CS6220: DATA MINING TECHNIQUES

Matrix Data: Clustering: Part 2

Instructor: Yizhou Sun

yzsun@ccs.neu.edu

February 17, 2016

Methods to Learn

	Matrix Data	Text Data	Set Data	Sequence Data	Time Series	Graph & Network	Images
Classification	Decision Tree; Naïve Bayes; Logistic Regression SVM; kNN			HMM		Label Propagation	Neural Network
Clustering	K-means; hierarchical clustering; DBSCAN; **Mixture Models; kernel k-means***	PLSA				SCAN; Spectral Clustering	
Frequent Pattern Mining			Apriori; FP-growth	GSP; PrefixSpan			
Prediction	Linear Regression				Autoregression	Collaborative Filtering	
Similarity Search					DTW	P-PageRank	
Ranking						PageRank	

Matrix Data: Clustering: Part 2

- Revisit K-means
- Mixture Model and EM algorithm
- Kernel K-means
- Summary

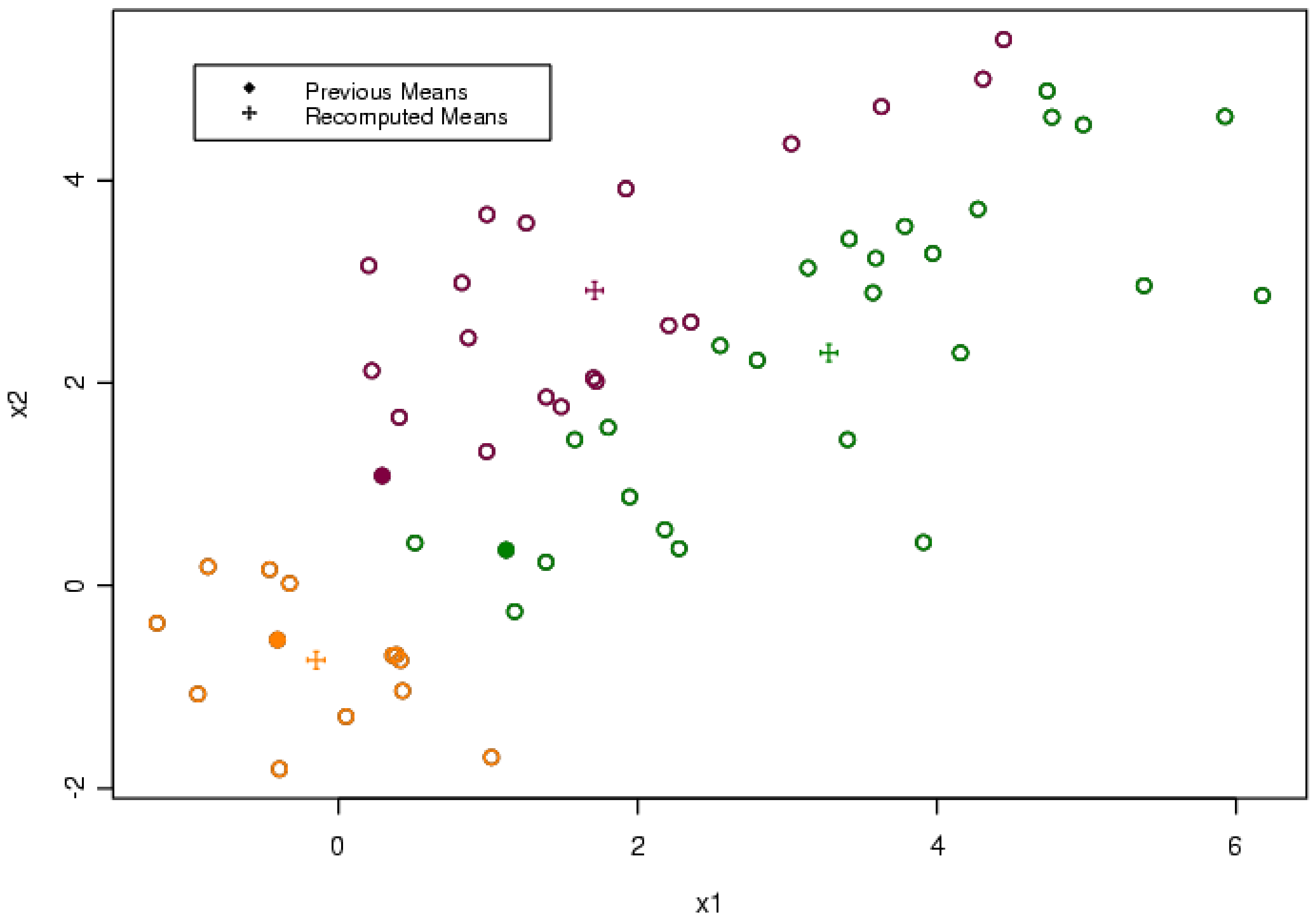
Recall K-Means

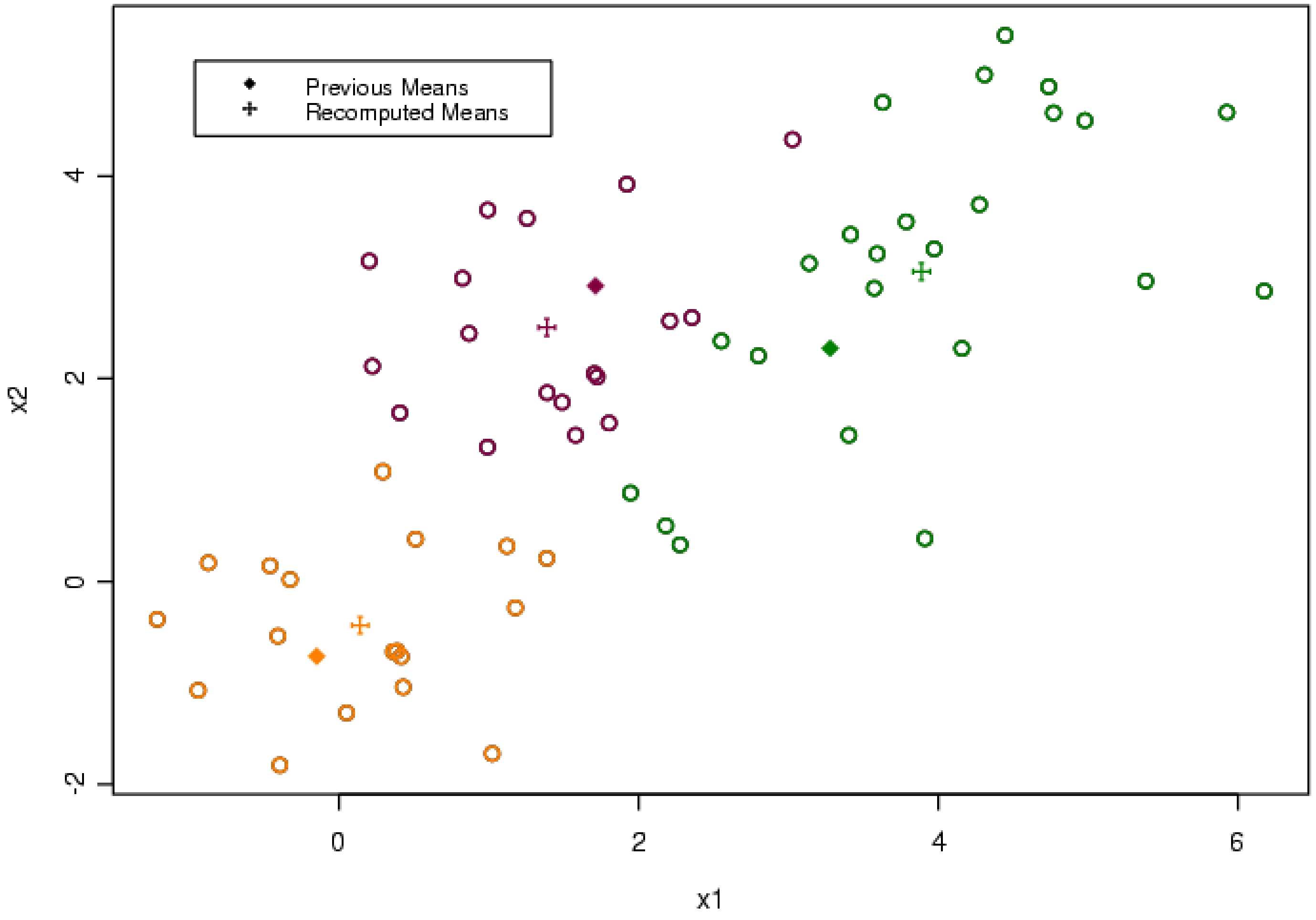
- Objective function
  - $J = \sum_{j=1}^{k} \sum_{C(i)=j} ||x_i - c_j||^2$
  - Total within-cluster variance
- Re-arrange the objective function
  - $J = \sum_{j=1}^{k} \sum_{i} w_{ij} ||x_i - c_j||^2$
    - $w_{ij} \in \{0,1\}$
    - $w_{ij} = 1, if\ x_i\ belongs\ to\ cluster\ j; w_{ij} = 0, otherwise$
  - Looking for:
    - The best assignment w_{ij}
    - The best center c_j

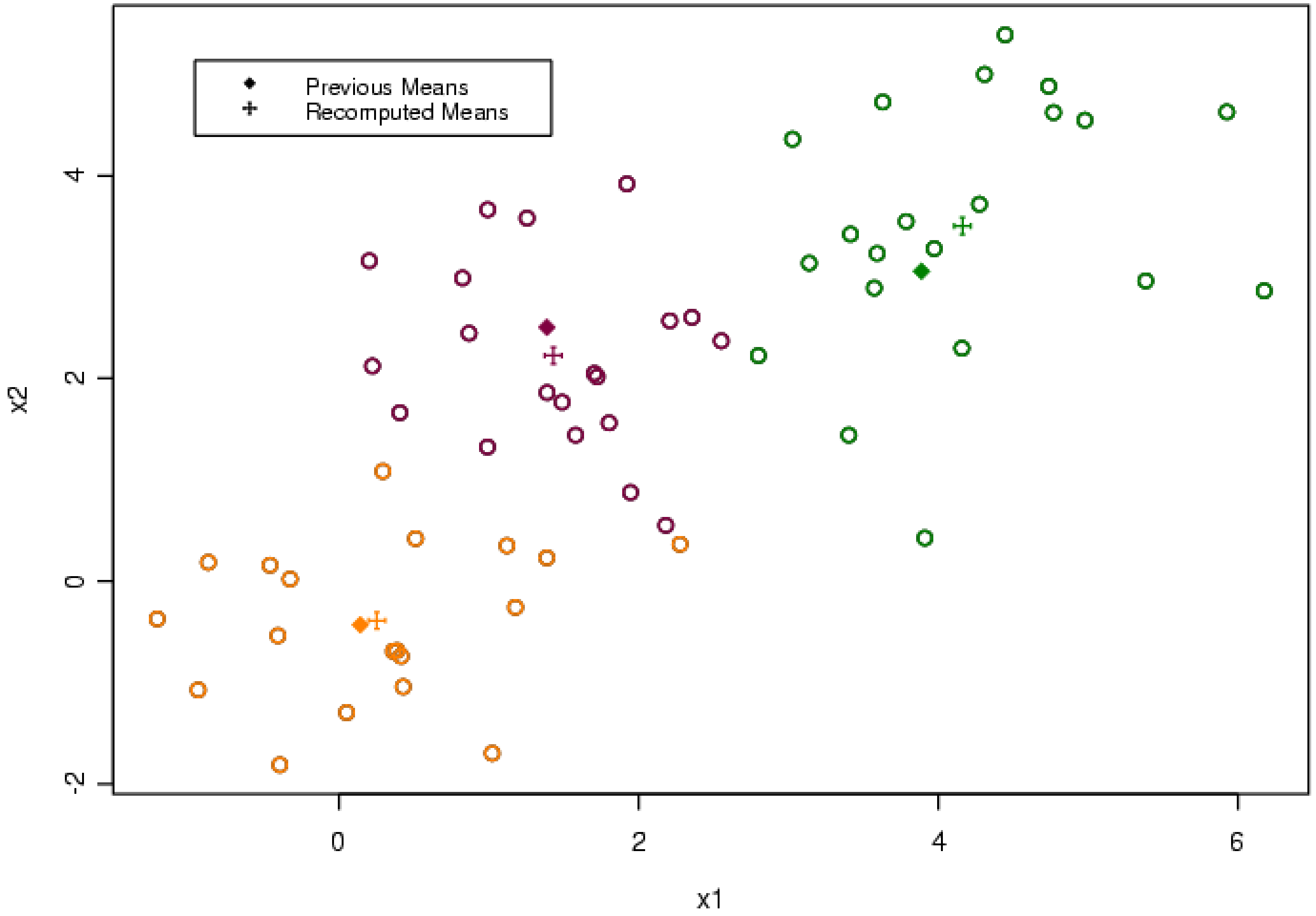
Solution of K-Means

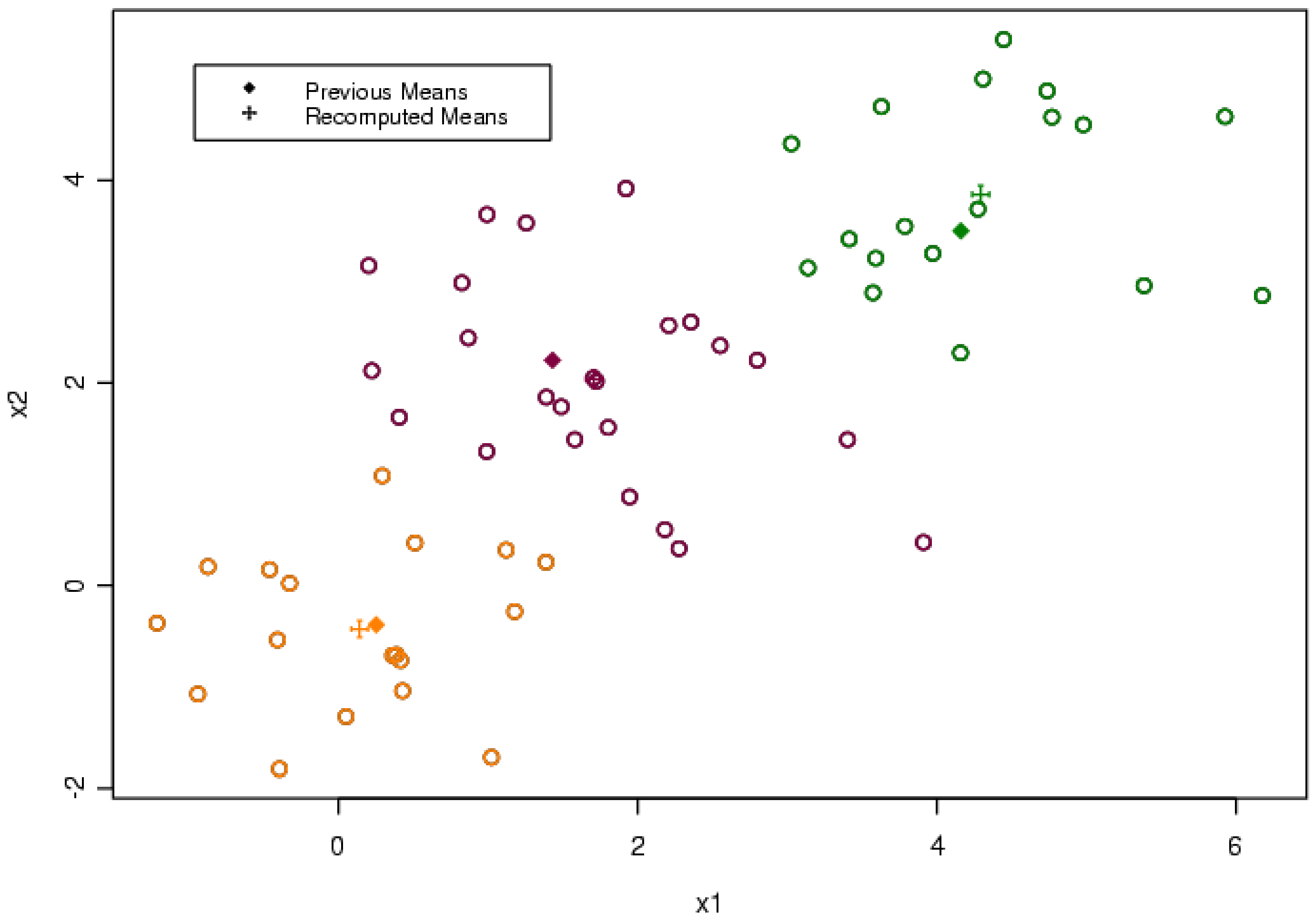
$$J = \sum_{j=1}^{k} \sum_{i} w_{ij} ||x_i - c_j||^2$$

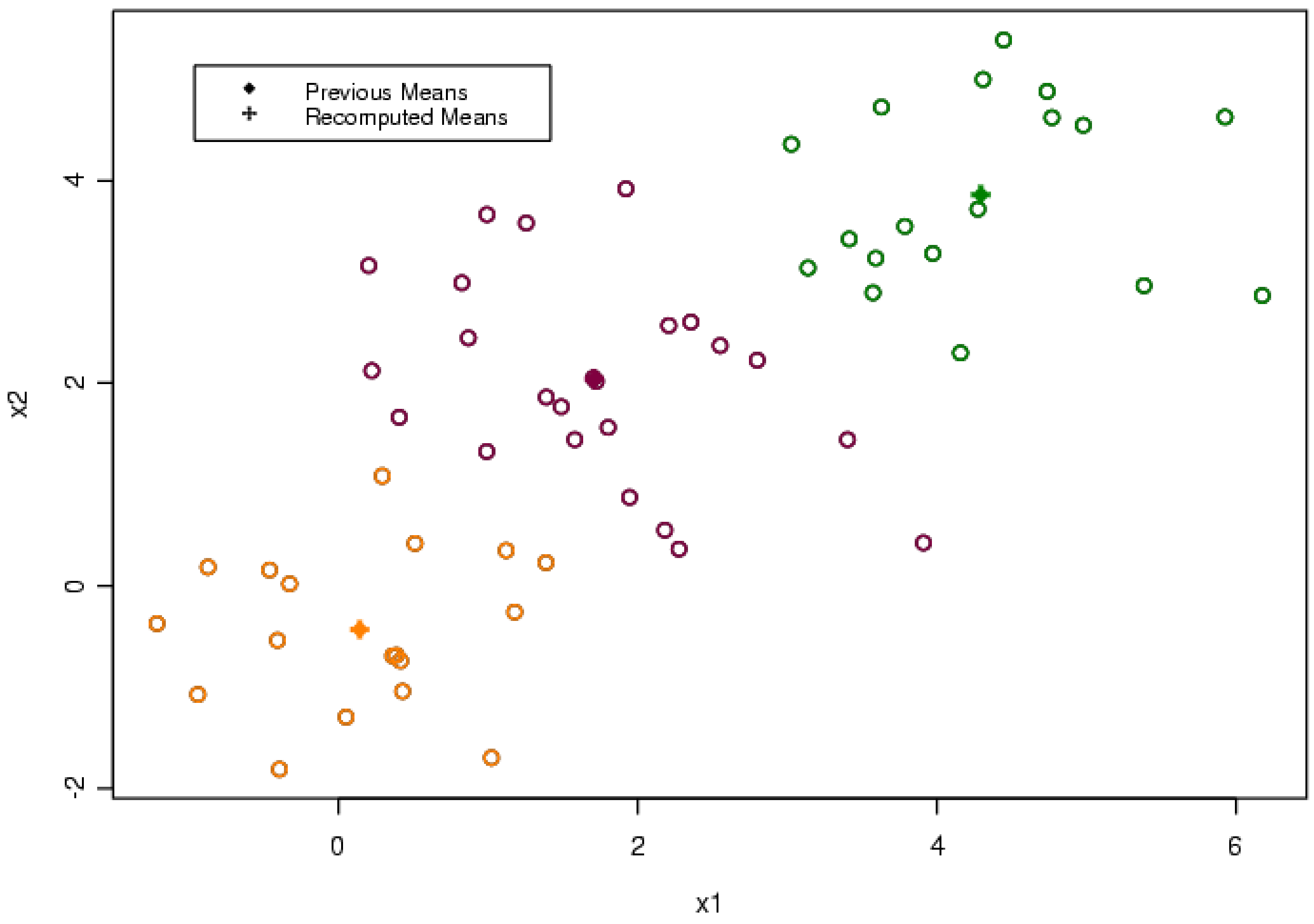
- Iterations
  - Step 1: Fix centers c_j, find assignment w_{ij} that minimizes J
    - => $w_{ij} = 1, if\ ||x_i - c_j||^2$ is the smallest
  - Step 2: Fix assignment w_{ij}, find centers that minimize J
    - => first derivative of J = 0
    - => $\frac{\partial J}{\partial c_j} = -2 \sum_i w_{ij}(x_i - c_j) = 0$
    - =>$c_j = \frac{\sum_i w_{ij} x_i}{\sum_i w_{ij}}$
      - Note $\sum_i w_{ij}$ is the total number of objects in cluster j


Previous Means
Recomputed Means
x2
x1
-2
0
2
4
6

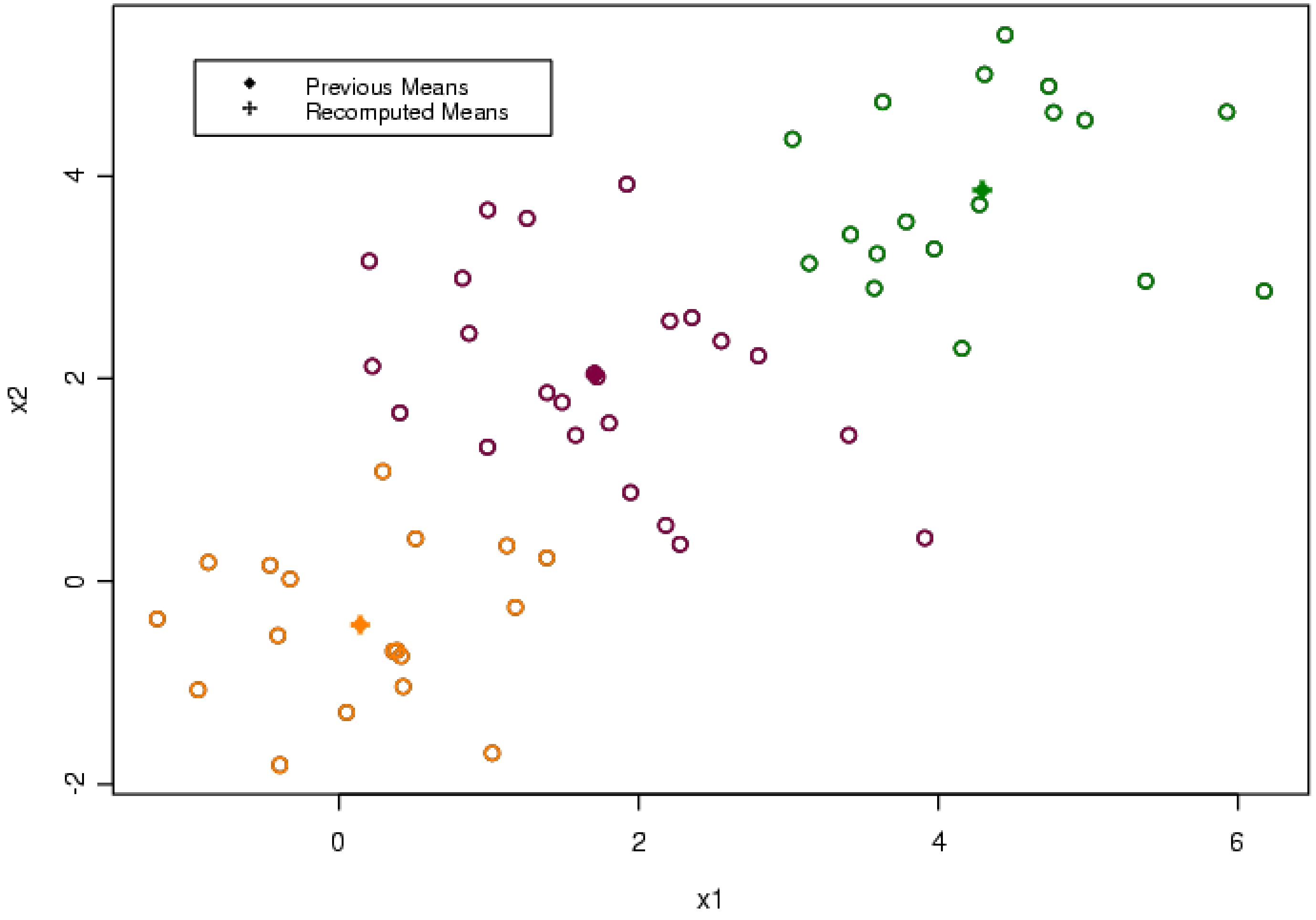

Previous Means
Recomputed Means
x1
x2
-2
0
2
4
6


Previous Means
Recomputed Means
x1
x2
0
2
4
6
-2
0
2
4


Previous Means
Recomputed Means
x2
x1
4
2
0
-2
0
2
4
6



Previous Means
Recomputed Means
x2
x1
-2
0
2
4
0
2
4
6

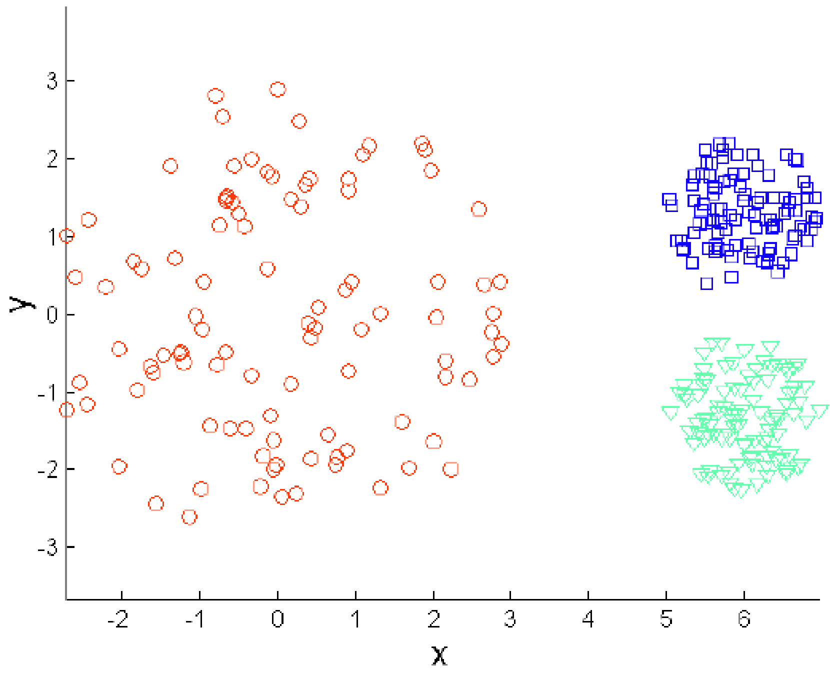


Converges! Why?

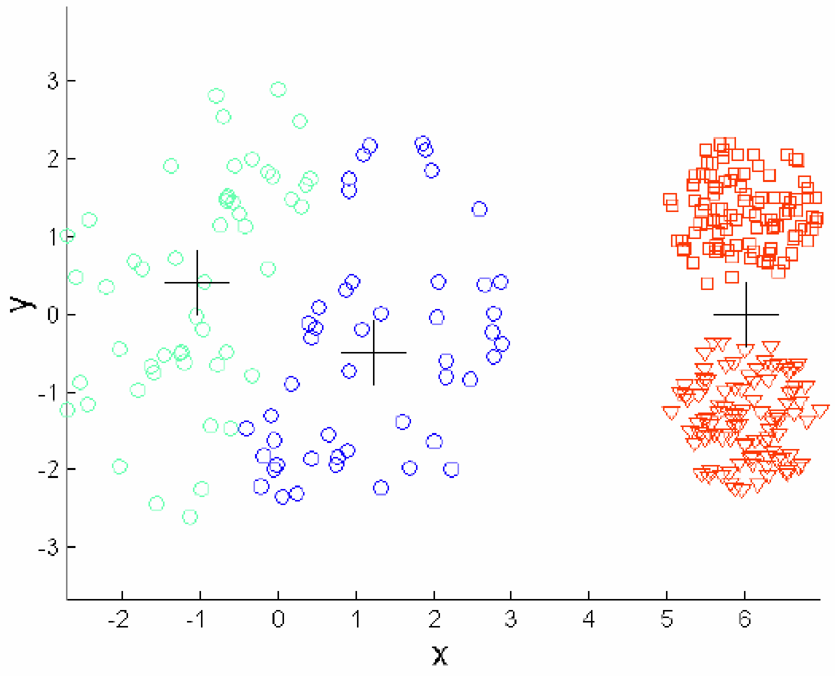
Limitations of K-Means

- K-means has problems when clusters are of different
  - Sizes
  - Densities
  - Non-Spherical Shapes

Limitations of K-Means: Different Density and Size

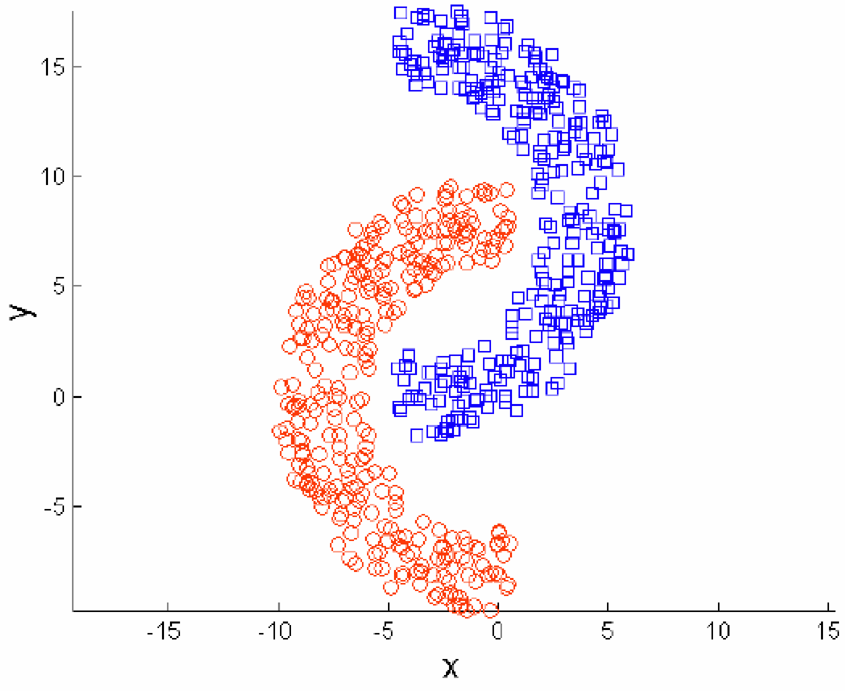


Original Points

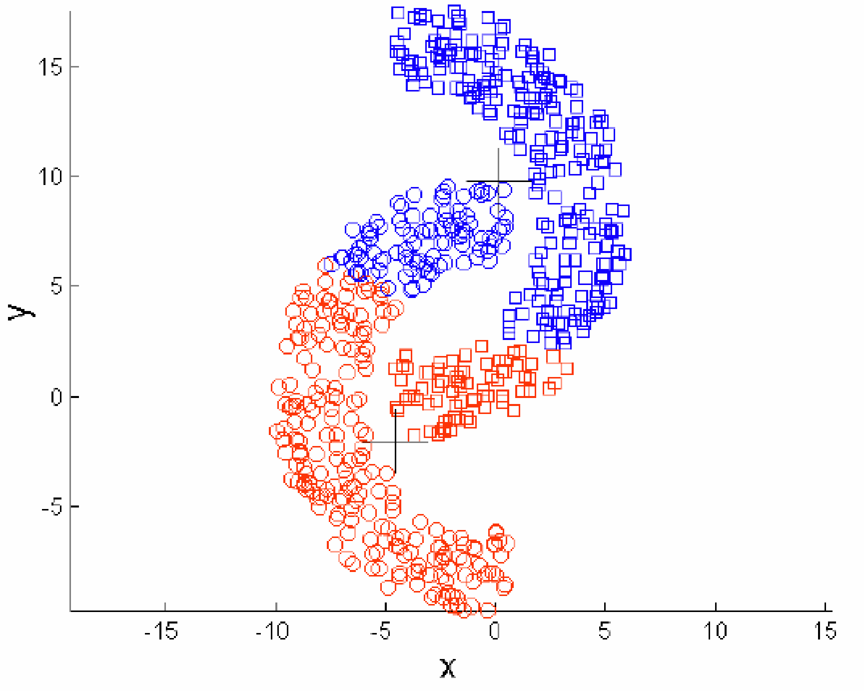


K-means (3 Clusters)

Limitations of K-Means: Non-Spherical Shapes



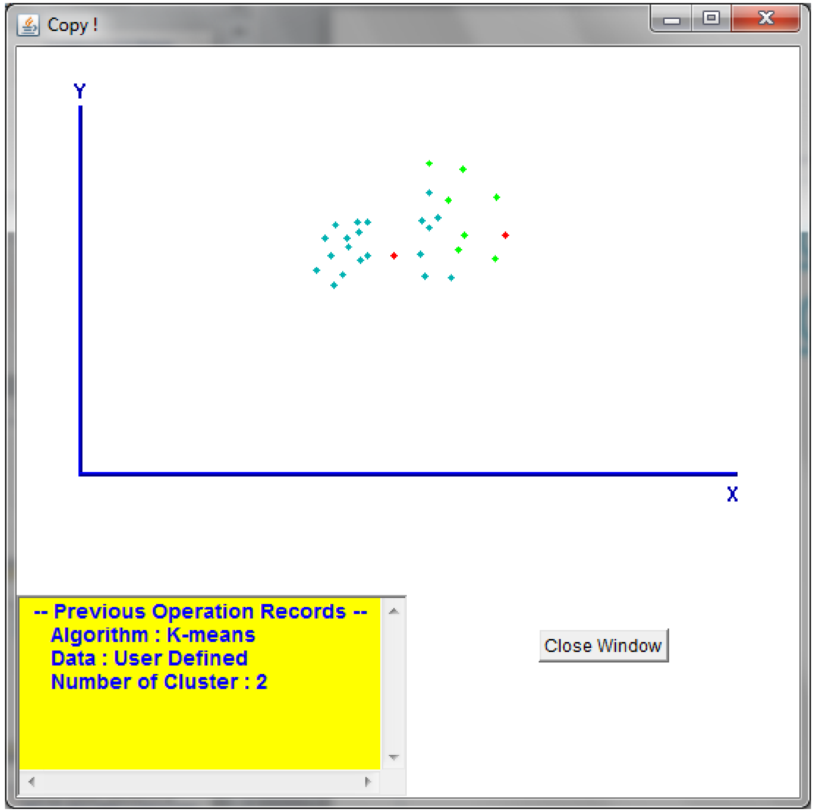
Original Points

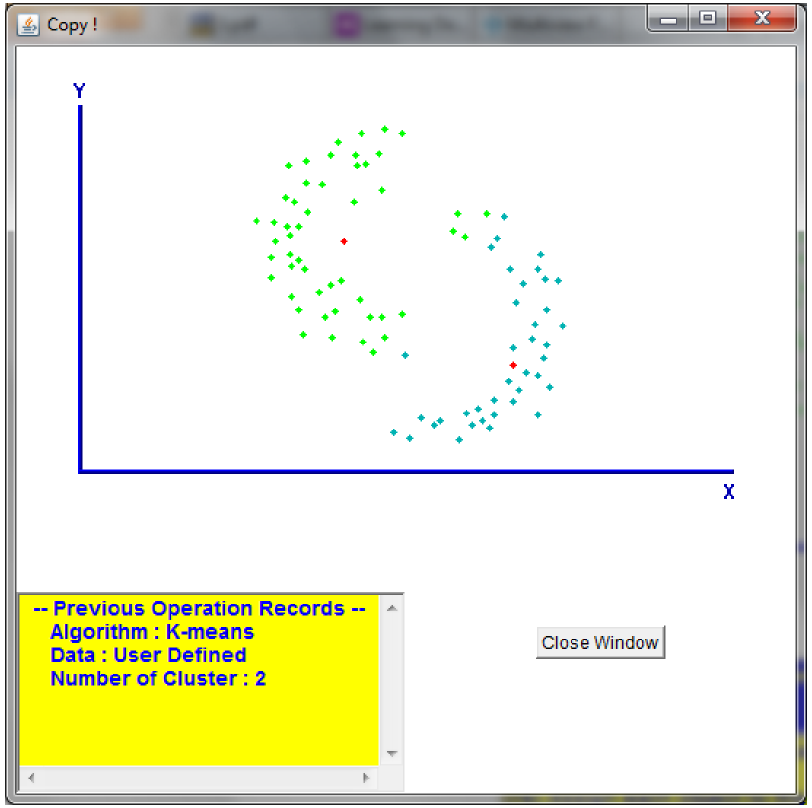


K-means (2 Clusters)

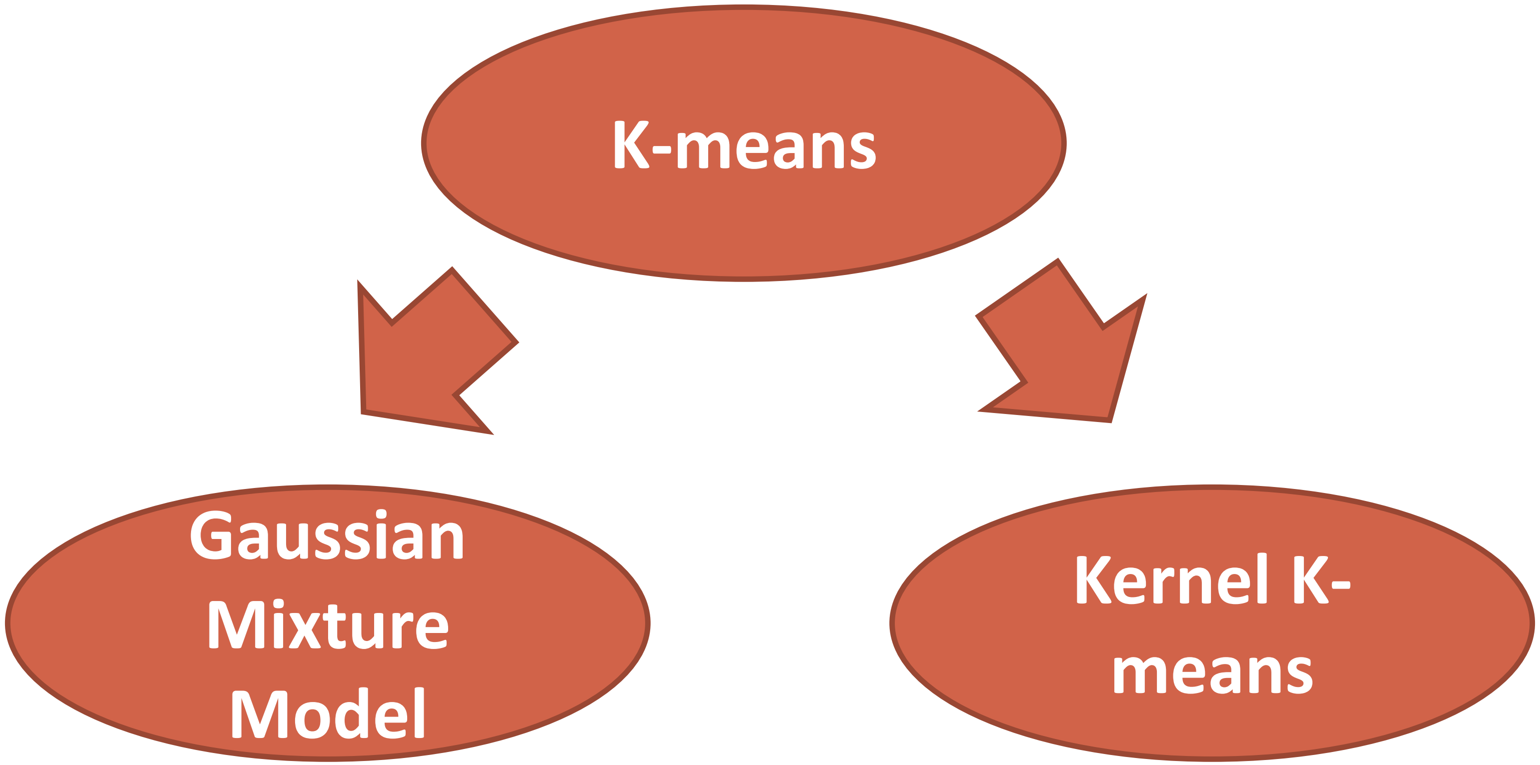
Demo

- http://webdocs.cs.ualberta.ca/~yaling/Cluster/Applet/Code/Cluster.html





Connections of K-means to Other Methods



Matrix Data: Clustering: Part 2

- Revisit K-means
- Mixture Model and EM algorithm
- Kernel K-means
- Summary

Fuzzy Set and Fuzzy Cluster

- Clustering methods discussed so far
  - Every data object is assigned to exactly one cluster
- Some applications may need for fuzzy or soft cluster assignment
  - Ex. An e-game could belong to both entertainment and software
- Methods: fuzzy clusters and probabilistic model-based clusters
- Fuzzy cluster: A fuzzy set S: $F_S : X \rightarrow [0, 1]$ (value between 0 and 1)

Mixture Model-Based Clustering

- A set C of k probabilistic clusters $C_1, \ldots, C_k$
  - probability density functions: $f_1, \ldots, f_k$,
  - Cluster prior probabilities: $w_1, \ldots, w_k$, $\sum_j w_j = 1$
- Probability of an object i generated by cluster C_j is:
  - $P(x_i, z_i = C_j) = w_j f_j(x_i)$
- Probability of i generated by the set of cluster $\boldsymbol{C}$ is:
  - $P(x_i) = \sum_j w_j f_j(x_i)$

Maximum Likelihood Estimation

- Since objects are assumed to be generated independently, for a data set D = {x_1, …, x_n}, we have,

$$P(D) = \prod_i P(x_i) = \prod_i \sum_j w_j f_j(x_i)$$

- Task: Find a set *C* of *k* probabilistic clusters s.t. *P*(*D*) is maximized

The EM (Expectation Maximization) Algorithm

- **The (EM) algorithm:** A framework to approach maximum likelihood or maximum a posteriori estimates of parameters in statistical models.
  - **E-step** assigns objects to clusters according to the current fuzzy clustering or parameters of probabilistic clusters
    - $w_{ij}^t = p\left(z_i = j \middle| \theta_j^t, x_i\right) \propto p\left(x_i \middle| C_j^t, \theta_j^t\right) p(C_j^t)$
  - **M-step** finds the new clustering or parameters that maximize the expected likelihood

Gaussian Mixture Model

- Generative model
  - For each object:
    - Pick its distribution component: $Z \sim Multi(w_1, \ldots, w_k)$
    - Sample a value from the selected distribution: $X \sim N\left(\mu_Z, \sigma_Z^2\right)$
- Overall likelihood function
  - $L(D|\,\theta) = \prod_i \sum_j w_j p(x_i|\mu_j, \sigma_j^2)$

    s.t. $\sum_j w_j = 1\ and\ w_j \geq 0$
  - Q: What is θ here?

Estimating Parameters

- $L(D;\theta) = \sum_i \log \sum_j w_j p(x_i|\mu_j, \sigma_j^2)$

Intractable!

- Considering the first derivative of μ_j:
- $\frac{\partial L}{\partial u_j} = \sum_i \frac{w_j}{\sum_j w_j p(x_i|\mu_j,\sigma_j^2)} \frac{\partial p(x_i|\mu_j,\sigma_j^2)}{\partial \mu_j}$
- $= \sum_i \frac{w_j p(x_i|\mu_j,\sigma_j^2)}{\sum_j w_j p(x_i|\mu_j,\sigma_j^2)} \frac{1}{p(x_i|\mu_j,\sigma_j^2)} \frac{\partial p(x_i|\mu_j,\sigma_j^2)}{\partial \mu_j}$
- $= \sum_i \boxed{\frac{w_j p(x_i|\mu_j,\sigma_j^2)}{\sum_j w_j p(x_i|\mu_j,\sigma_j^2)}} \boxed{\frac{\partial logp(x_i|\mu_j,\sigma_j^2)}{\partial u_j}}$

$w_{ij} = P(Z = j|X = x_i, \theta)$

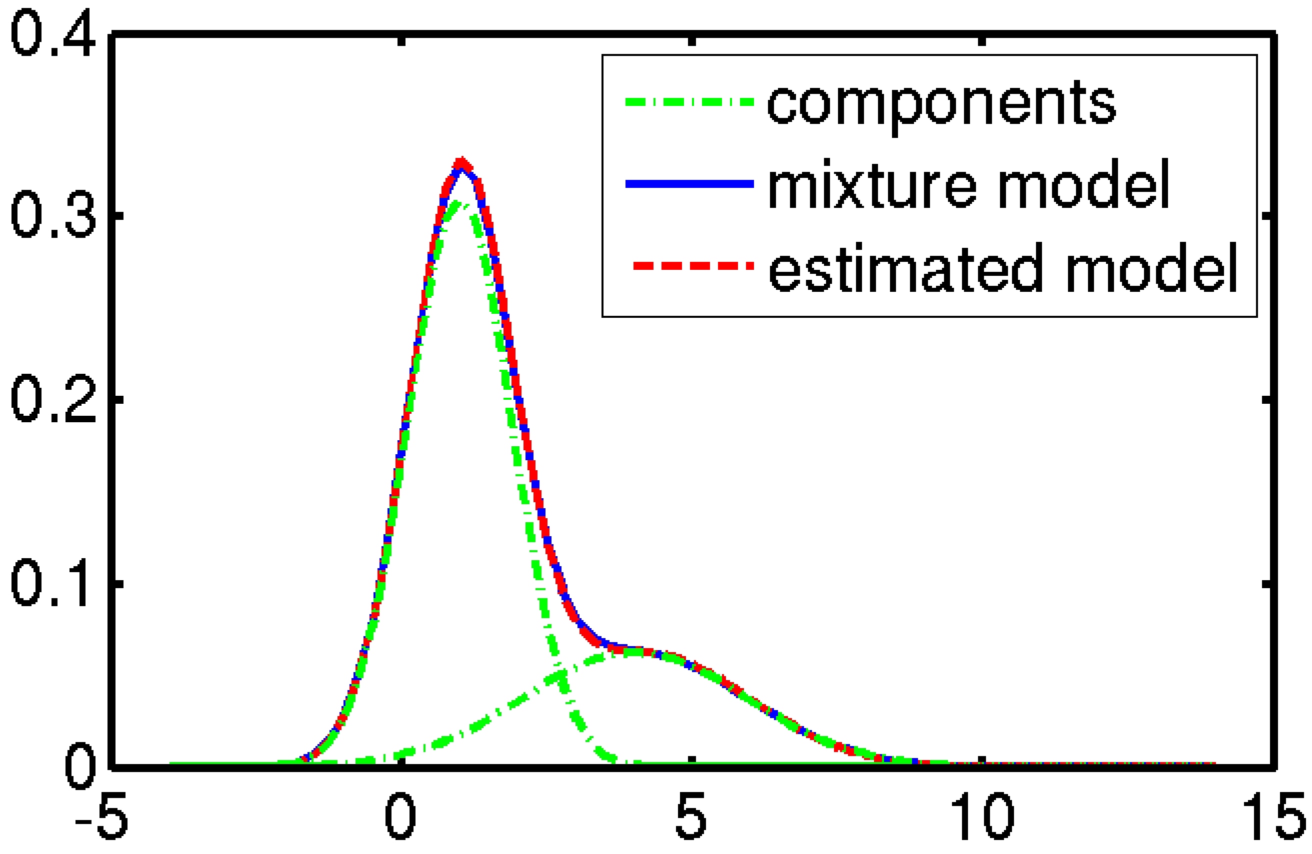
$\partial l(x_i)/\partial \mu_j$

Like weighted likelihood estimation;
But the weight is determined by the parameters!

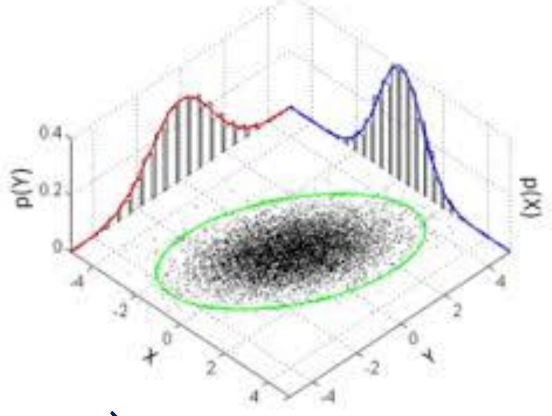
Apply EM algorithm: 1-d

- An iterative algorithm (at iteration t+1)
  - E(expectation)-step
    - Evaluate the weight w_{ij} when μ_j, σ_j, w_j are given
    - $w_{ij}^t = \frac{w_j^t p(x_i|\mu_j^t, (\sigma_j^2)^t)}{\sum_j w_j^t p(x_i|\mu_j^t, (\sigma_j^2)^t)}$
  - M(maximization)-step
    - Evaluate μ_j, σ_j, w_j when w_{ij}'s are given that maximize the weighted likelihood
    - It is equivalent to Gaussian distribution parameter estimation when each point has a weight belonging to each distribution
    - $\mu_j^{t+1} = \frac{\sum_i w_{ij}^t x_i}{\sum_i w_{ij}^t}; (\sigma_j^2)^{t+1} = \frac{\sum_i w_{ij}^t \left\|x_i - \mu_j^t\right\|^2}{\sum_i w_{ij}^t}; w_j^{t+1} \propto \sum_i w_{ij}^t$

Example: 1-D GMM



2-d Gaussian



- Bivariate Gaussian distribution
  - Two dimensional random variable: $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N(\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma(X_1, X_2) \\ \sigma(X_1, X_2) & \sigma_2^2 \end{pmatrix})$$

  - μ_1 *and* μ_2 are means of $X_1 and\ X_2$
  - σ_1 *and* σ_2 *are* standard deviations of $X_1 and\ X_2$
  - $\sigma(X_1, X_2)\ is\ the\ covariance\ between\ X_1 and\ X_2, i.e., \sigma(X_1, X_2) = E(X_1 - \mu_1)(X_2 - \mu_2)$

Apply EM algorithm: 2-d

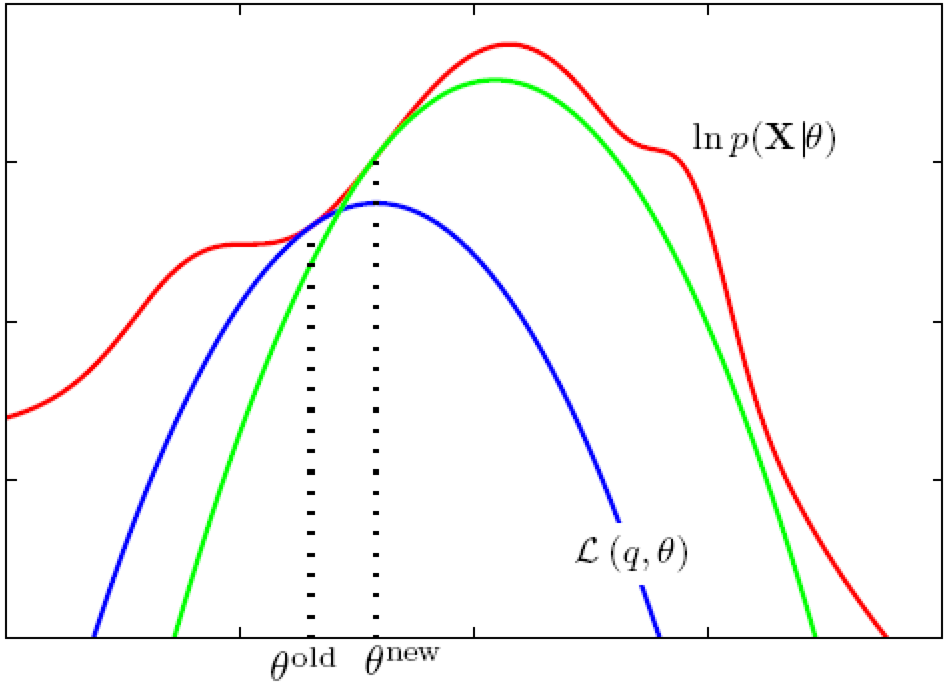
- An iterative algorithm (at iteration t+1)
  - E(expectation)-step
    - Evaluate the weight w_{ij} when $\boldsymbol{\mu}_j, \Sigma_j, w_j$ are given
    - $w_{ij}^t = \frac{w_j^t p(\boldsymbol{x}_i | \boldsymbol{\mu}_j^t, \Sigma_j^t)}{\sum_j w_j^t p(\boldsymbol{x}_i | \boldsymbol{\mu}_j^t, \Sigma_j^t)}$
  - M(maximization)-step
    - Evaluate $\boldsymbol{\mu}_j, \Sigma_j, w_j$ when w_{ij}'s are given that maximize the weighted likelihood
    - It is equivalent to Gaussian distribution parameter estimation when each point has a weight belonging to each distribution
      - $\boldsymbol{\mu}_j^{t+1} = \frac{\sum_i w_{ij}^t \boldsymbol{x}_i}{\sum_i w_{ij}^t}$; $(\sigma_{j,1}^2)^{t+1} = \frac{\sum_i w_{ij}^t \left|\left| x_{i,1} - \mu_{j,1}^t \right|\right|^2}{\sum_i w_{ij}^t}$; $(\sigma_{j,2}^2)^{t+1} = \frac{\sum_i w_{ij}^t \left|\left| x_{i,2} - \mu_{j,2}^t \right|\right|^2}{\sum_i w_{ij}^t}$;
      - $(\sigma(X_1, X_2)_j)^{t+1} = \frac{\sum_i w_{ij}^t (x_{i,1} - \mu_{j,1}^t)(x_{i,2} - \mu_{j,2}^t)}{\sum_i w_{ij}^t}$; $w_j^{t+1} \propto \sum_i w_{ij}^t$

K-Means: A Special Case of Gaussian Mixture Model

- When each Gaussian component with covariance matrix $\sigma^2 I$
  - Soft K-means
  - $p(x_i | \mu_j, \sigma^2) \propto \exp\{-\boxed{(x_i - \mu_j)^2}/\sigma^2\}$ Distance!
- When $\sigma^2 \to 0$
  - Soft assignment becomes hard assignment
  - $w_{ij} \to 1, if\ x_i$ is closest to μ_j (why?)

*Why EM Works?

- E-Step: computing a tight lower bound f of the original objective function at θ_{old}
- M-Step: find θ_{new} to maximize the lower bound
- $l(\theta_{new}) \geq f(\theta_{new}) \geq f(\theta_{old}) = l(\theta_{old})$



*How to Find Tight Lower Bound?

- $$\begin{aligned} \ell(\theta) &= \log \sum_h p(d,h;\theta) \\ &= \log \sum_h \frac{q(h)}{q(h)} p(d,h;\theta) \\ &= \log \sum_h q(h) \frac{p(d,h;\theta)}{q(h)} \end{aligned}$$

$q(h)$: the tight lower bound we want to get

- Jensen's inequality
  - $$\log \sum_h q(h) \frac{p(d,h;\theta)}{q(h)} \geq \sum_h q(h) \log \frac{p(d,h;\theta)}{q(h)}$$
  - When "=" holds to get a tight lower bound?
    - $q(h) = p(h|d,\theta)$ (why?)

Advantages and Disadvantages of GMM

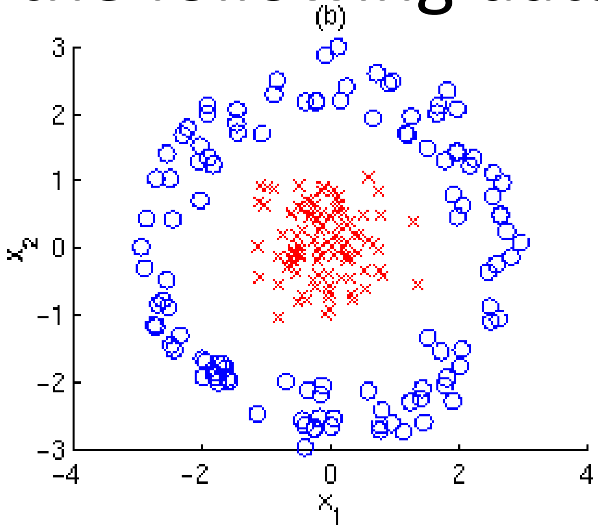
- Strength
  - Mixture models are more general than partitioning: different densities and sizes of clusters
  - Clusters can be characterized by a small number of parameters
  - The results may satisfy the statistical assumptions of the generative models
- Weakness
  - Converge to local optimal (overcome: run multi-times w. random initialization)
  - Computationally expensive if the number of distributions is large, or the data set contains very few observed data points
  - Hard to estimate the number of clusters
  - Can only deal with spherical clusters

Matrix Data: Clustering: Part 2

- Revisit K-means
- Mixture Model and EM algorithm
- Kernel K-means
- Summary

*Kernel K-Means

- How to cluster the following data?



- A non-linear map: $\phi: R^n \to F$
  - Map a data point into a higher/infinite dimensional space
  - $x \to \phi(x)$
- Dot product matrix K_{ij}
  - $K_{ij} =< \phi(x_i), \phi(x_j) >$

Typical Kernel Functions

- Recall kernel SVM:

$$\text{Polynomial kernel of degree } h: \quad K(\boldsymbol{X_i}, \boldsymbol{X_j}) = (\boldsymbol{X_i} \cdot \boldsymbol{X_j} + 1)^h$$

$$\text{Gaussian radial basis function kernel}: \quad K(\boldsymbol{X_i}, \boldsymbol{X_j}) = e^{-\|\boldsymbol{X_i} - \boldsymbol{X_j}\|^2 / 2\sigma^2}$$

$$\text{Sigmoid kernel}: \quad K(\boldsymbol{X_i}, \boldsymbol{X_j}) = \tanh(\kappa \boldsymbol{X_i} \cdot \boldsymbol{X_j} - \delta)$$

Solution of Kernel K-Means

- Objective function under new feature space:
  - $J = \sum_{j=1}^{k} \sum_i w_{ij} ||\phi(x_i) - c_j||^2$
- Algorithm
  - By fixing assignment w_{ij}
    - $c_j = \sum_i w_{ij}\, \phi(x_i) / \sum_i w_{ij}$
  - In the assignment step, assign the data points to the closest center
    - $d(x_i, c_j) = \left\| \phi(x_i) - \frac{\sum_{i'} w_{i'j} \phi(x_{i'})}{\sum_{i'} w_{i'j}} \right\|^2 = \phi(x_i) \cdot \phi(x_i) - 2 \frac{\sum_{i'} w_{i'j} \phi(x_i) \cdot \phi(x_{i'})}{\sum_{i'} w_{i'j}} + \frac{\sum_{i'} \sum_l w_{i'j} w_{lj} \phi(x_{i'}) \cdot \phi(x_l)}{(\sum_{i'} w_{i'j})^2}$

Do not really need to know $\phi(x)$, *but only* K_{ij}

Advantages and Disadvantages of Kernel K-Means

- **Advantages**
  - Algorithm is able to identify the non-linear structures.
- **Disadvantages**
  - Number of cluster centers need to be predefined.
  - Algorithm is complex in nature and time complexity is large.
- **References**
  - Kernel k-means and Spectral Clustering by Max Welling.
  - Kernel k-means, Spectral Clustering and Normalized Cut by Inderjit S. Dhillon, Yuqiang Guan and Brian Kulis.
  - An Introduction to kernel methods by Colin Campbell.

Matrix Data: Clustering: Part 2

- Revisit K-means
- Mixture Model and EM algorithm
- Kernel K-means
- Summary

Summary

- Revisit k-means
  - Derivative
- Mixture models
  - Gaussian mixture model; multinomial mixture model; EM algorithm; Connection to k-means
- Kernel k-means*
  - Objective function; solution; connection to k-means