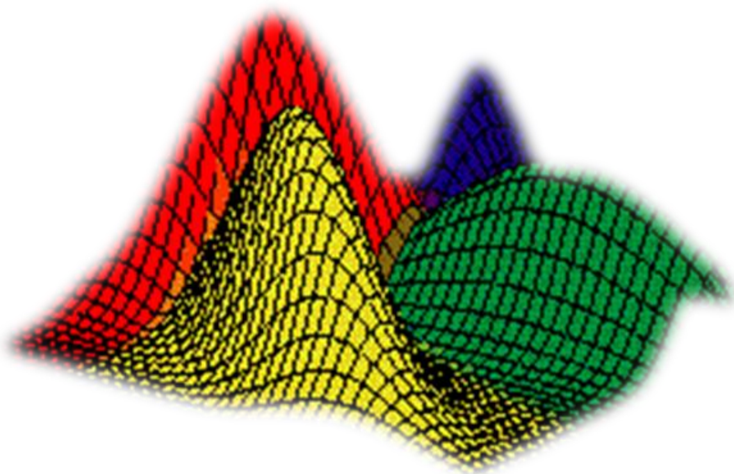


CSE 473: Pattern Recognition

Unsupervised Learning:
Clustering

- Reassignment of vectors

Why necessary?

The problem of sensitivity to the order of data presentation:

"A vector $\underline{x}$ may have been assigned to a cluster C_i at the current stage but another cluster C_j may be formed at a later stage that lies closer to $\underline{x}$"

– A simple reassignment procedure

- For i=1 to N
  - Find C_j such that $d(\underline{x}_i,C_j)=min_{k=1,\dots,m}d(\underline{x}_i,C_k)$
  - Set $b(i)=j$ \{ $b(i)$ is the index of the cluster that lies closet to $\underline{x}_i$ \}
- End {for}
- For j=1 to m
  - Set $C_j=\{\underline{x}_i \in X: b(i)=j\}$
  - If necessary, update representatives
- End {for}

k-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, *k*, must be specified
- The basic algorithm is very simple

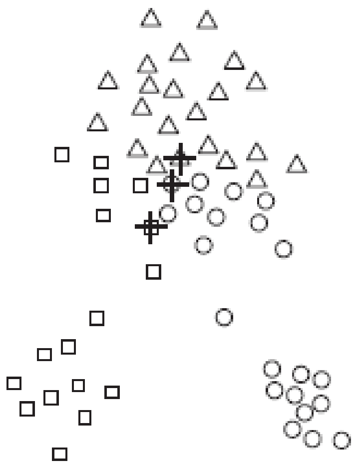
k-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, *k*, must be specified
- The basic algorithm is very simple

1: Select K points as the initial centroids.
2: **repeat**
3:     Form K clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.
5: **until** The centroids don't change

---

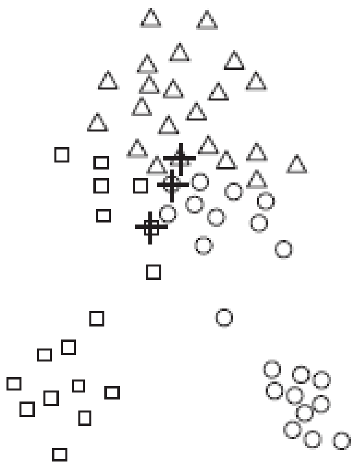
k-means Clustering



Iteration 1.

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.

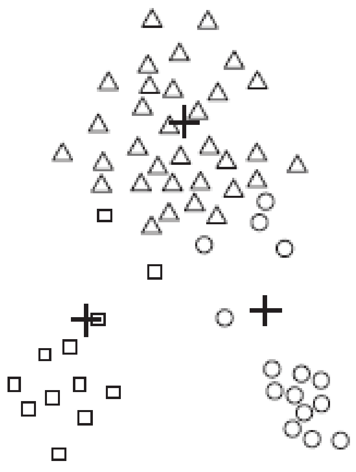
k-means Clustering



Iteration 1.

- Points are distributed to the closest centroid.
- 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.

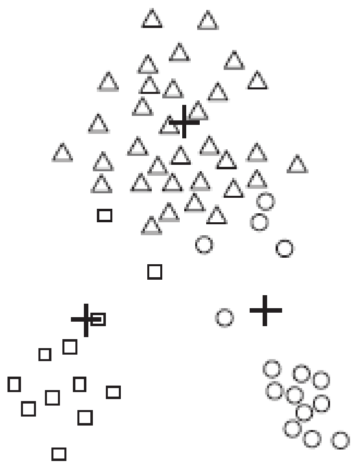
k-means Clustering



Iteration 2.

- centroid is (typically) the mean of the points in the cluster.

K-means Clustering

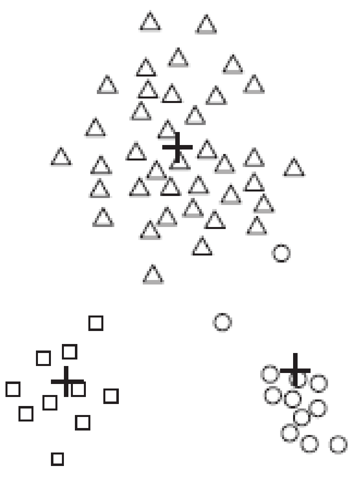


Iteration 2.

- centroid is (typically) the mean of the points in the cluster.

$$c_i = \frac{1}{m_i} \sum_{x \in C_i} x$$

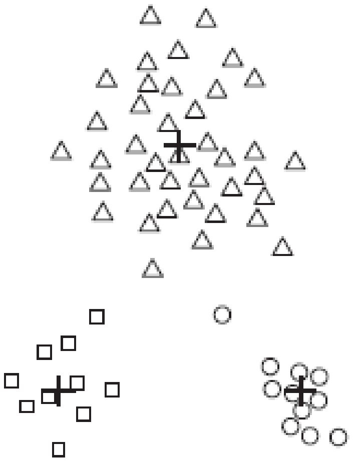
k-means Clustering



Iteration 3.

- *k*-means will converge for common similarity measures

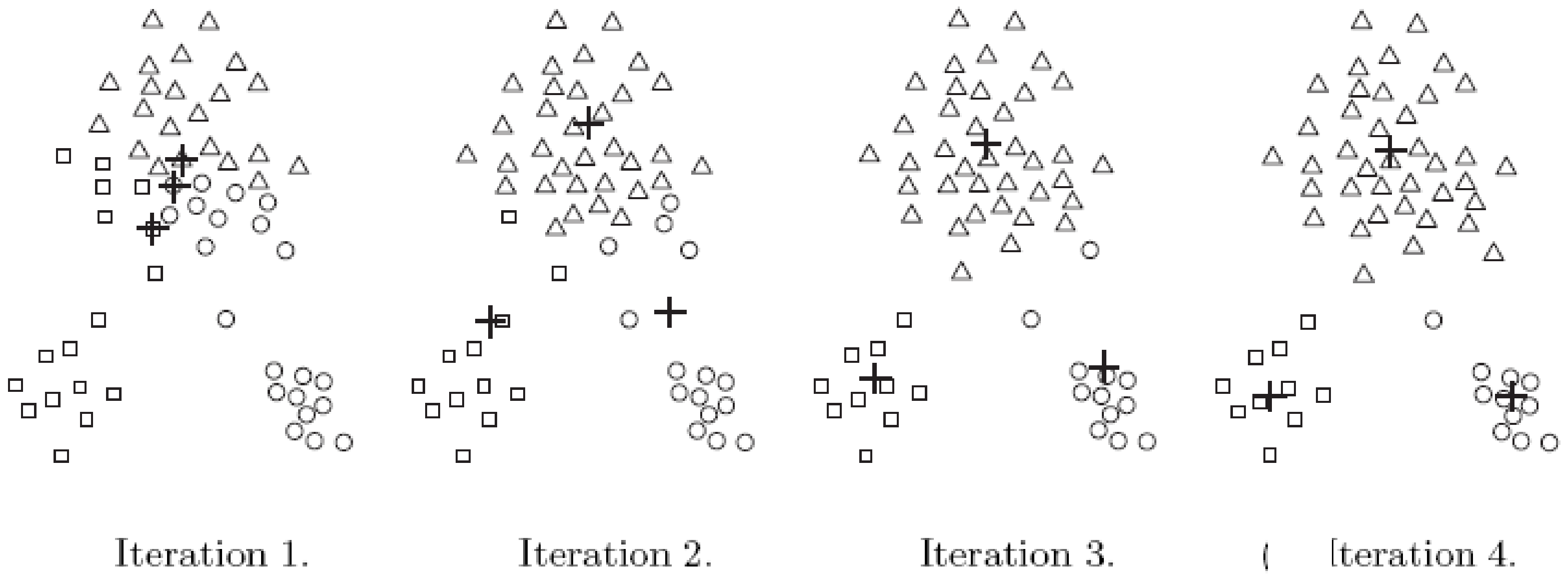
k-means Clustering



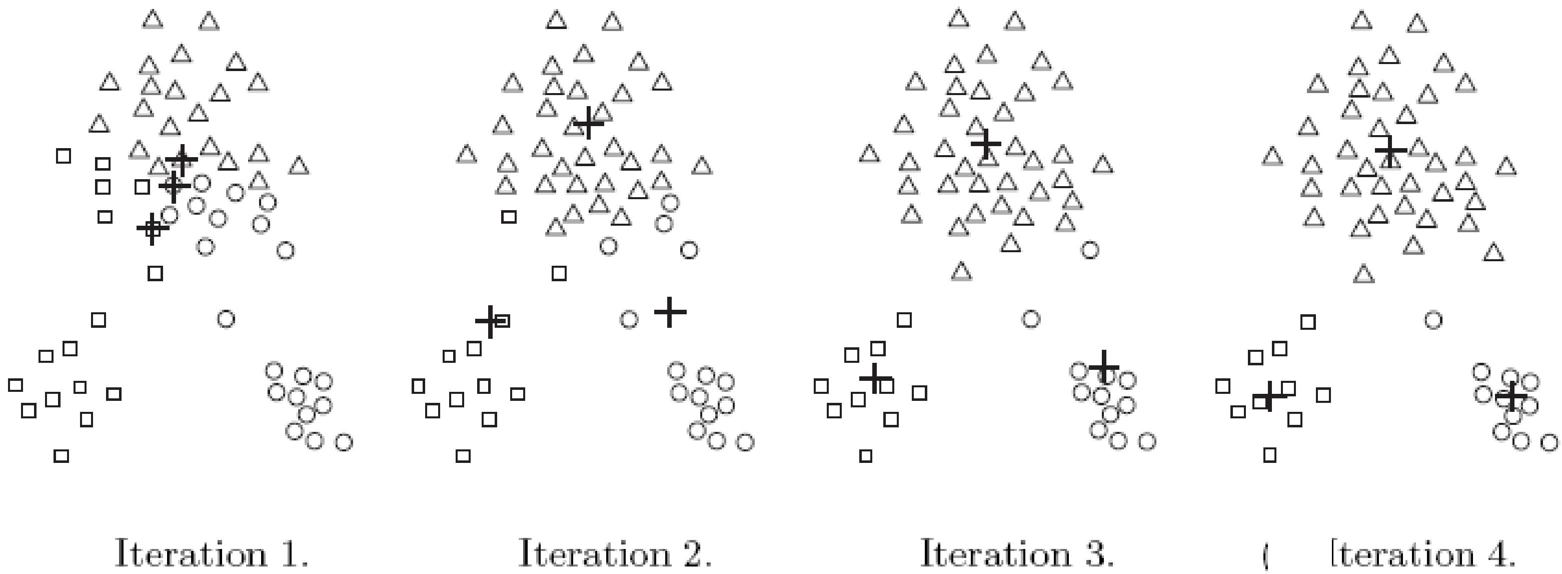
Iteration 4.

- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’

k-means Clustering

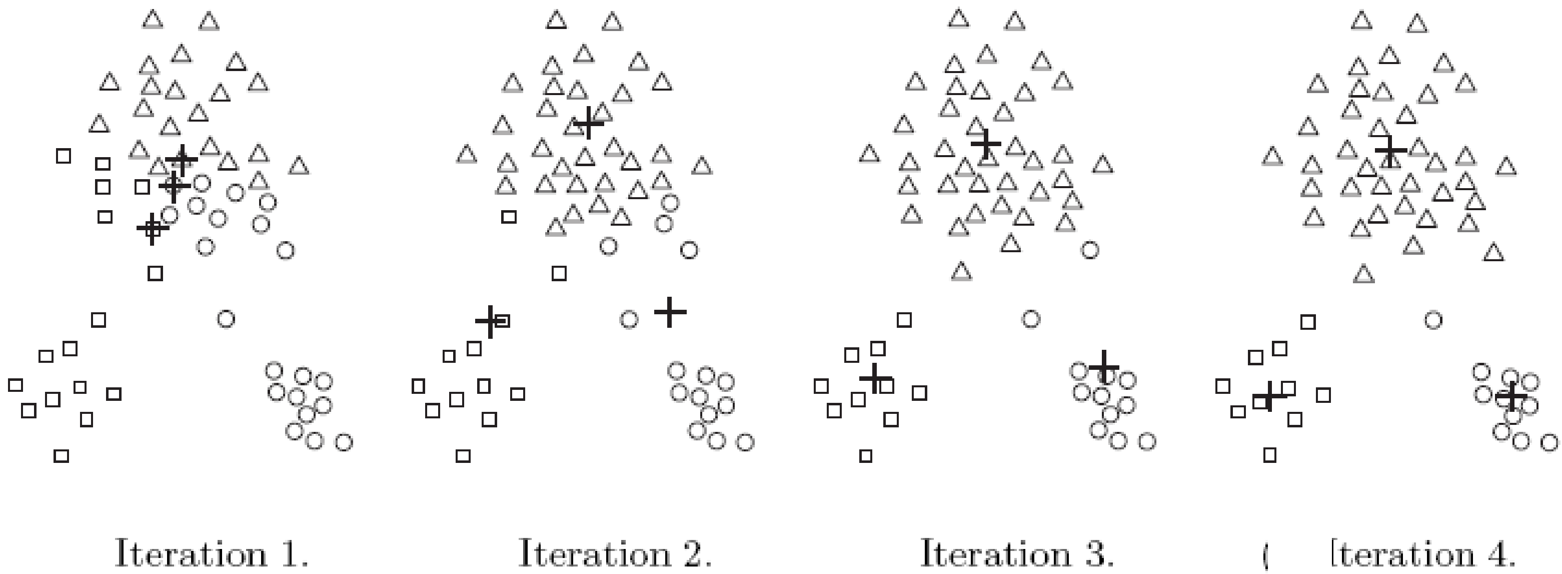


k-means Clustering



- *k*-means try to optimize an objective function

k-means Clustering



- *k*-means try to optimize an objective function

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(c_i, x)$$

Evaluating *k*-means Clusters

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(c_i, x)$$

– Given two clusterings, choose the one with the smallest error

– An easy way to reduce SSE is to increase *k*

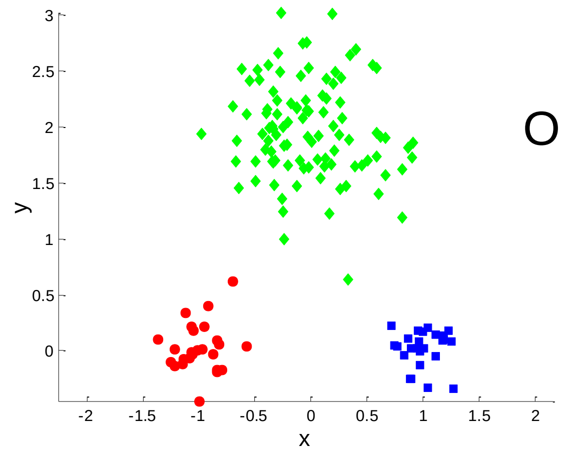
- A good clustering with smaller *k* can have a lower SSE than a poor clustering with higher *k*

k-means Clustering – Complexity

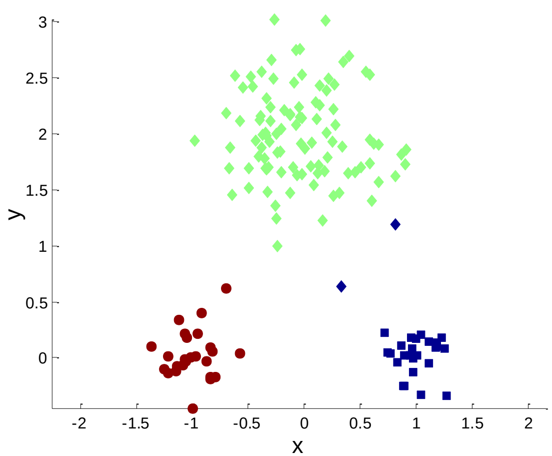
- Storage Complexity is O $((m+k)n)$
  - m = number of points
  - k = number of clusters
  - n = number of attributes

- Time Complexity is $O(I * k * m * n)$
  - I = number of iterations

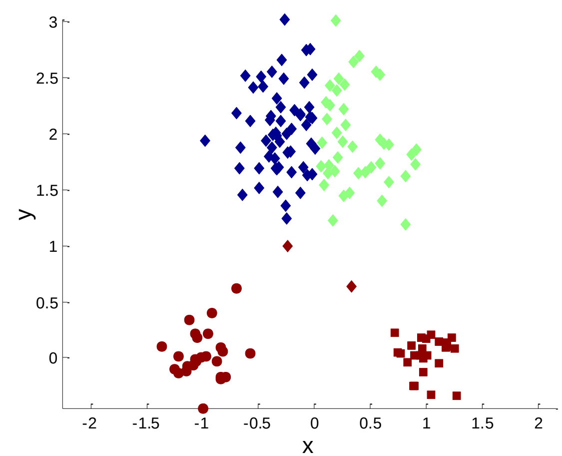
k-means may lead to suboptimal solution



Original Points

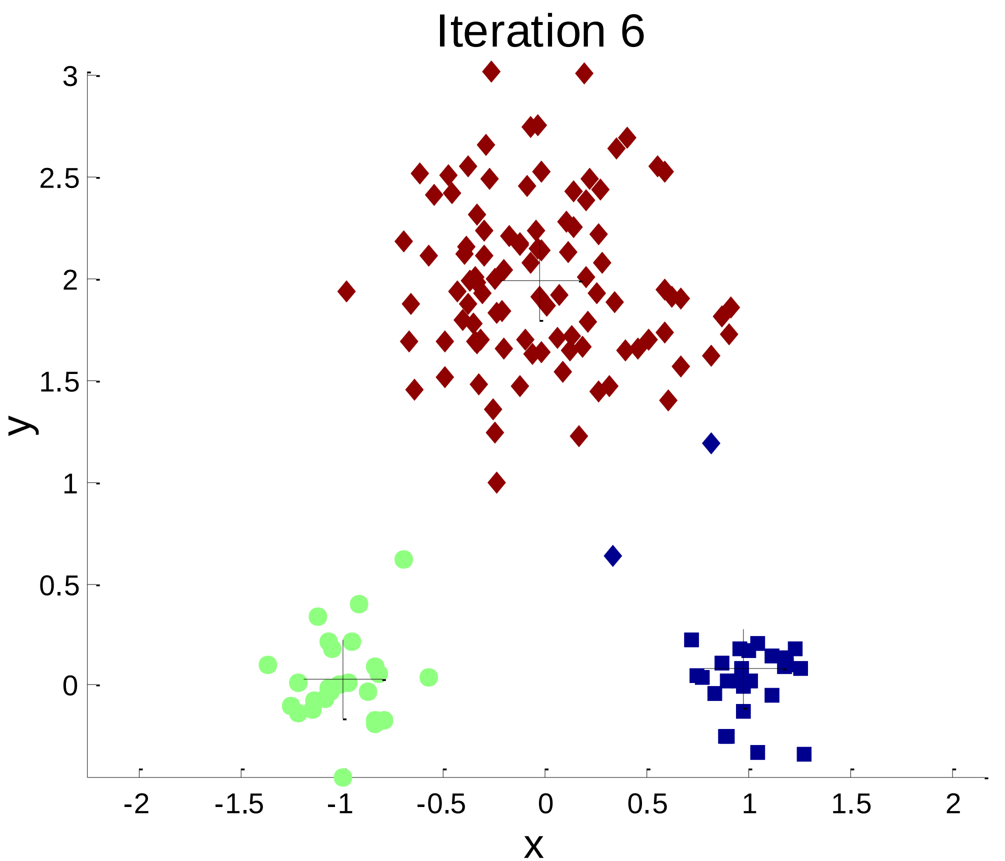


Optimal Clustering

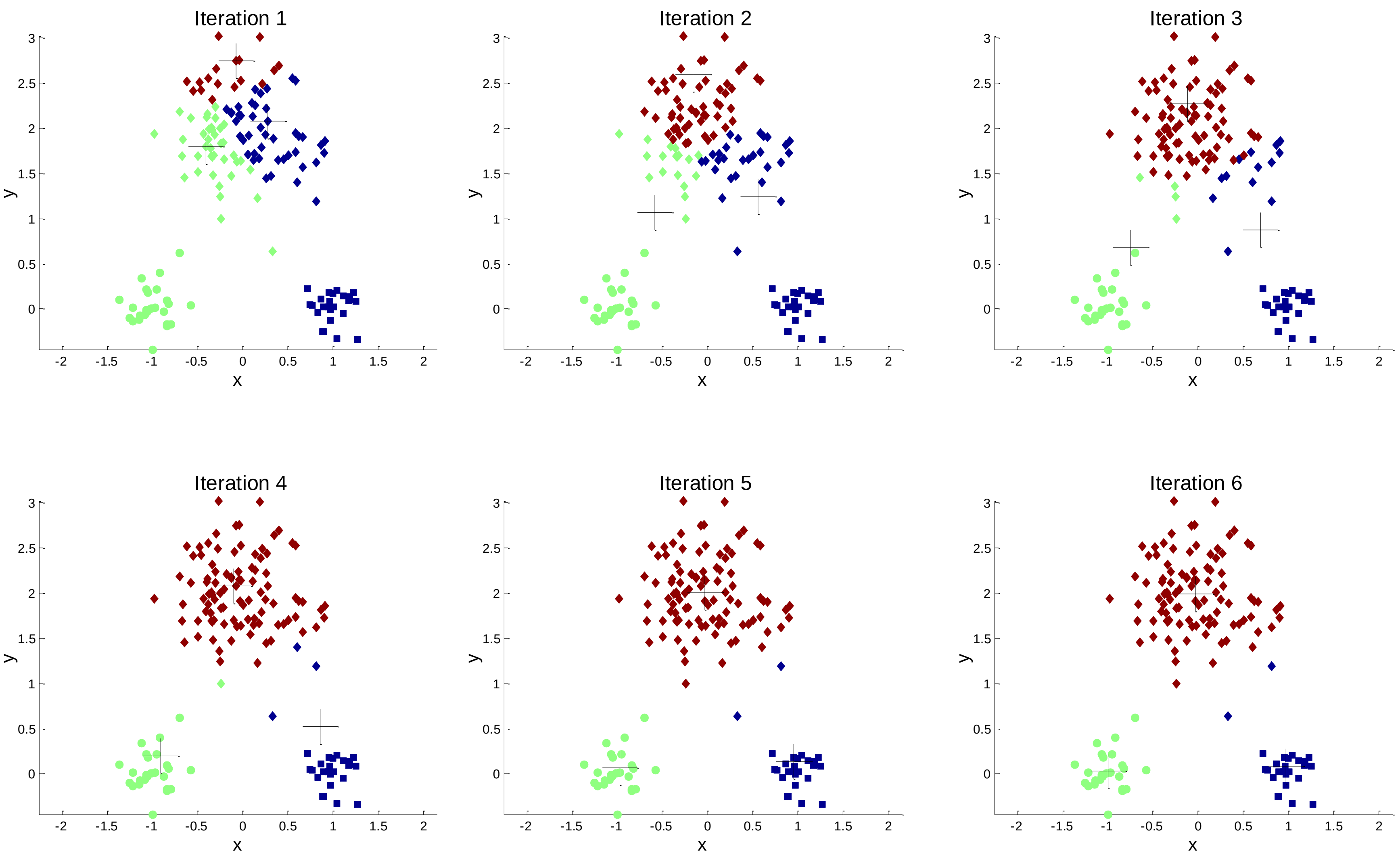


Sub-optimal Clustering

Importance of Choosing Initial Centroids

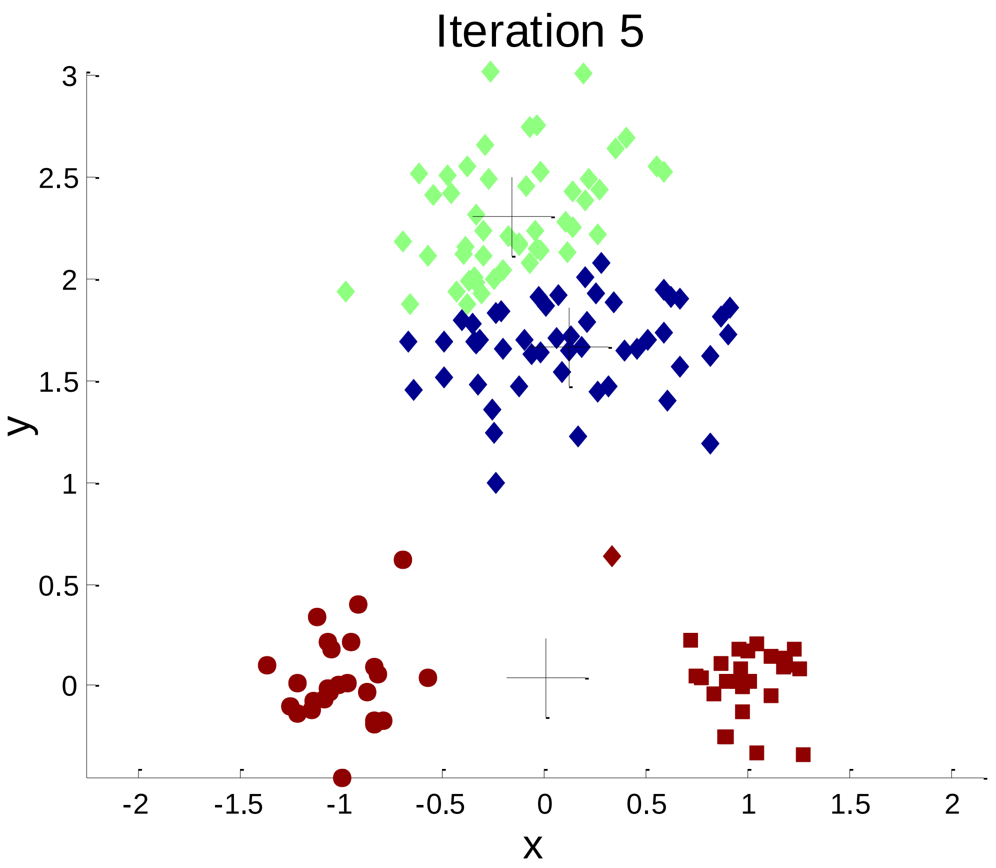


Importance of Choosing Initial Centroids

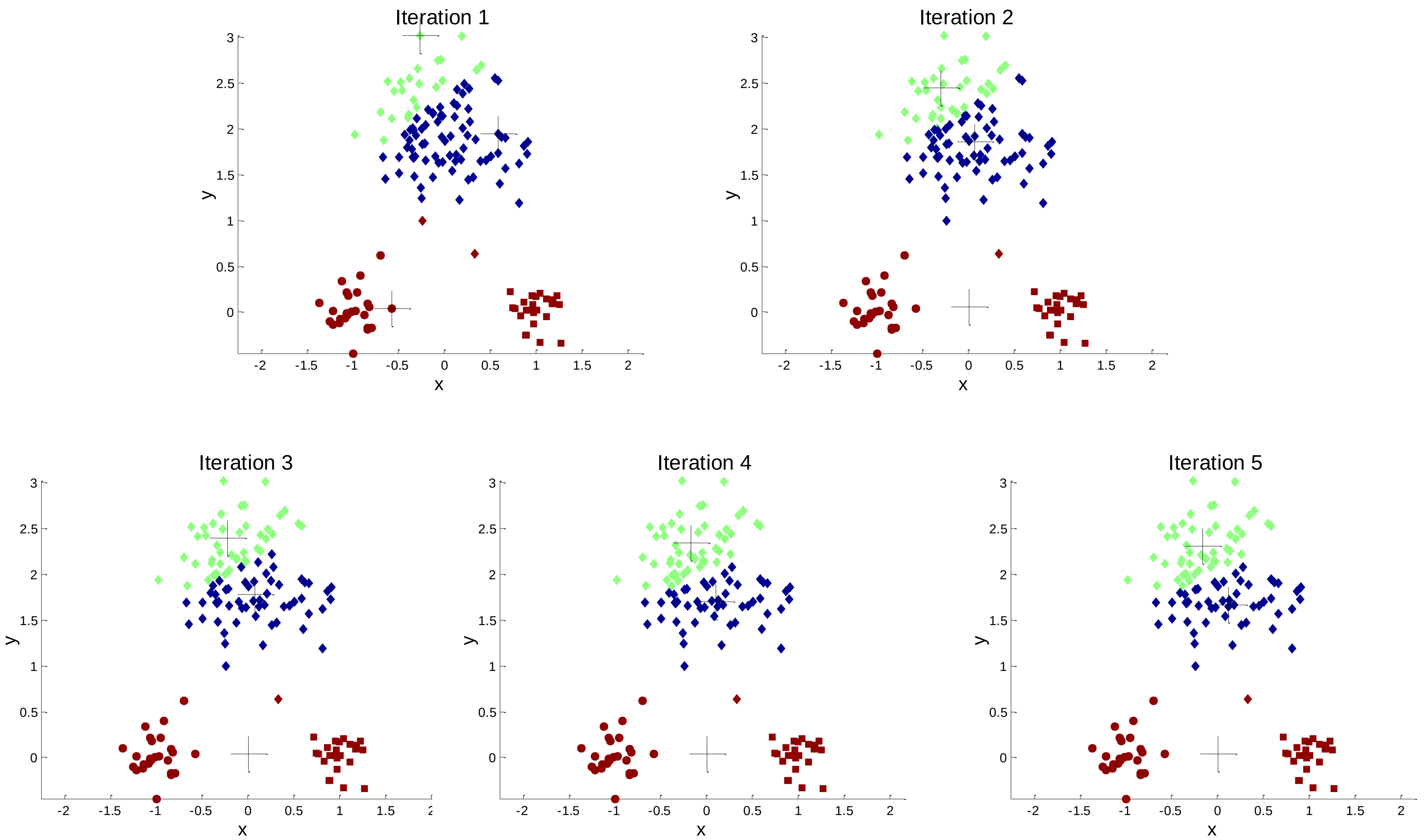


- find optimal SSE, although initial centroids are from one natural cluster

Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …

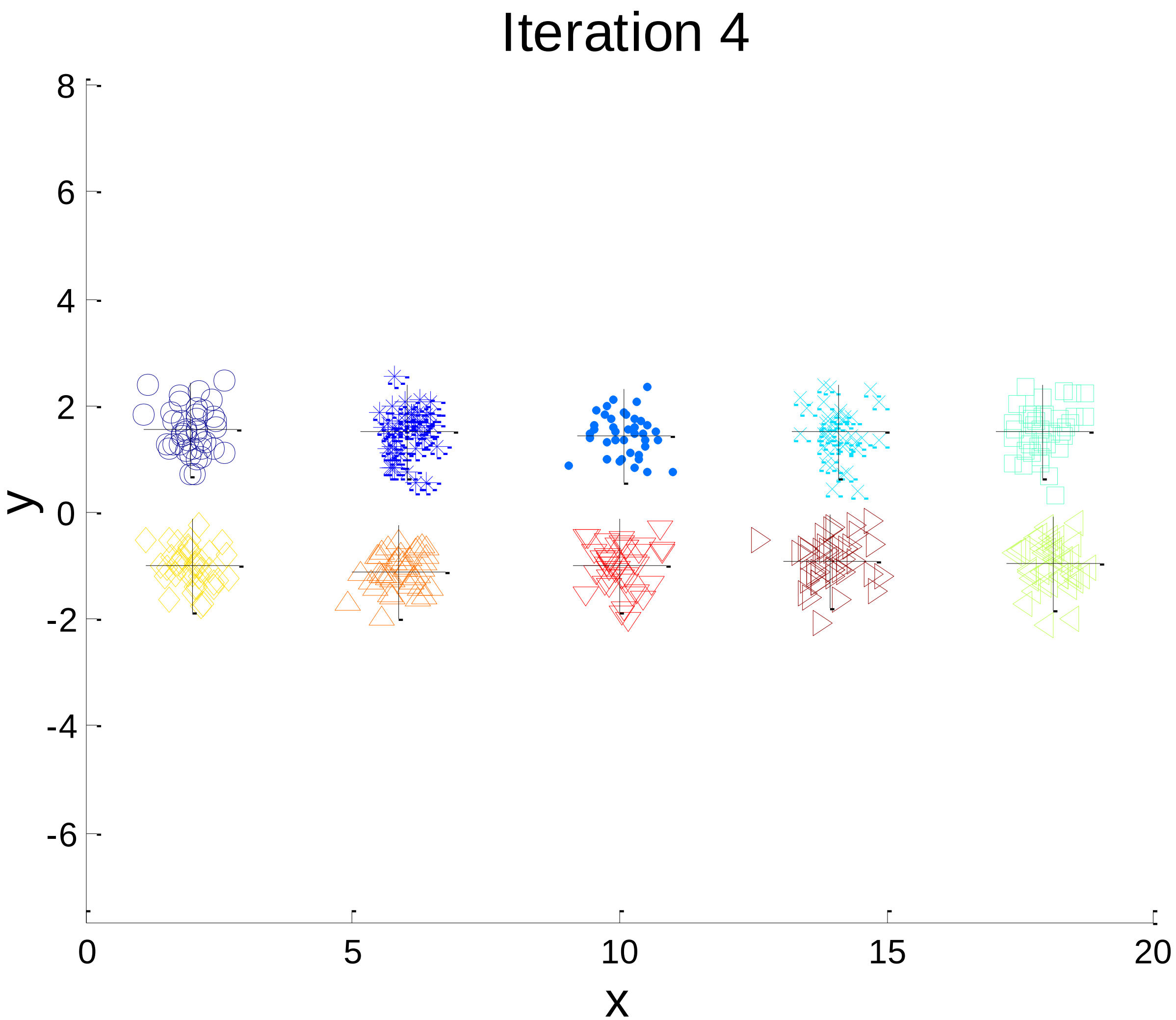


- cannot find optimal SSE

Problems with Selecting Initial Points

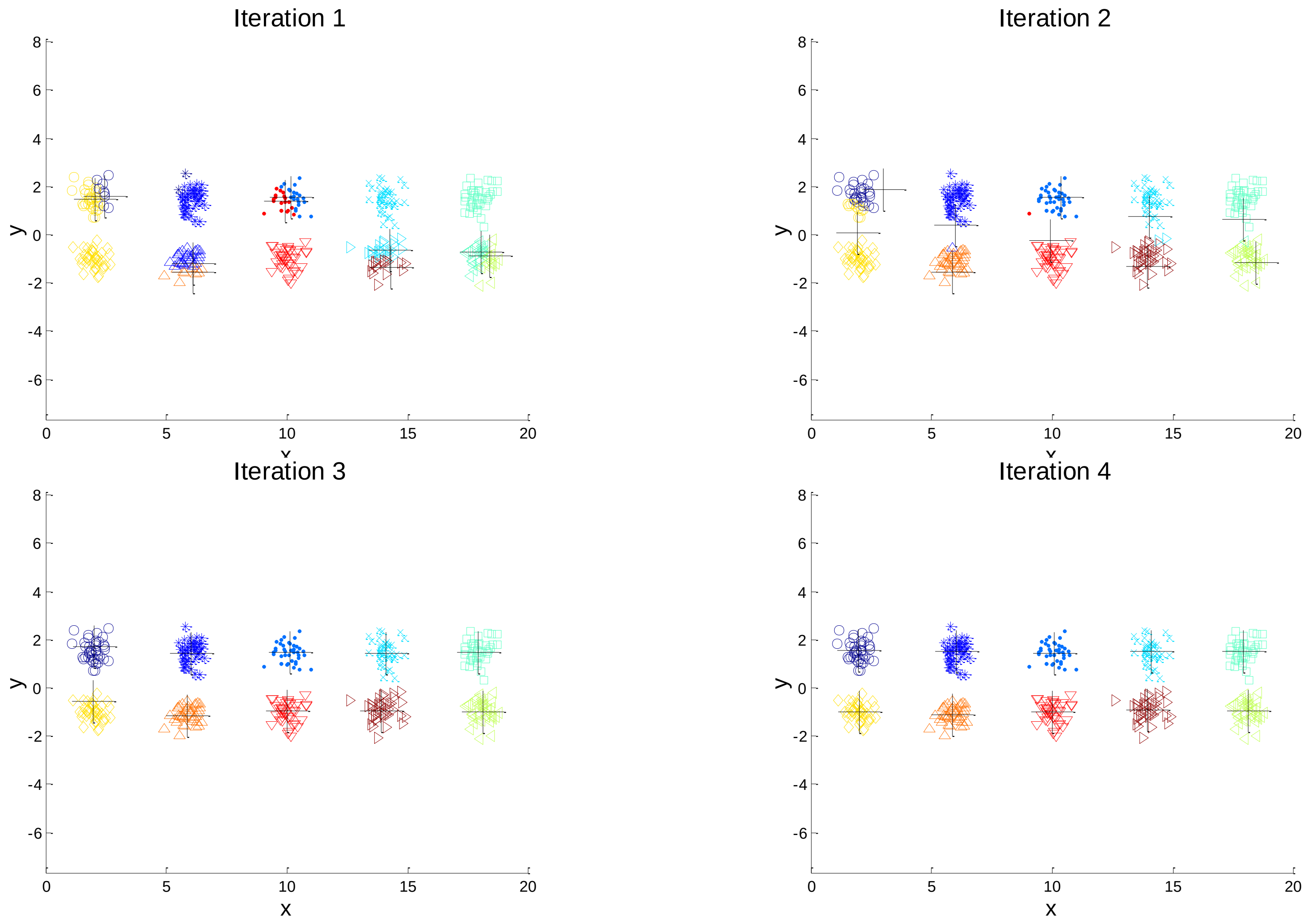
- If there are *k* ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when *k* is large
  - Consider an example of five pairs of clusters

10 Clusters Example



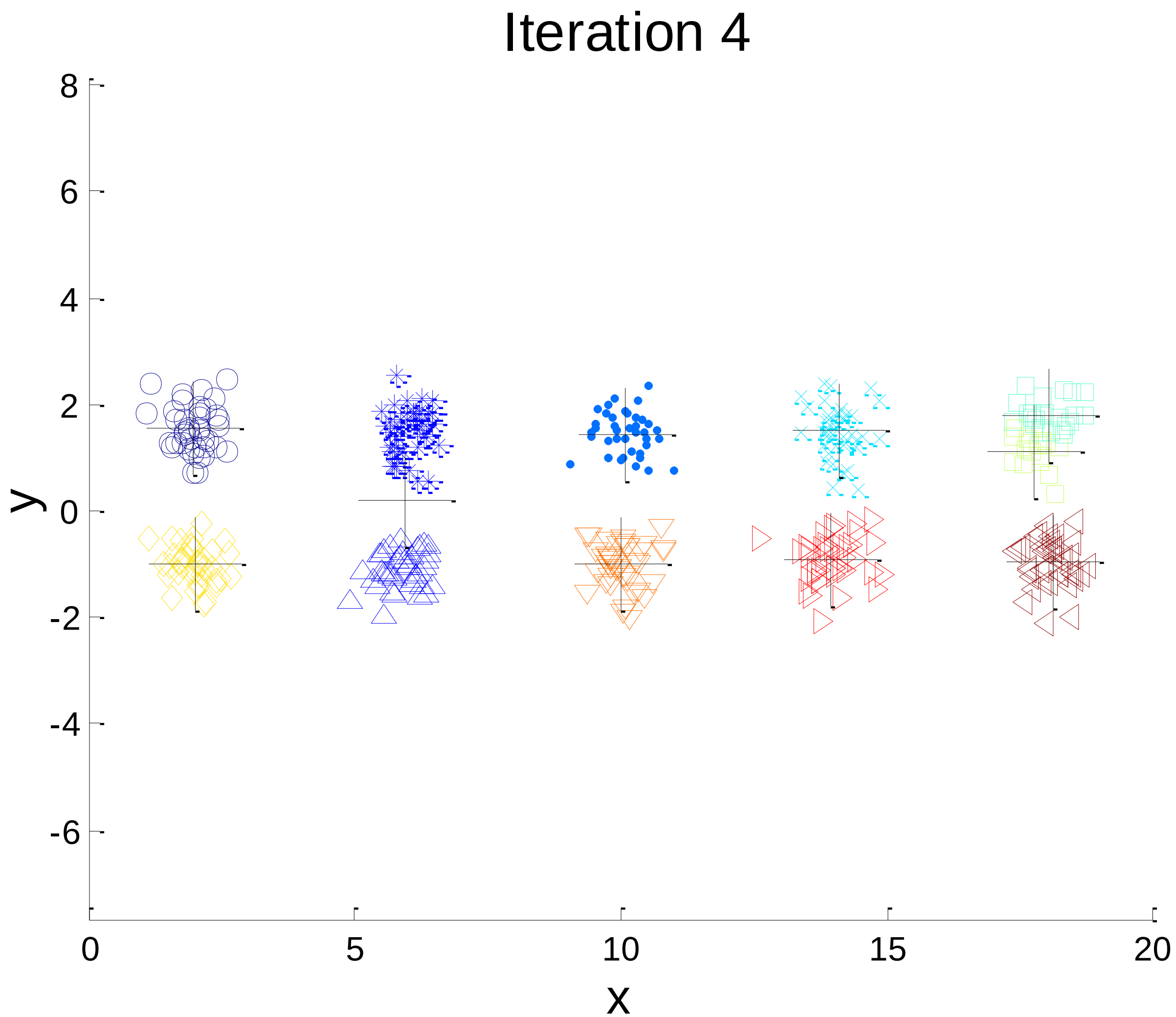
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



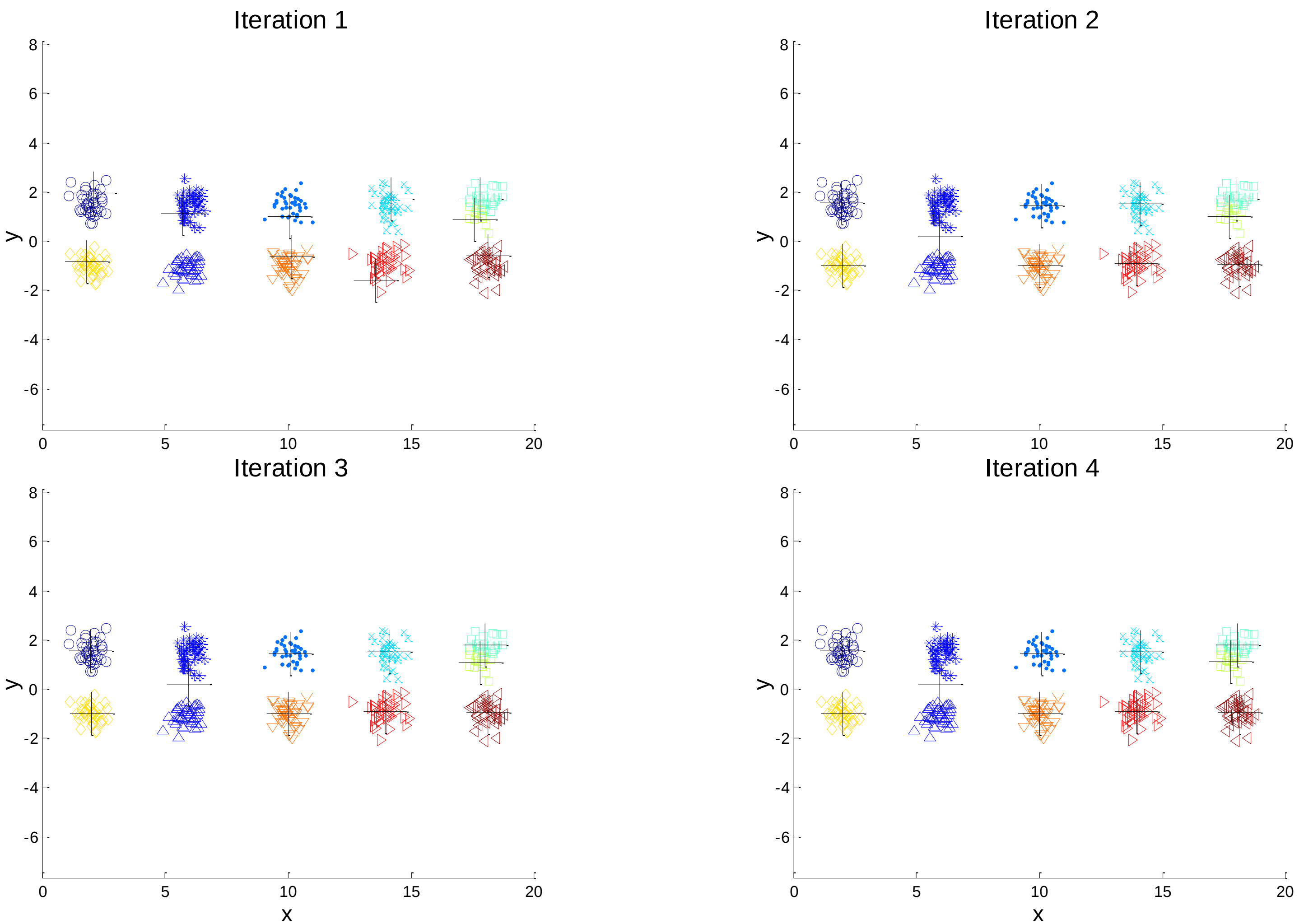
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids

Solutions to Initial Centroids Problem

- Select more than *k* initial centroids and then select among these initial centroids
  - Select most widely separated
    - can select outliers, too
- Post-processing
- Bisecting K-means
  - Not as susceptible to initialization issues

Additional Issues: Handling Empty Clusters

- Basic *k*-means algorithm can yield empty clusters

- Several strategies
  - Choose the point that contributes most to SSE
  - Choose a point from the cluster with the highest SSE
  - If there are several empty clusters, the above can be repeated several times.

Updating Centroids Incrementally

- In the basic *k*-means algorithm, centroids are updated after all points are assigned to a centroid

- An alternative is to update the centroids after each assignment (incremental approach)
  - Each assignment updates zero or two centroids
  - More expensive
  - Introduces an order dependency!
  - Never gets an empty cluster

Pre-processing and Post-processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split 'loose' clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are 'close' and that have relatively low SSE