide the cluster membership of instances by assigning them to the nearest cluster centroid

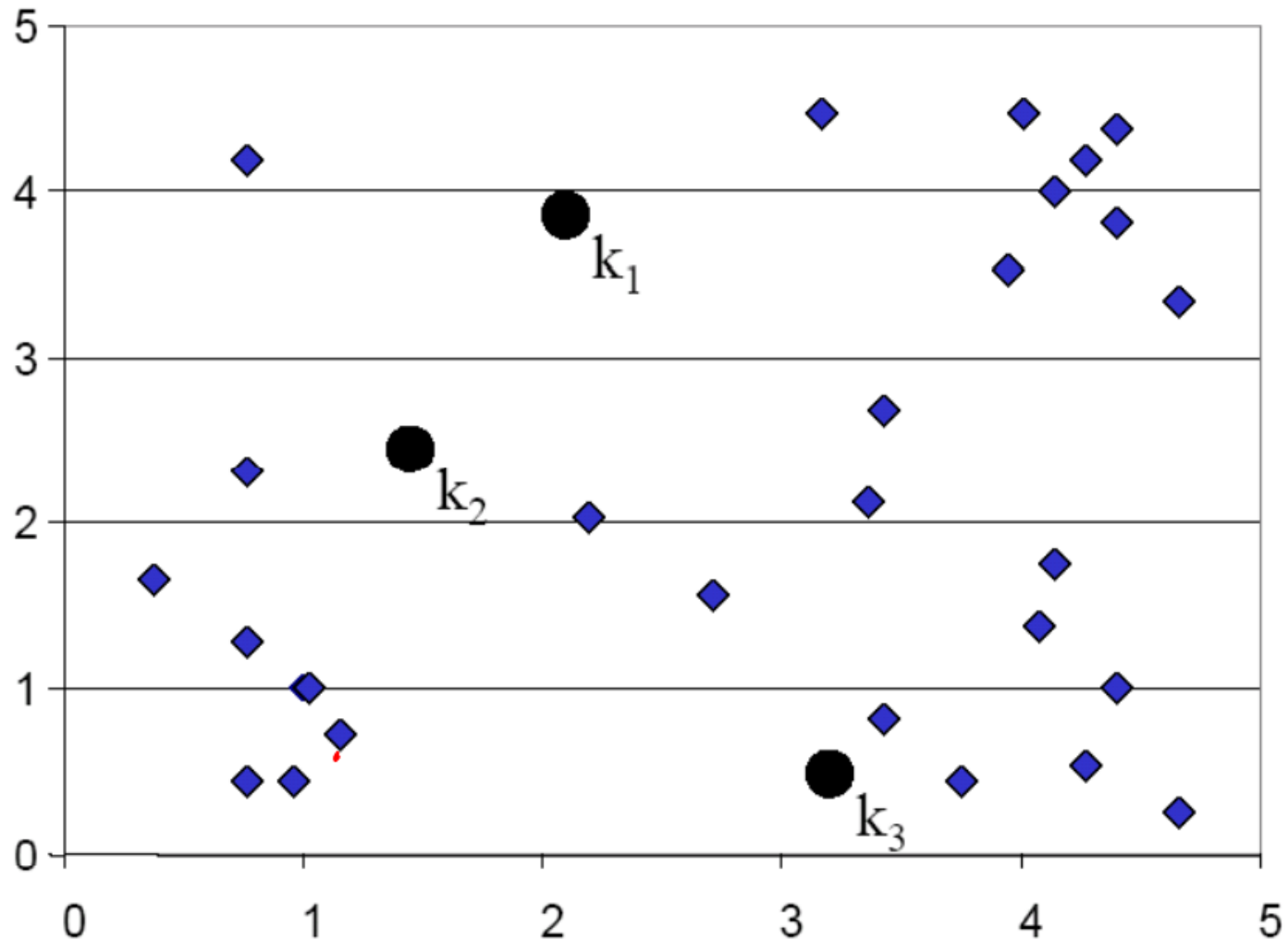
$$z_i = \operatorname{argmin}_k d(c_k, x_i) \quad \text{Minimize intra distance}$$

- Update the k cluster centroids based on the assigned cluster membership

$$c_k = \frac{\sum_i \delta(z_i = c_k) x_i}{\sum_i \delta(z_i = c_k)} \quad \text{Maximize inter distance}$$

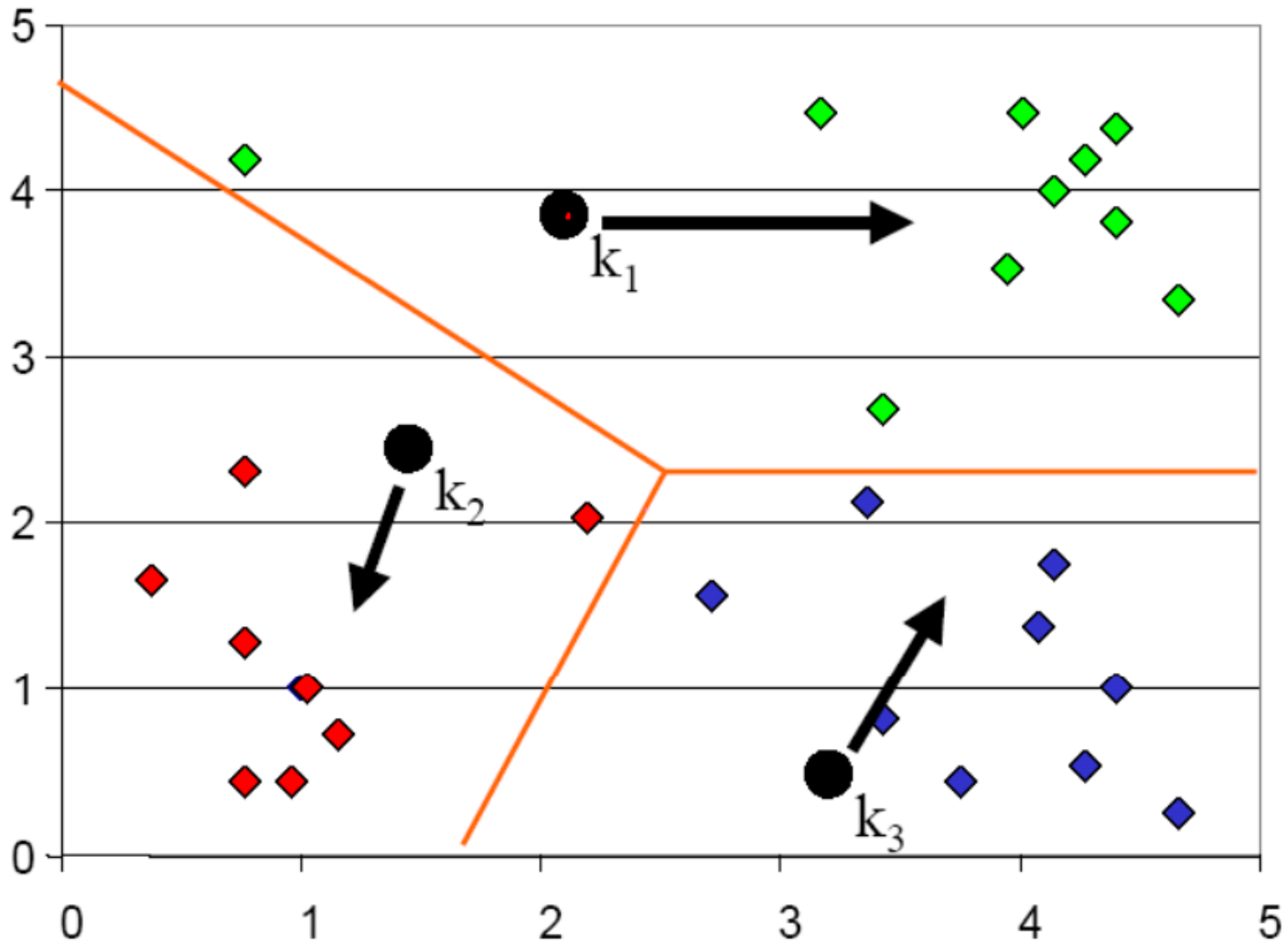


k -means illustration

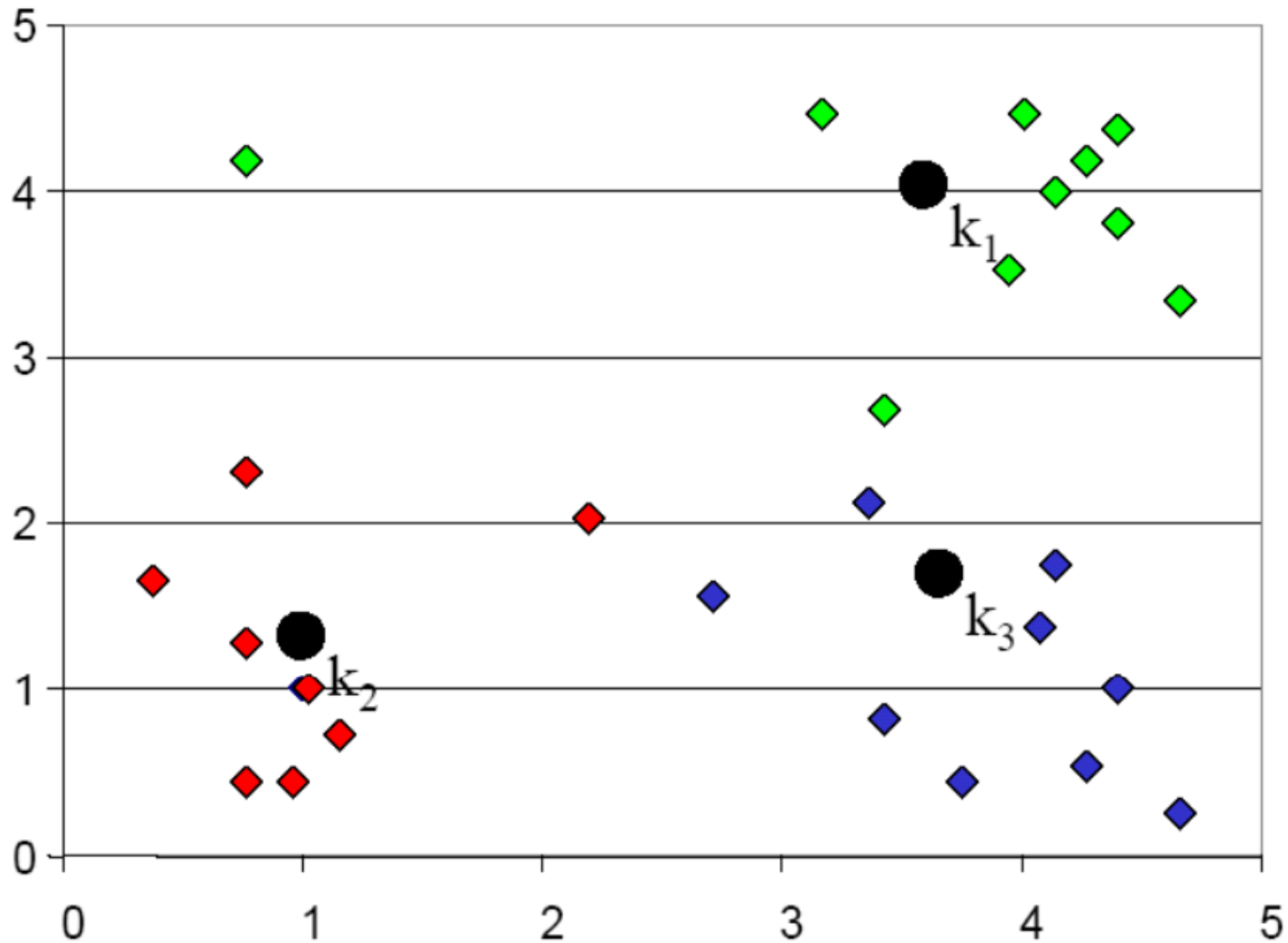


k -means illustration

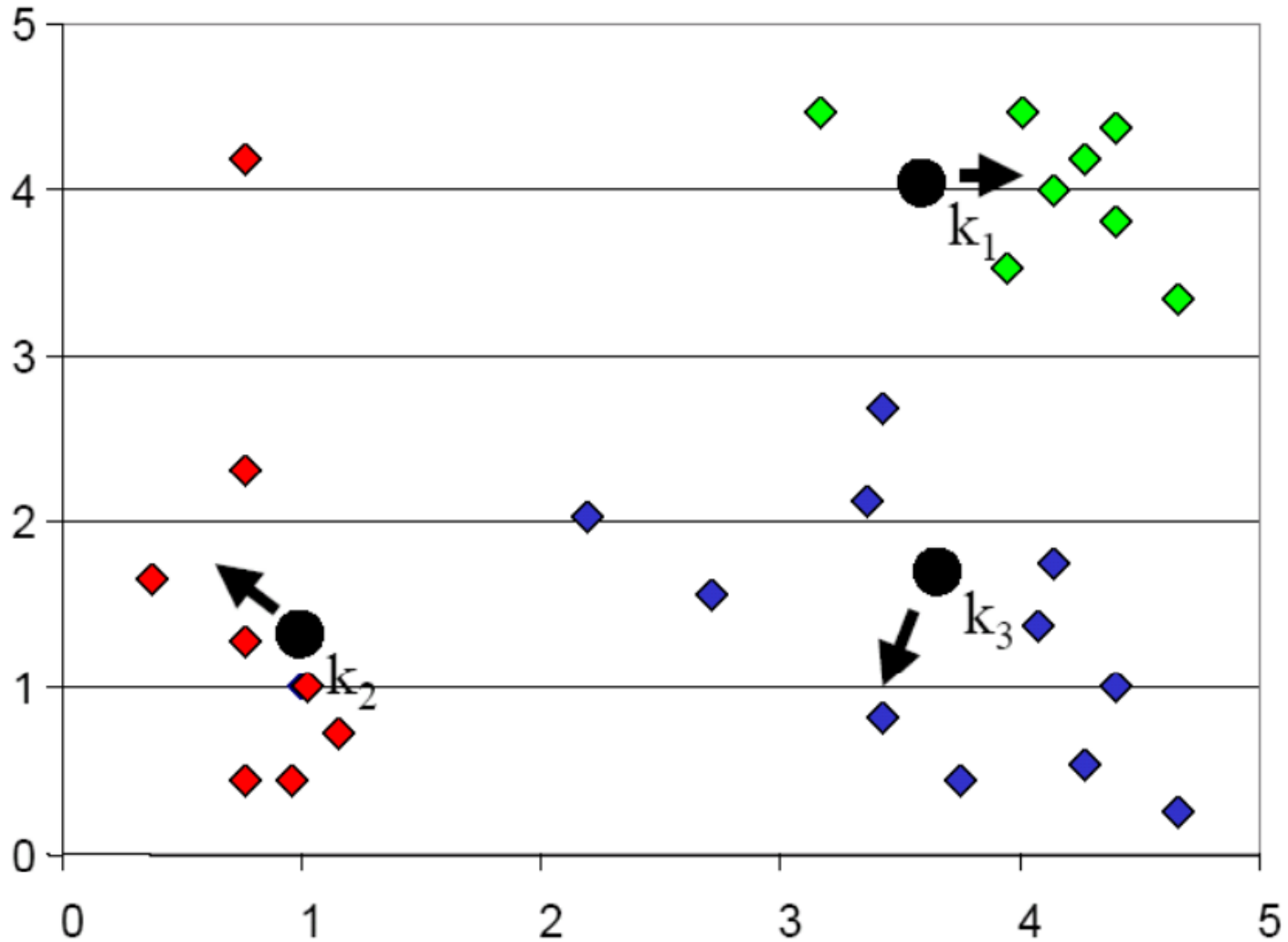
Voronoi diagram



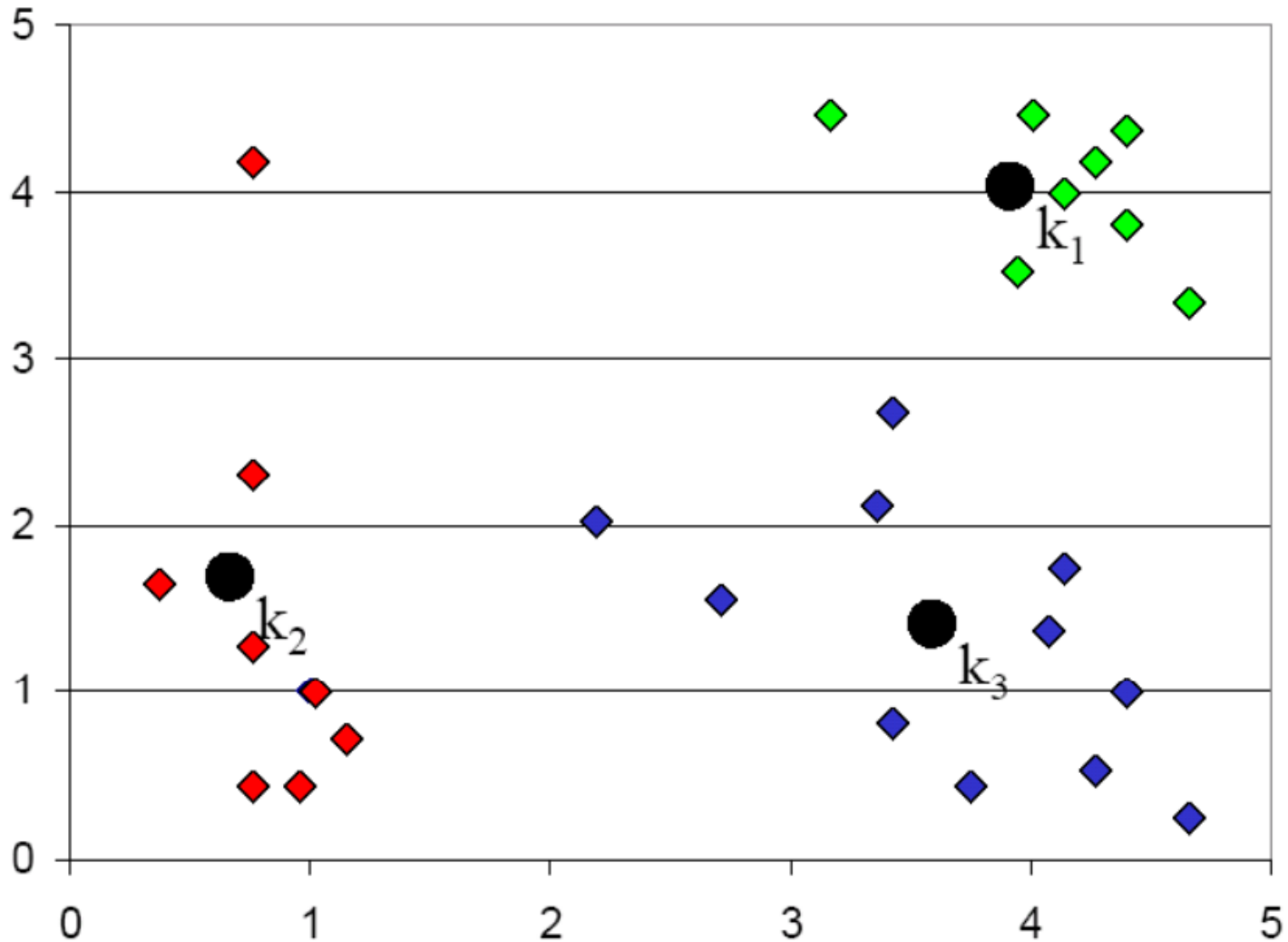
k -means illustration



k -means illustration



k -means illustration



Complexity analysis

- Decide cluster membership
 - $O(kn)$
- Compute cluster centroid
 - $O(n)$
- Assume k -means stops after l iterations
 - $O(knl)$

Don't forget the complexity of distance computation, e.g., $O(V)$ for Euclidean distance



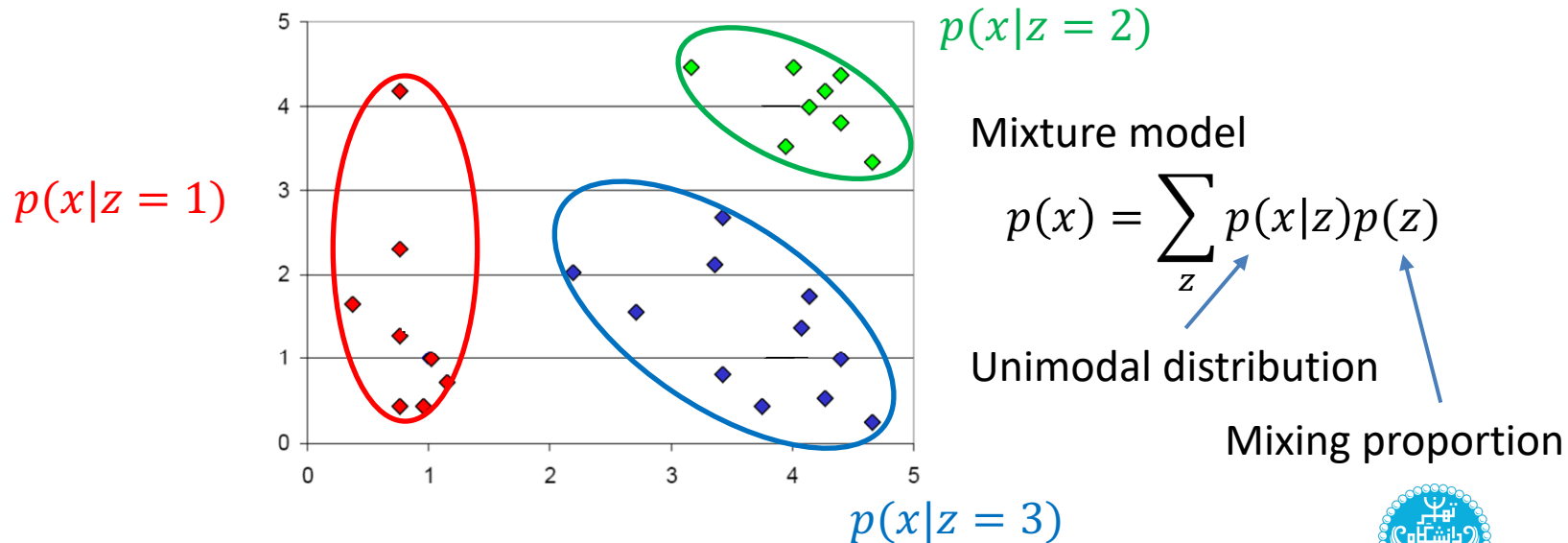
Convergence property

- Why will k -means stop?
 - Answer: it is a special version of Expectation Maximization (EM) algorithm, and EM is guaranteed to converge
 - However, it is only guaranteed to converge to local optimal, since k -means (EM) is a greedy algorithm



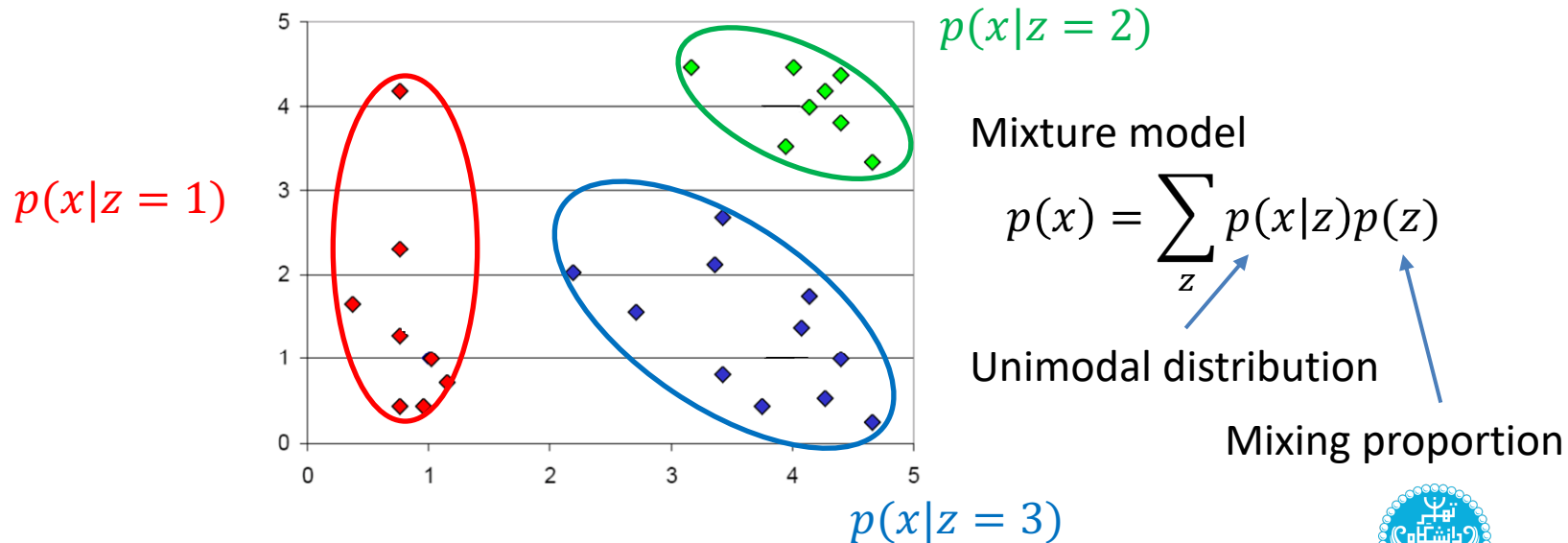
Probabilistic interpretation of clustering

- The density model of $p(x)$ is multi-modal
- Each mode represents a sub-population
 - E.g., unimodal Gaussian for each group



Probabilistic interpretation of clustering

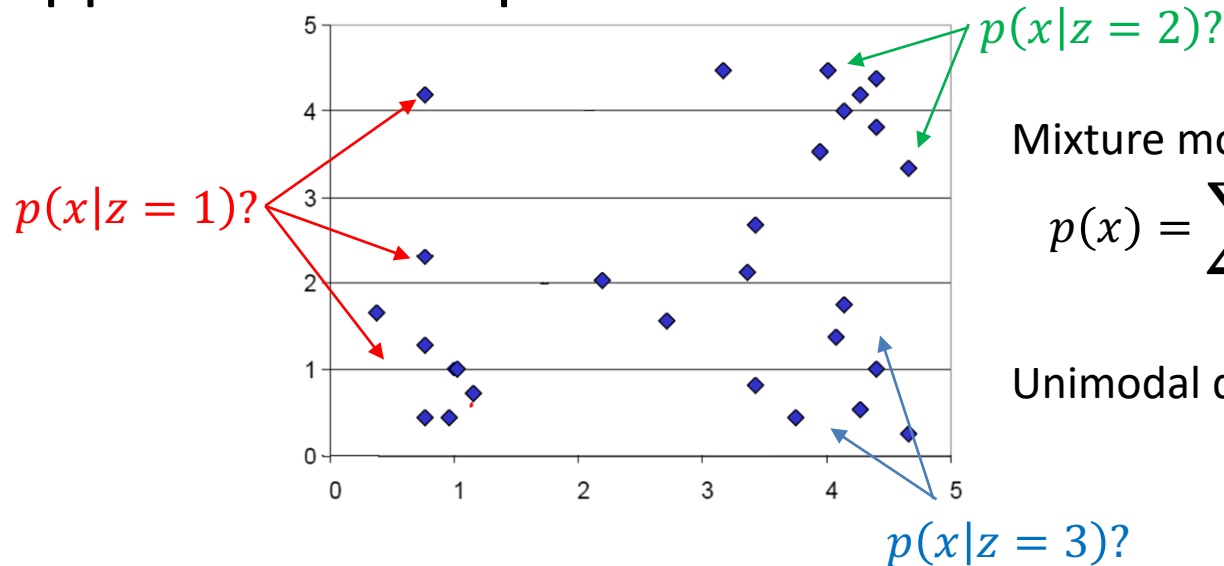
- If z is known for every x
 - Estimating $p(z)$ and $p(x|z)$ is easy
 - Maximum likelihood estimation
 - This is Naïve Bayes



Probabilistic interpretation of clustering

- But z is unknown for all x
 - Estimating $p(z)$ and $p(x|z)$ is generally hard
 - $\max_{\alpha, \beta} \sum_i \log \sum_{z_i} p(x_i|z_i, \beta) p(z_i|\alpha)$
 - Appeal to the Expectation Maximization algorithm

Usually a constrained optimization problem



Mixture model

$$p(x) = \sum_z p(x|z)p(z)$$

Unimodal distribution

Mixing proportion



Introduction to EM

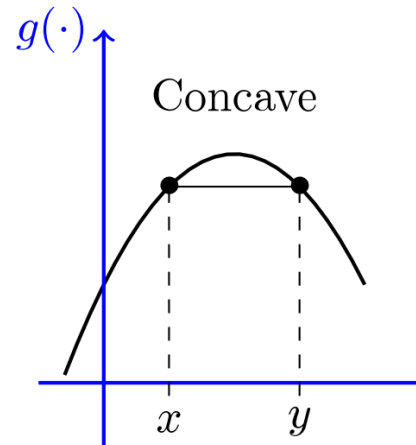
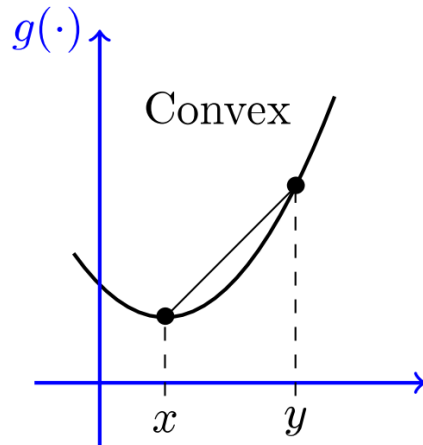
- Parameter estimation
 - All data is observable
 - Maximum likelihood estimator
 - Optimize the analytic form of $L(\theta) = \log p(X|\theta)$
 - Missing/unobservable data
 - Data: X (observed) + Z (hidden) ↖ *E.g. cluster membership*
 - Likelihood: $L(\theta) = \log \sum_z p(X, Z|\theta)$
 - Approximate it! ↖ *Most of cases are intractable*



Background knowledge

- Jensen's inequality
 - For any convex function $f(x)$ and positive weights λ ,

$$f\left(\sum_i \lambda_i x_i\right) \leq \sum_i \lambda_i f(x_i) \quad \sum_i \lambda_i = 1$$



Expectation Maximization

- Maximize data likelihood function by pushing the lower bound

Proposal distributions for Z

$$-L(\theta) = \log \sum_Z p(X, Z|\theta) = \log \sum_Z \frac{q(Z)p(X, Z|\theta)}{q(Z)}$$

Jensen's inequality $f(E[x]) \geq E[f(x)]$

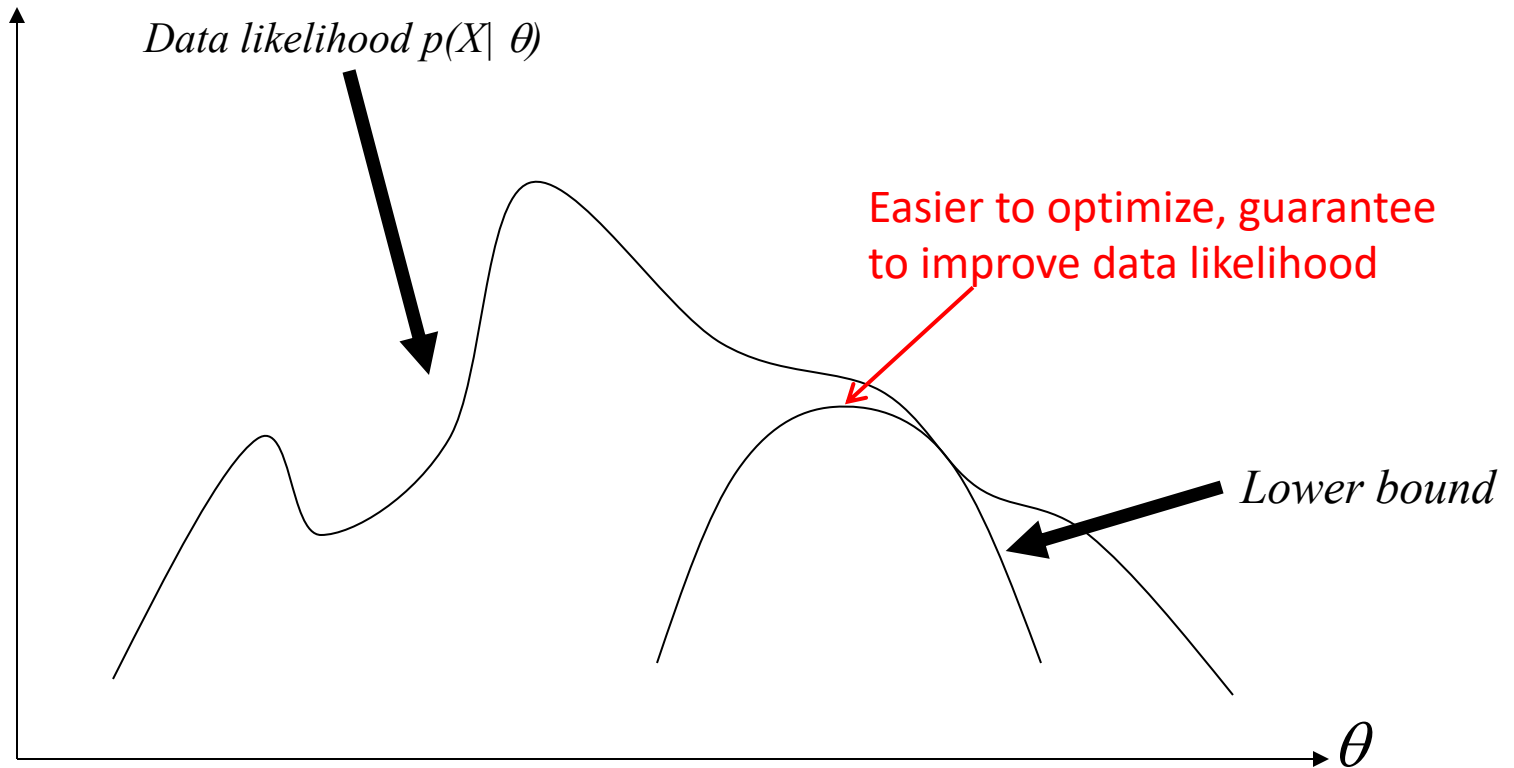
$$\geq \sum_Z q(Z) \log p(X, Z|\theta) - \sum_Z q(Z) \log q(Z)$$

Lower bound!

Components we need to tune when optimizing $L(\theta)$: $q(Z)$ and θ !



Intuitive understanding of EM



Expectation Maximization (cont)

- Optimize the lower bound w.r.t. $q(Z)$

$$-L(\theta) \geq \sum_Z q(Z) \log p(X, Z|\theta) - \sum_Z q(Z) \log q(Z)$$

$$= \sum_Z q(Z) [\log p(Z|X, \theta) + \log p(X|\theta)] - \sum_Z q(Z) \log q(Z)$$

$$= \sum_Z q(Z) \log \frac{p(Z|X, \theta)}{q(Z)} + \log p(X|\theta)$$

negative KL-divergence between $q(Z)$ and $p(Z|X, \theta)$ Constant with respect to $q(Z)$

$$KL(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$



Expectation Maximization (cont)

- Optimize the lower bound w.r.t. $q(Z)$
 - $L(\theta) \geq -KL(q(Z)||p(Z|X, \theta)) + L(\theta)$
 - KL-divergence is non-negative, and equals to zero i.f.f. $q(Z) = p(Z|X, \theta)$
 - A step further: when $q(Z) = p(Z|X, \theta)$, we will get $L(\theta) \geq L(\theta)$, i.e., the lower bound is tight!
 - Other choice of $q(Z)$ cannot lead to this tight bound, but might reduce computational complexity
 - **Note:** calculation of $q(Z)$ is based on current θ



Expectation Maximization (cont)

- Optimize the lower bound w.r.t. $q(Z)$
 - Optimal solution: $q(Z) = p(Z|X, \theta^t)$



Posterior distribution of Z given current model θ^t

In k-means: this corresponds to assigning instance x_i to its closest cluster centroid c_k

$$z_i = \operatorname{argmin}_k d(c_k, x_i)$$



Expectation Maximization (cont)

- Optimize the lower bound w.r.t. θ
 - $L(\theta) \geq \sum_Z p(Z|X, \theta^t) \log p(X, Z|\theta) -$
 ~~$\sum_Z p(Z|X, \theta^t) \log p(Z|X, \theta^t)$~~ \leftarrow Constant w.r.t. θ
 - $\theta^{t+1} = \operatorname{argmax}_{\theta} \sum_Z p(Z|X, \theta^t) \log p(X, Z|\theta)$
 $= \operatorname{argmax}_{\theta} E_{Z|X, \theta^t} [\log p(X, Z|\theta)]$



Expectation of complete data likelihood

In k-means, we are not computing the expectation, but the most probable

configuration, and then $c_k = \frac{\sum_i \delta(z_i=c_k)x_i}{\sum_i \delta(z_i=c_k)}$



Expectation Maximization

- EM tries to iteratively maximize likelihood
 - “Complete” data likelihood: $L^c(\theta) = \log p(X, Z|\theta)$
 - Starting from an initial guess $\theta^{(0)}$,
 1. **E-step**: compute the expectation of the complete data likelihood

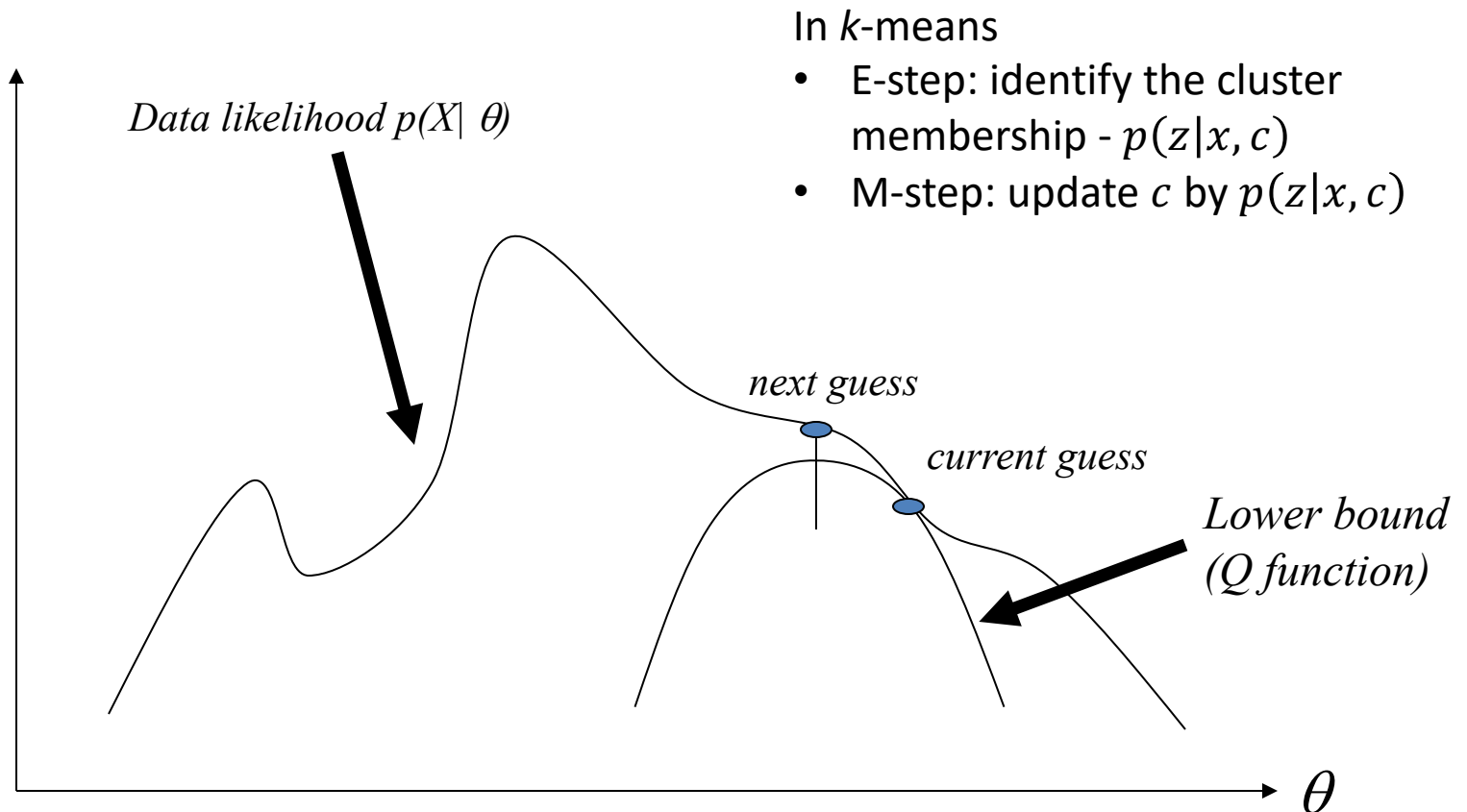
$$Q(\theta; \theta^t) = E_{Z|X, \theta^t}[L^c(\theta)] = \sum_Z \underbrace{p(Z|X, \theta^t)}_{\text{Key step!}} \log p(X, Z|\theta)$$

2. **M-step**: compute $\theta^{(t+1)}$ by maximizing the Q-function

$$\theta^{t+1} = \operatorname{argmax}_{\theta} Q(\theta; \theta^t)$$



An intuitive understanding of EM



E-step = computing the lower bound

M-step = maximizing the lower bound



Convergence guarantee

- Proof of EM

$$\log p(X|\theta) = \log p(Z, X|\theta) - \log p(Z|X, \theta)$$

Taking expectation with respect to $p(Z|X, \theta^t)$ of both sides:

$$\begin{aligned}\log p(X|\theta) &= \sum_Z p(Z|X, \theta^t) \log p(Z, X|\theta) - \sum_Z p(Z|X, \theta^t) \log p(Z|X, \theta) \\ &= Q(\theta; \theta^t) + \underline{H(\theta; \theta^t)} \quad \leftarrow \text{Cross-entropy}\end{aligned}$$

Then the change of log data likelihood between EM iteration is:

$$\log p(X|\theta) - \log p(X|\theta^t) = Q(\theta; \theta^t) + H(\theta; \theta^t) - Q(\theta^t; \theta^t) - H(\theta^t; \theta^t)$$

By Jensen's inequality, we know $H(\theta; \theta^t) \geq H(\theta^t; \theta^t)$, that means

$$\log p(X|\theta) - \log p(X|\theta^t) \geq Q(\theta; \theta^t) - Q(\theta^t; \theta^t) \geq \underline{0}$$

M-step guarantee this



What is not guaranteed

- Global optimal is not guaranteed!
 - Likelihood: $L(\theta) = \log \sum_Z p(X, Z|\theta)$ is non-convex in most of cases
 - EM boils down to a greedy algorithm
 - Alternative ascent
- Generalized EM
 - E-step: $\hat{q}(Z) = \operatorname{argmin}_{q(Z)} KL(q(Z) || p(Z|X, \theta^t))$
 - M-step: choose θ that improves $Q(\theta; \theta^t)$



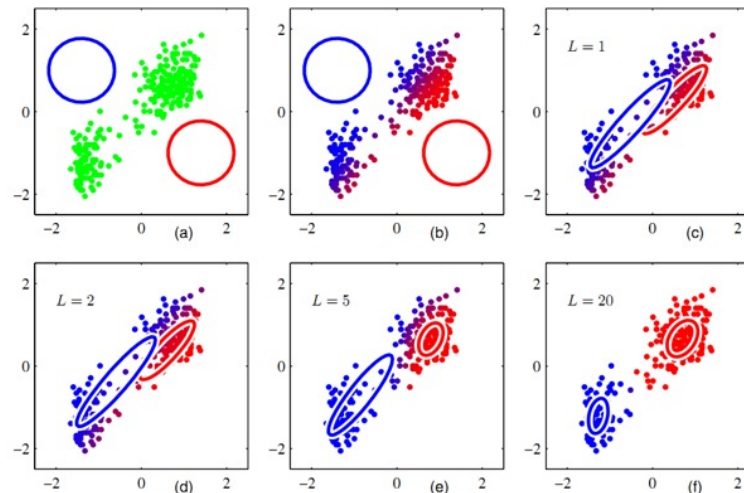
k-means v.s. Gaussian Mixture

- If we use Euclidean distance in *k*-means
 - We have explicitly assumed $p(x|z)$ is Gaussian
 - Gaussian Mixture Model (GMM)

- $p(x|z) = N(\mu_z, \Sigma_z)$
- $p(z) = \alpha_z$ ← Multinomial

$$P(x|z) = \frac{1}{\sqrt{(2\pi)^k \Sigma_z}} e^{-\frac{(x-\mu_z)^T \Sigma_z^{-1} (x-\mu_z)}{2}}$$

We do not consider cluster size in *k*-means

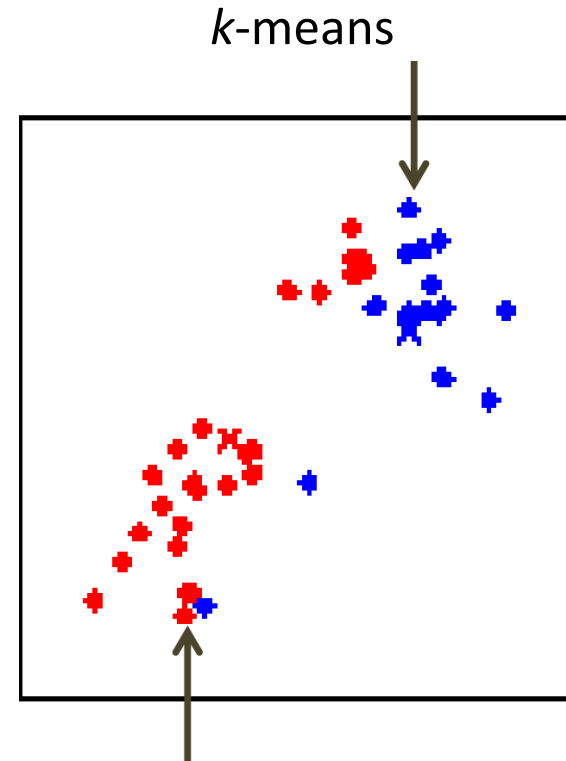
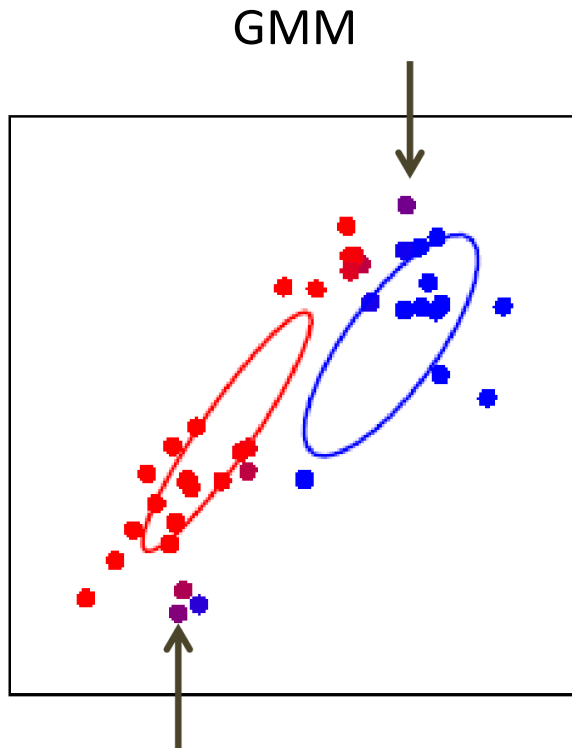


In *k*-means, we assume equal variance across clusters, so we don't need to estimate them



k -means v.s. Gaussian Mixture

- Soft v.s., hard posterior assignment



k -means in practice

- Extremely fast and scalable
 - One of the most popularly used clustering methods
 - Top 10 data mining algorithms – ICDM 2006
 - Can be easily parallelized
 - Map-Reduce implementation
 - Mapper: assign each instance to its closest centroid
 - Reducer: update centroid based on the cluster membership
 - Sensitive to initialization
 - Prone to local optimal



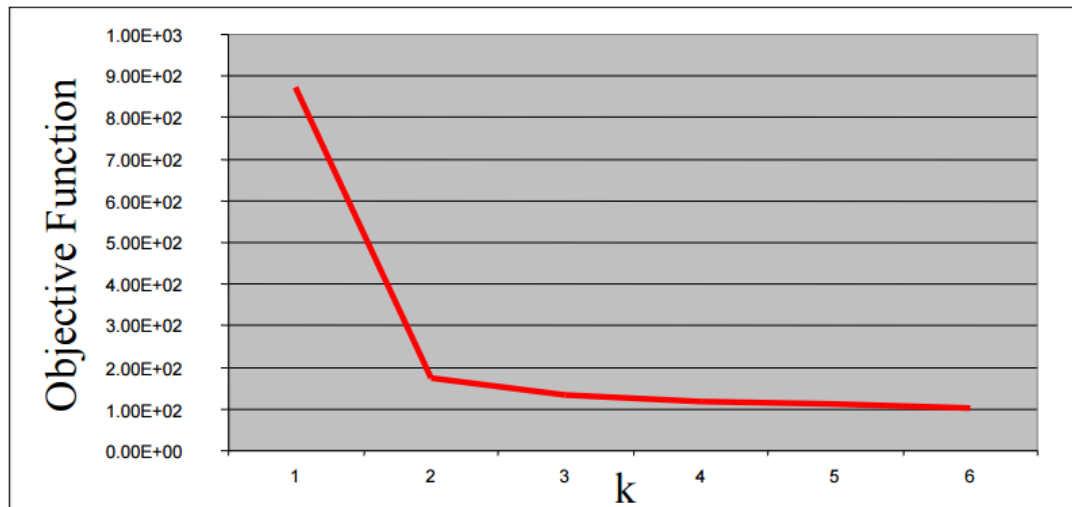
Better initialization: k -means++

1. Choose the first cluster center at uniformly random
2. Repeat until all k centers have been found
 - For each instance compute $D_x = \min_k d(x, c_k)$
 - Choose a new cluster center with probability $p(x) \propto D_x^2$ ← *new center should be far away from existing centers*
3. Run k -means with selected centers as initialization



How to determine k

- Vary k to optimize clustering criterion
 - Internal v.s. external validation
 - Cross validation
 - Abrupt change in objective function



How to determine k

- Vary k to optimize clustering criterion
 - Internal v.s. external validation
 - Cross validation
 - Abrupt change in objective function
 - Model selection criterion – penalizing too many clusters
 - AIC, BIC

