Clustering algorithms: K-means

Lecture 19

Clustering algorithms

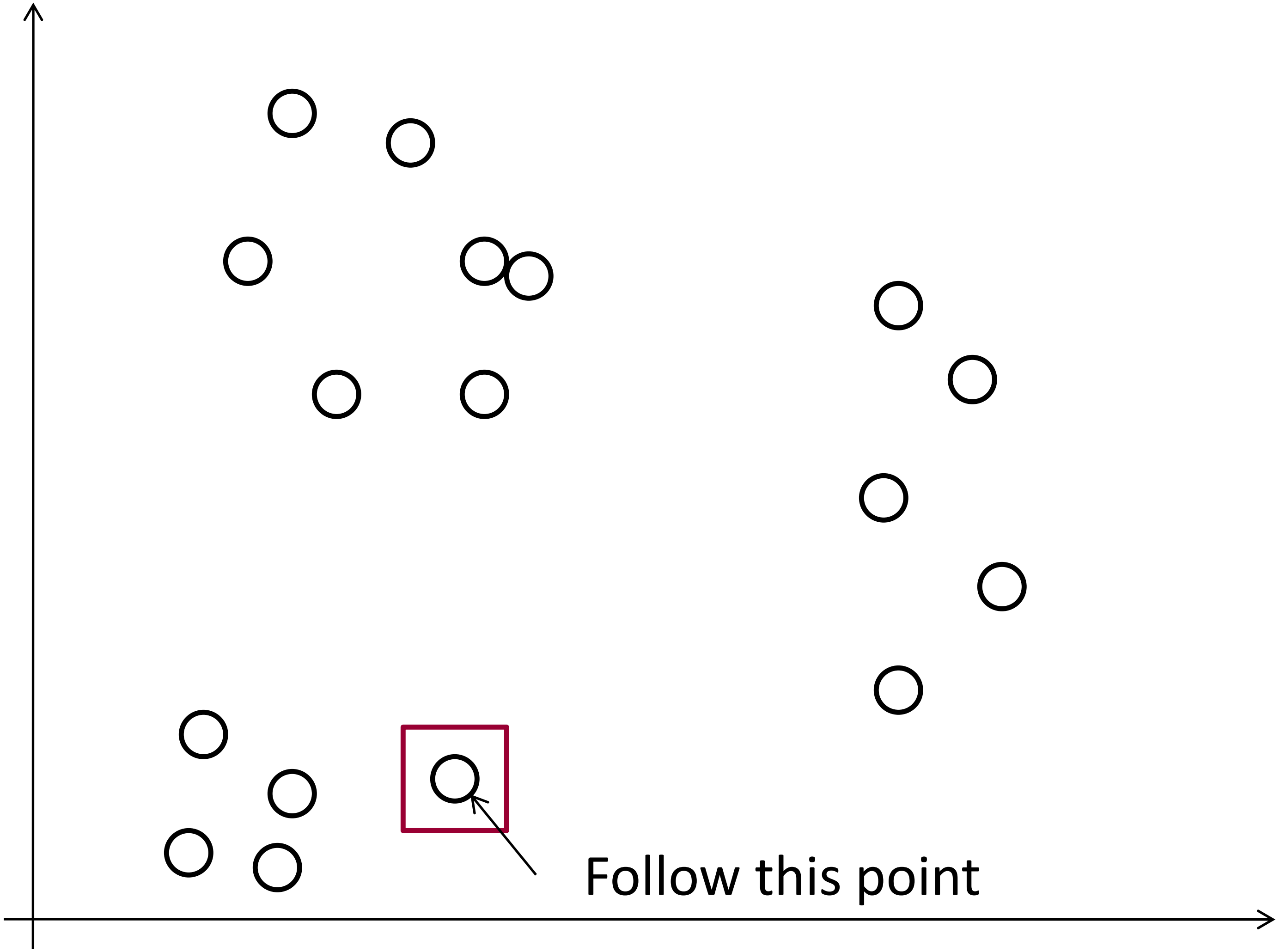
- *K*-means clustering
- Agglomerative hierarchical clustering
- Density-based clustering

Iterative solution: *K*-means clustering algorithm

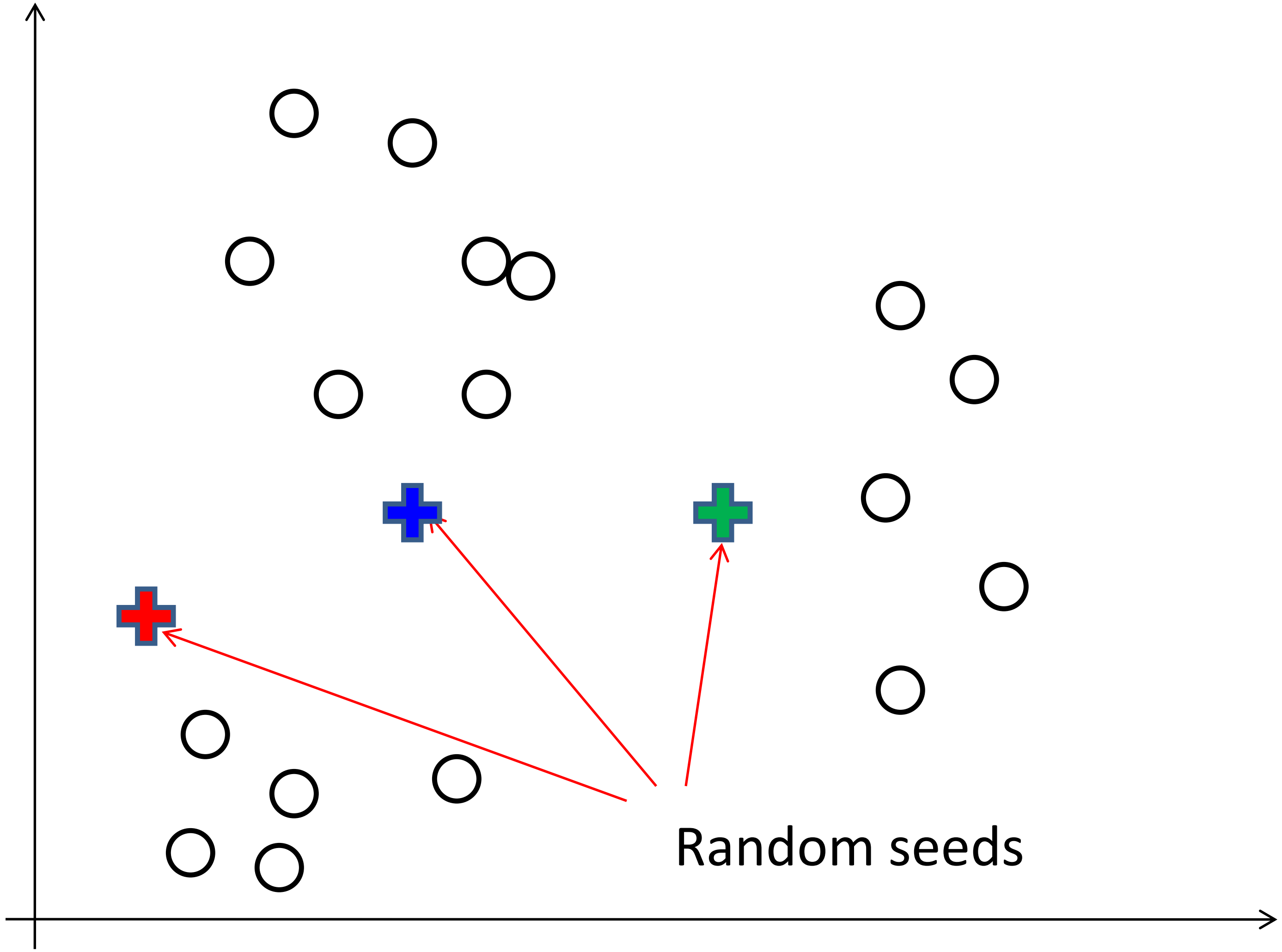
Select *K* random seeds
Do
Assign each record to the closest seed
Calculate centroid of each cluster
(take average value for each dimension
of all records in the cluster)
Set these centroids as new seeds
Until coordinates of seeds *do not change*

This algorithm in each iteration makes assignment of points such that intra-cluster distances are decreasing.
Local optimization technique – moves into the direction of local minimum, might miss the best solution

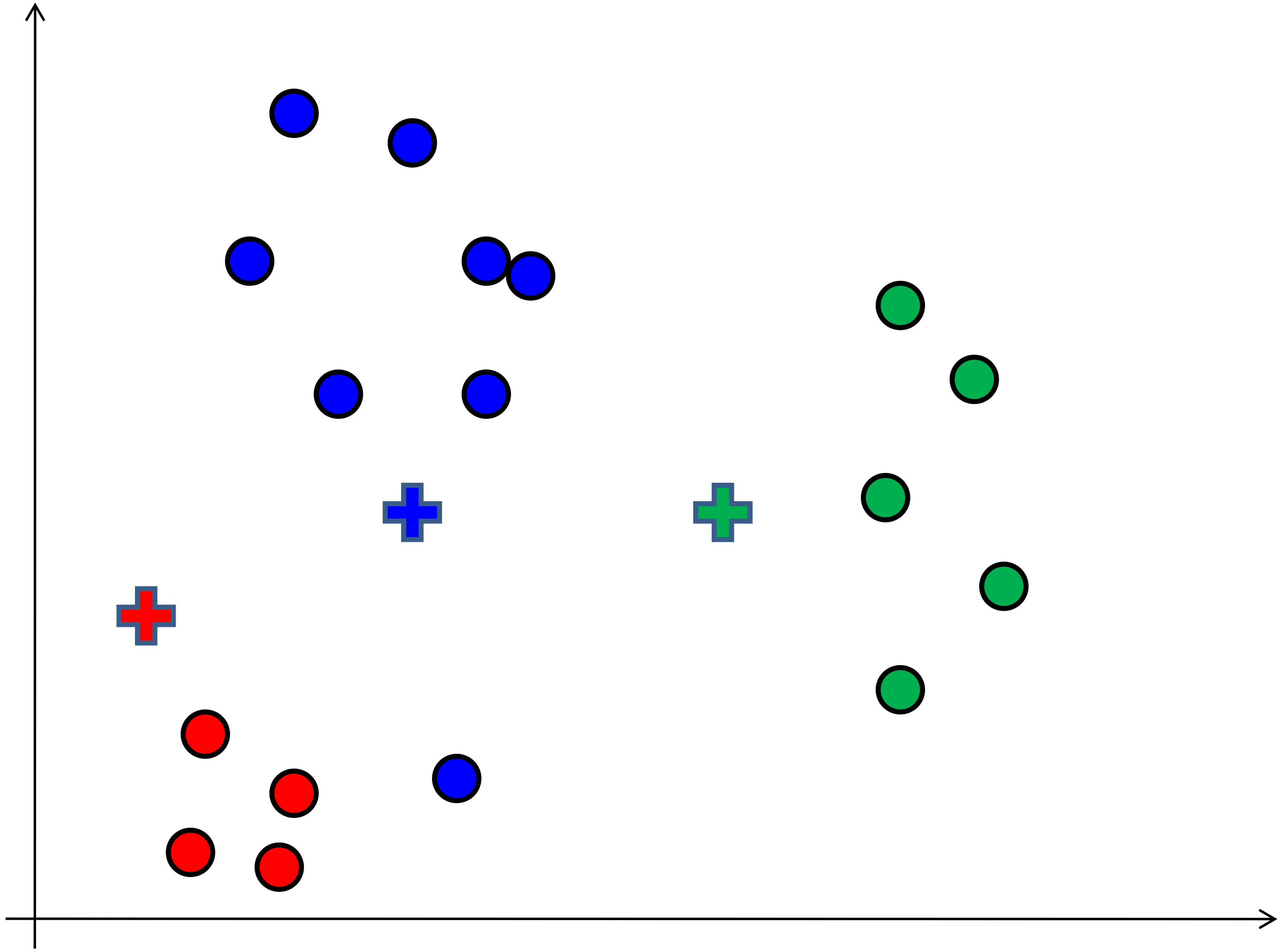
Example 1: K=3



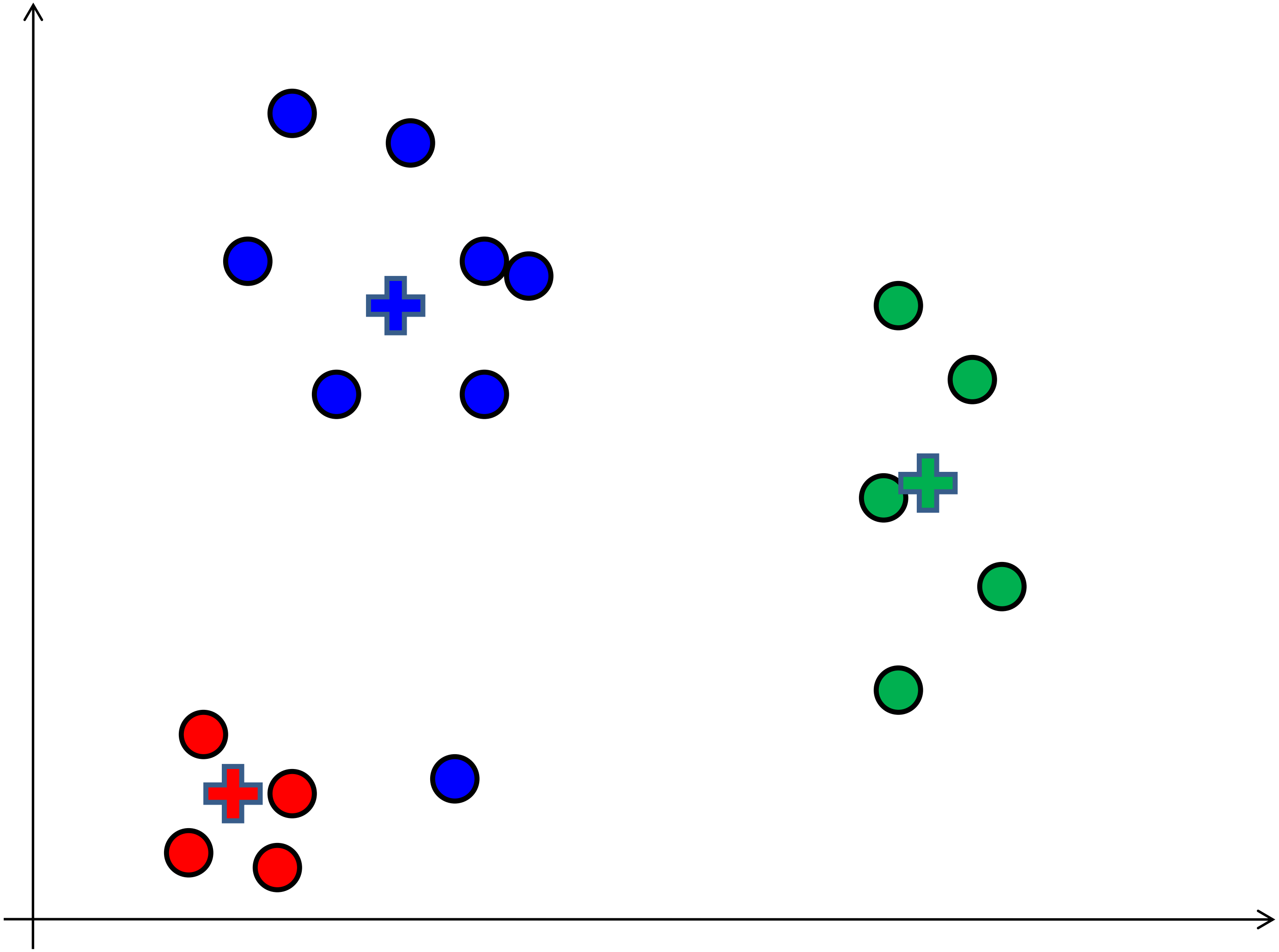
Example 1: initialization



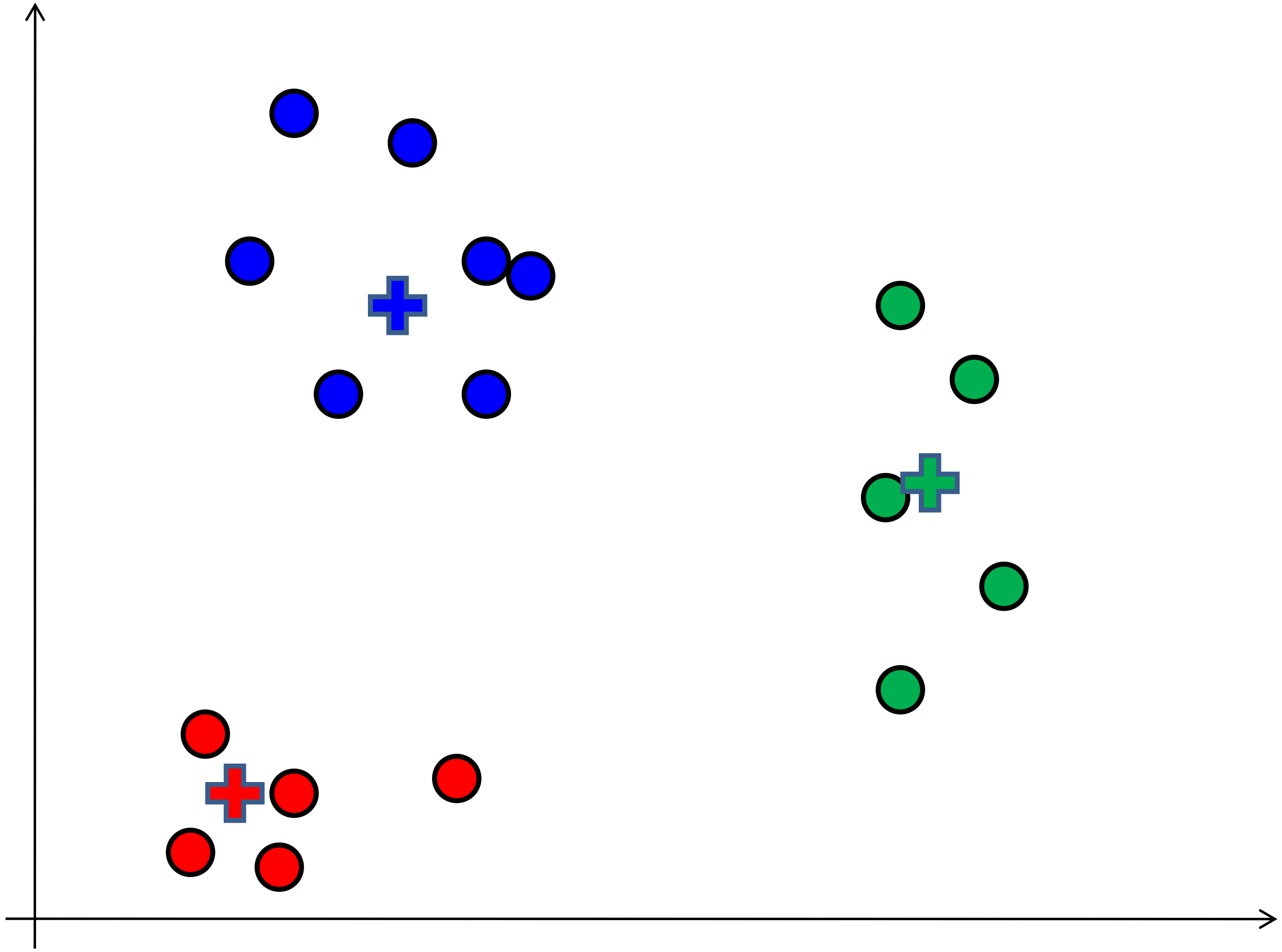
Example 1: iteration 1. Assign each point to the closest seed



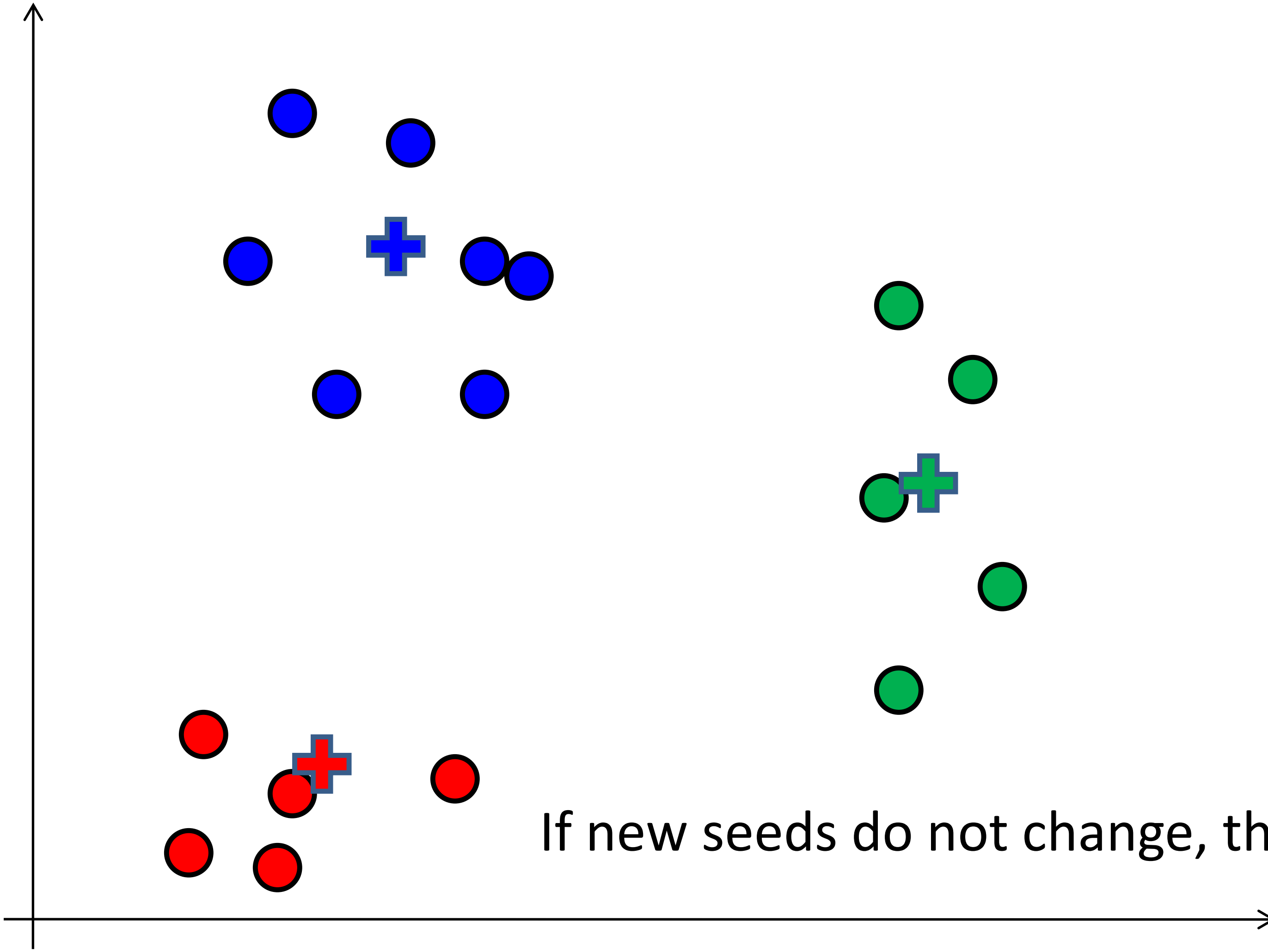
Example 1: iteration 1. Recalculate centroids



Example 1: iteration 2.
Assign each point to the closest seed

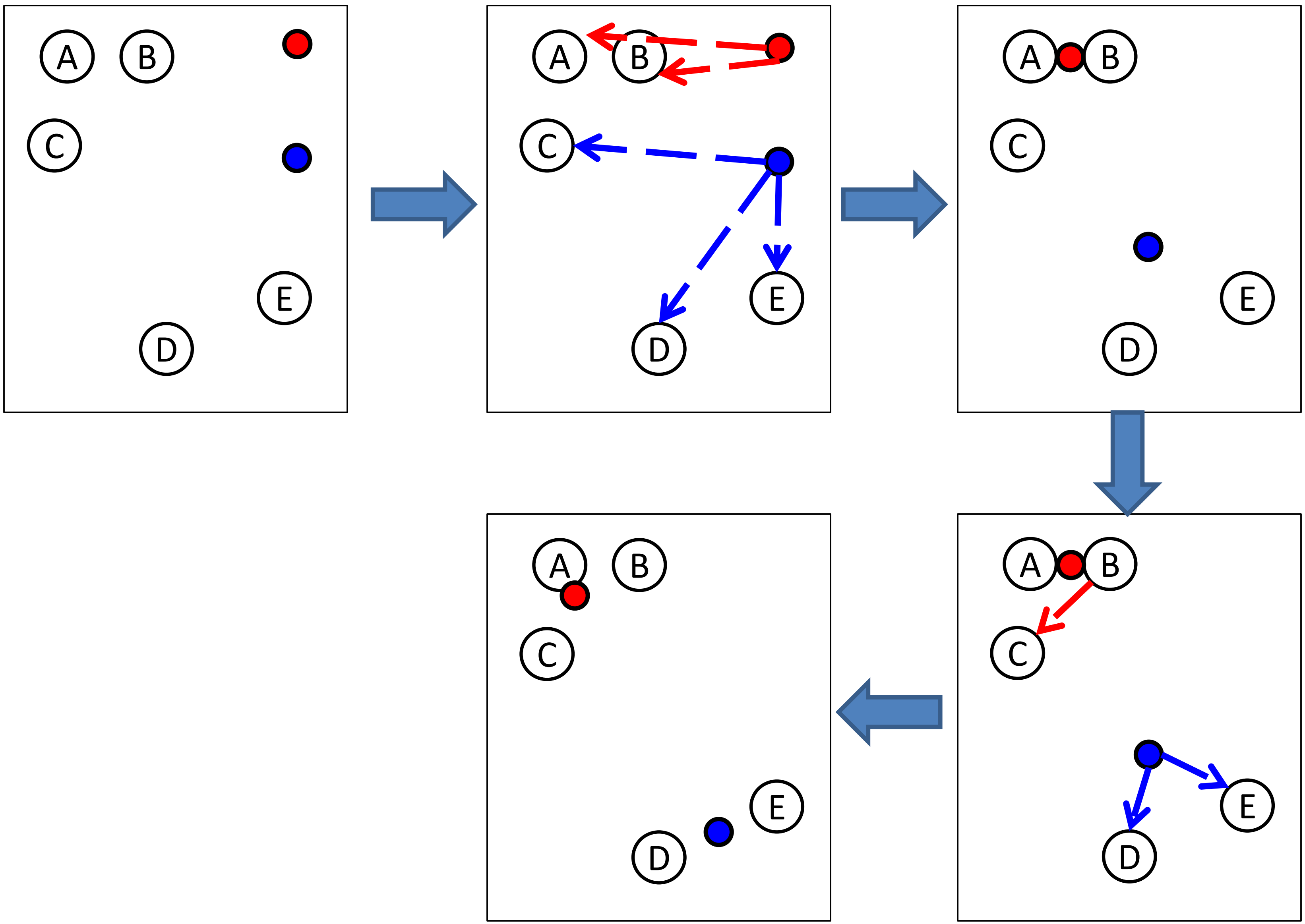


Example 1: iteration 2. recalculate centroids – new seeds



If new seeds do not change, then stop

Example 2: K=2



Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster centroid
  - To get SSE, we square these errors and sum them up.

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

x is a data point in cluster C_i and

m_i is the representative point for cluster C_i (in our case, centroid)

K-means Clustering – Details

- Centroid that minimizes SSE of each cluster is a mean
  - (can be shown mathematically – see page 513 of the textbook)

- At each iteration, we decrease total SSE, but with respect to a given set of centroids and point assignments

K-means Clustering – Details

- Initial centroids may be chosen randomly.
  - Clusters produced vary from one run to another.

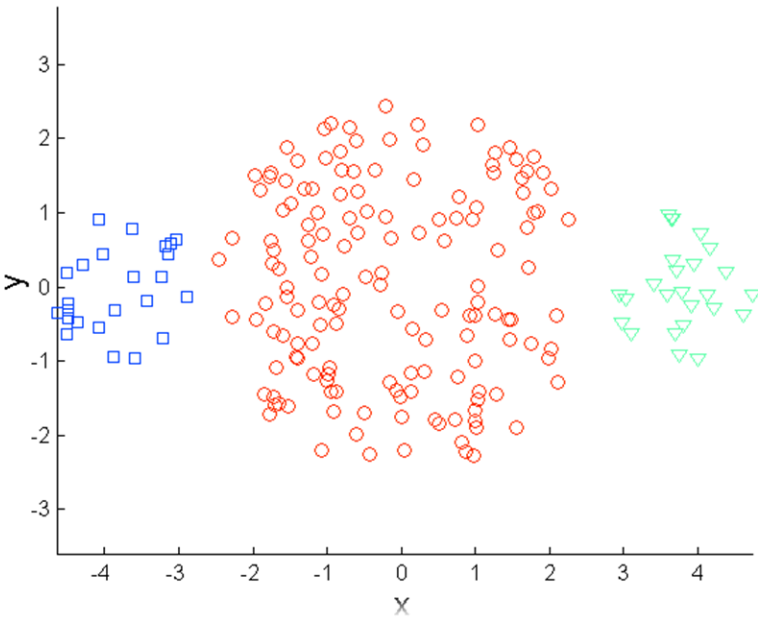
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'

- Complexity is $O(I * K * n * d)$
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

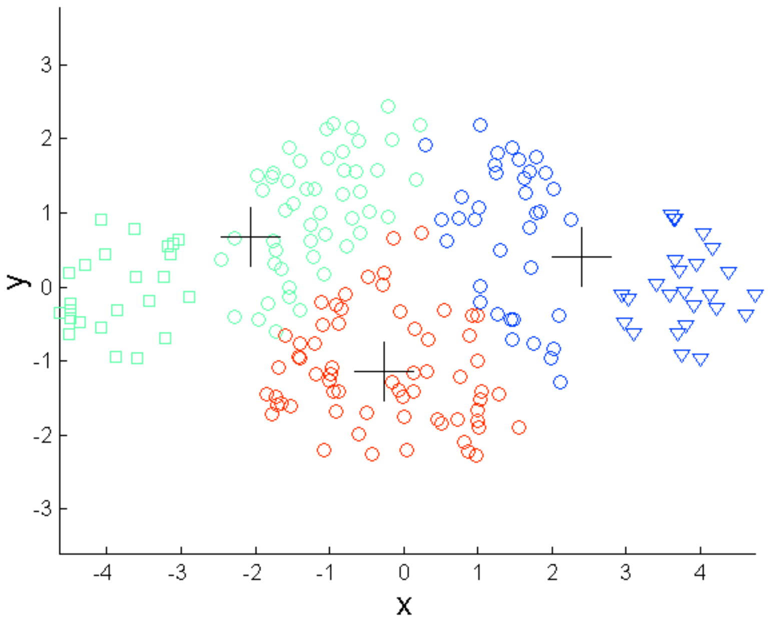
Limitations of K-means

- K-means has problems when clusters are of
  - Differing Sizes
  - Differing Densities
  - Non-globular shapes

Limitations of K-means: Differing Sizes

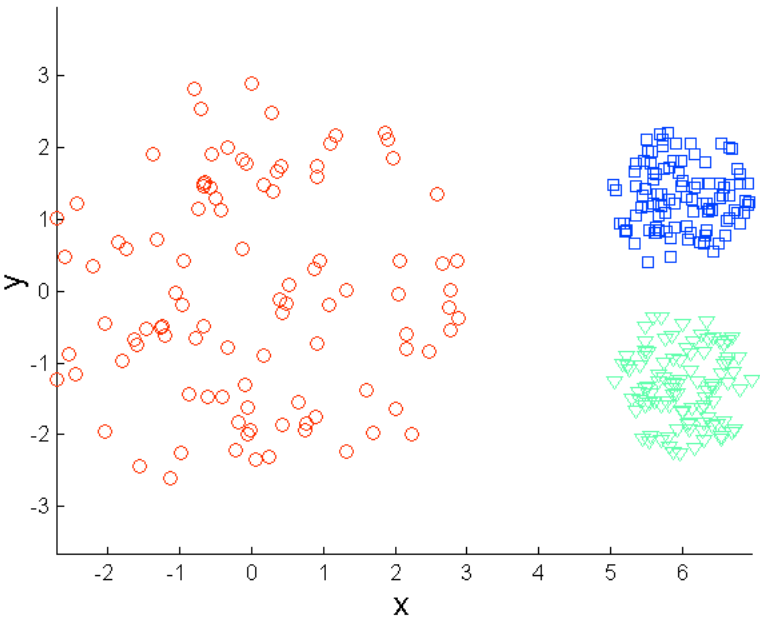


Original Points

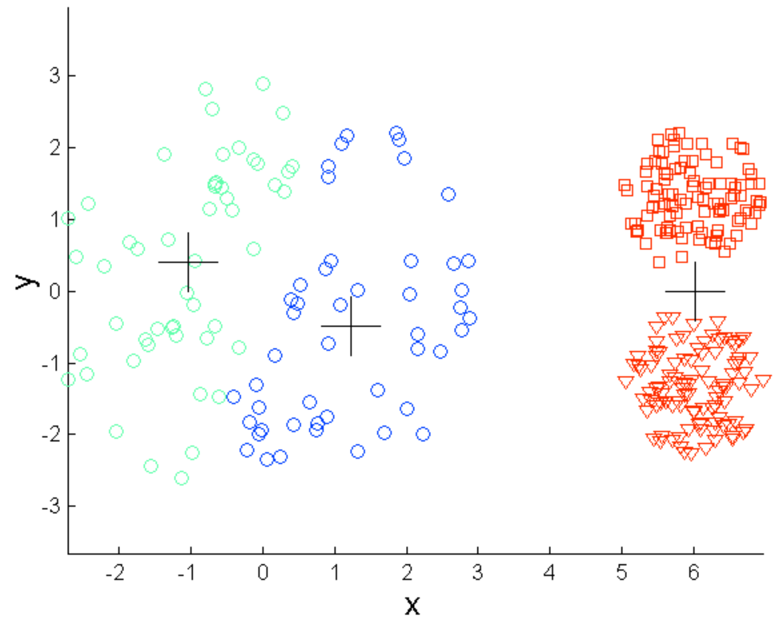


K-means (3 Clusters)

Limitations of K-means: Differing Density

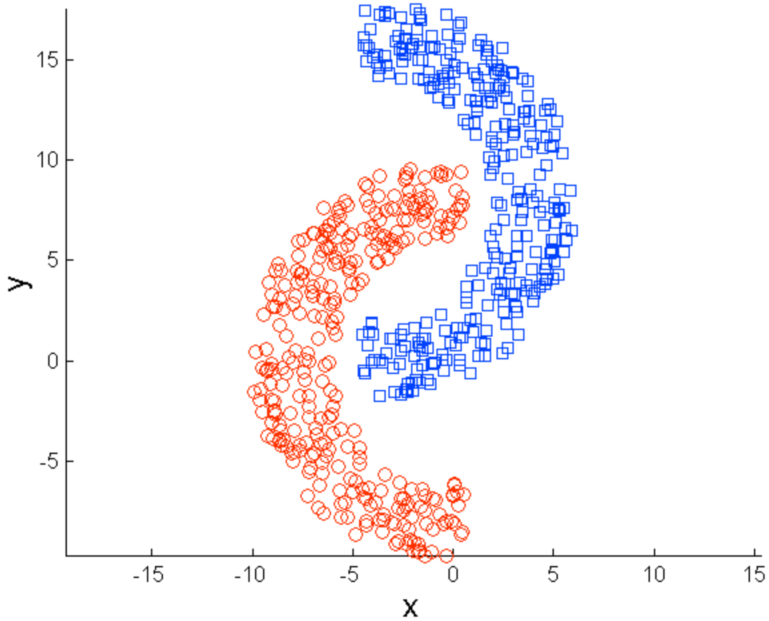


Original Points

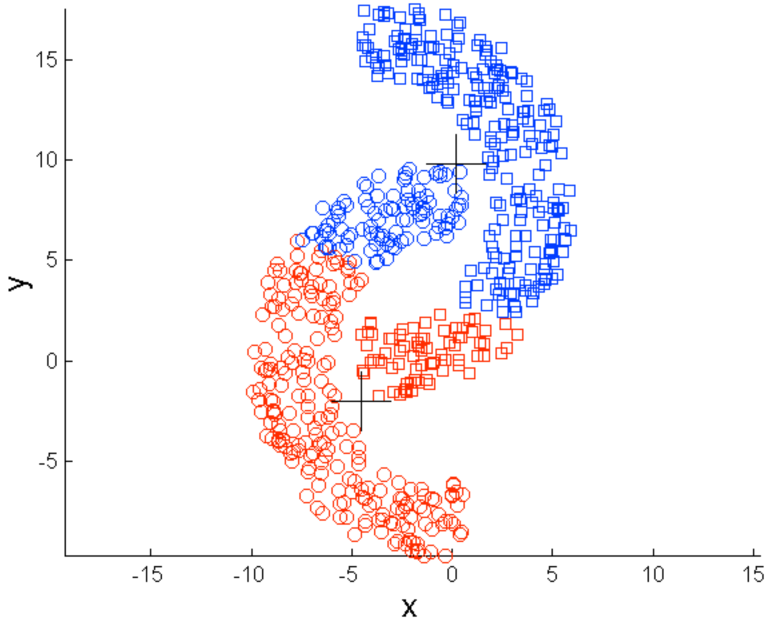


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



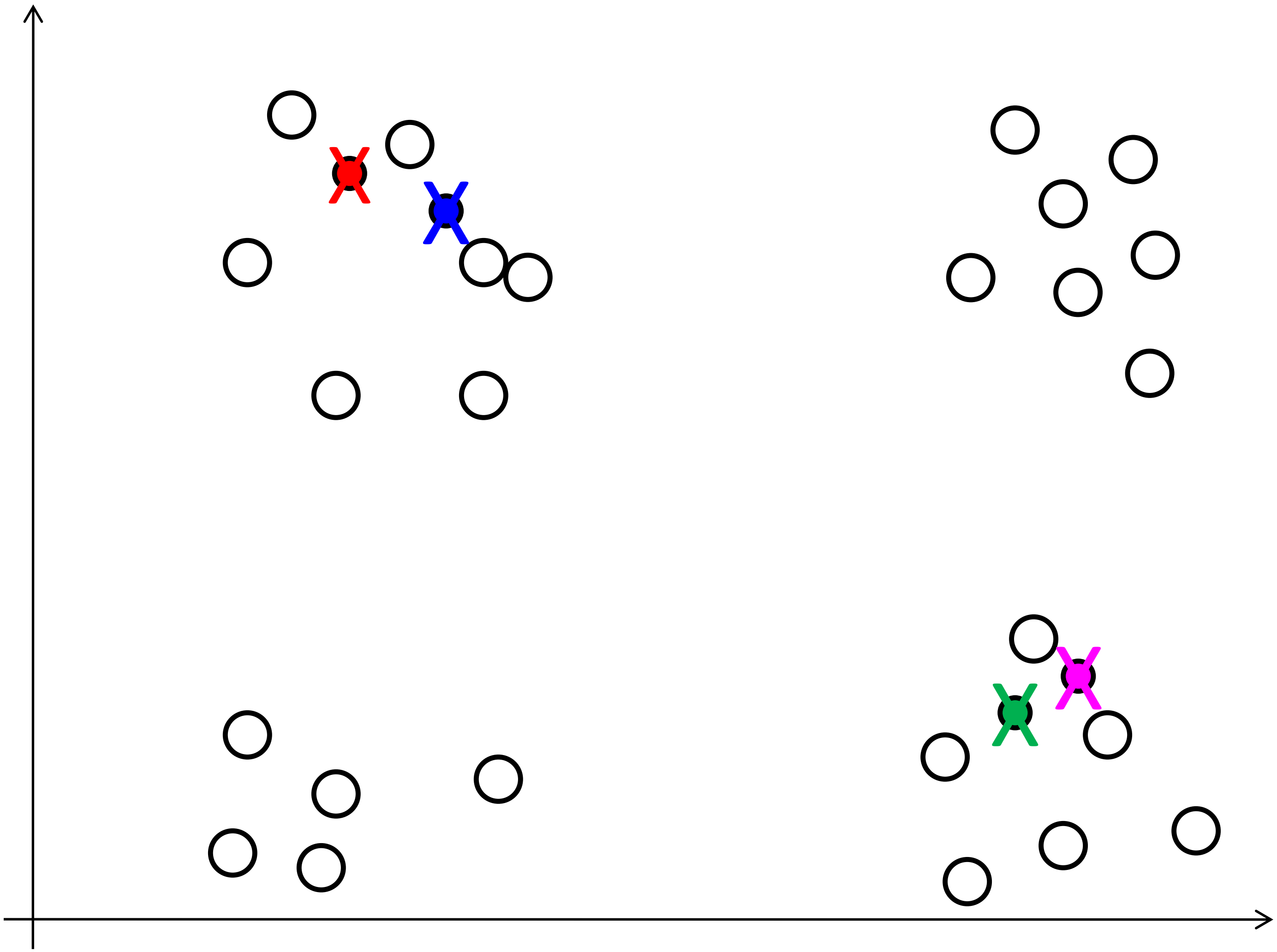
K-means (2 Clusters)

Limitations of K-means

- K-means has problems when clusters are of
  - Differing Sizes
  - Differing Densities
  - Non-globular shapes

- But even for globular clusters, the choice of initial centroids influences the quality of clustering

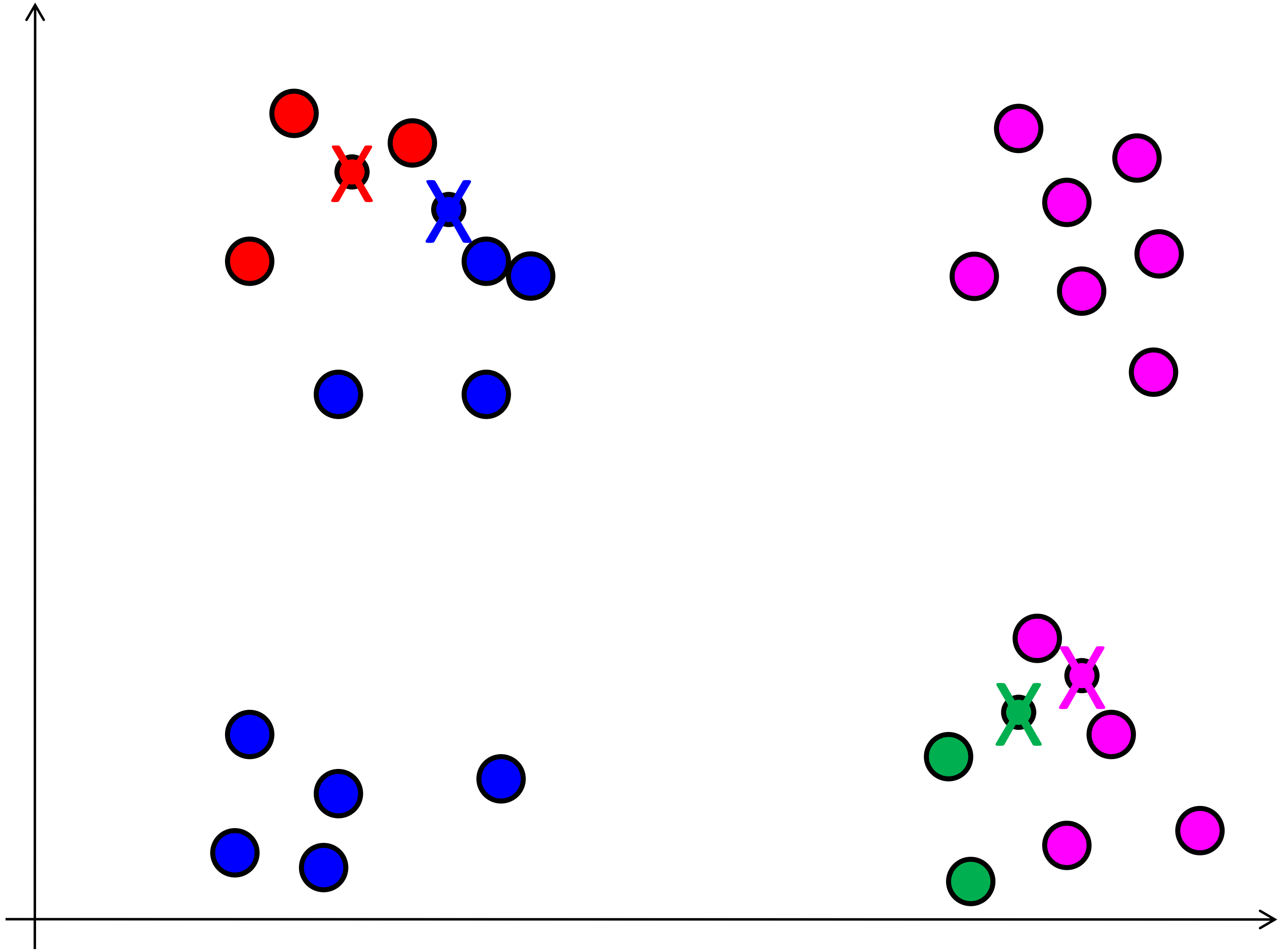
1. Importance of choosing initial centroids: *K*=4



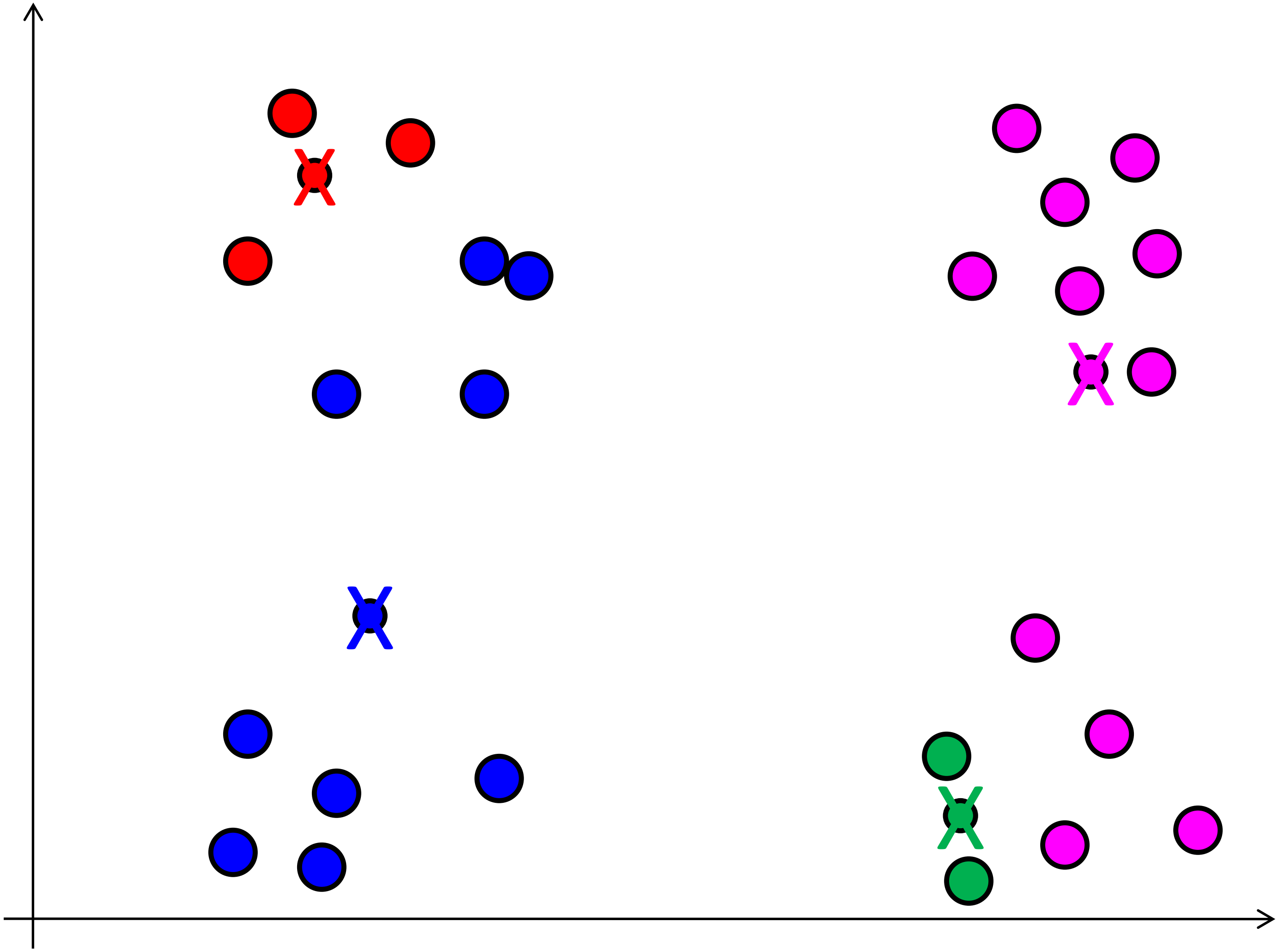
2 pairs of clusters

Initial seeds are chosen: 2 seeds per each pair

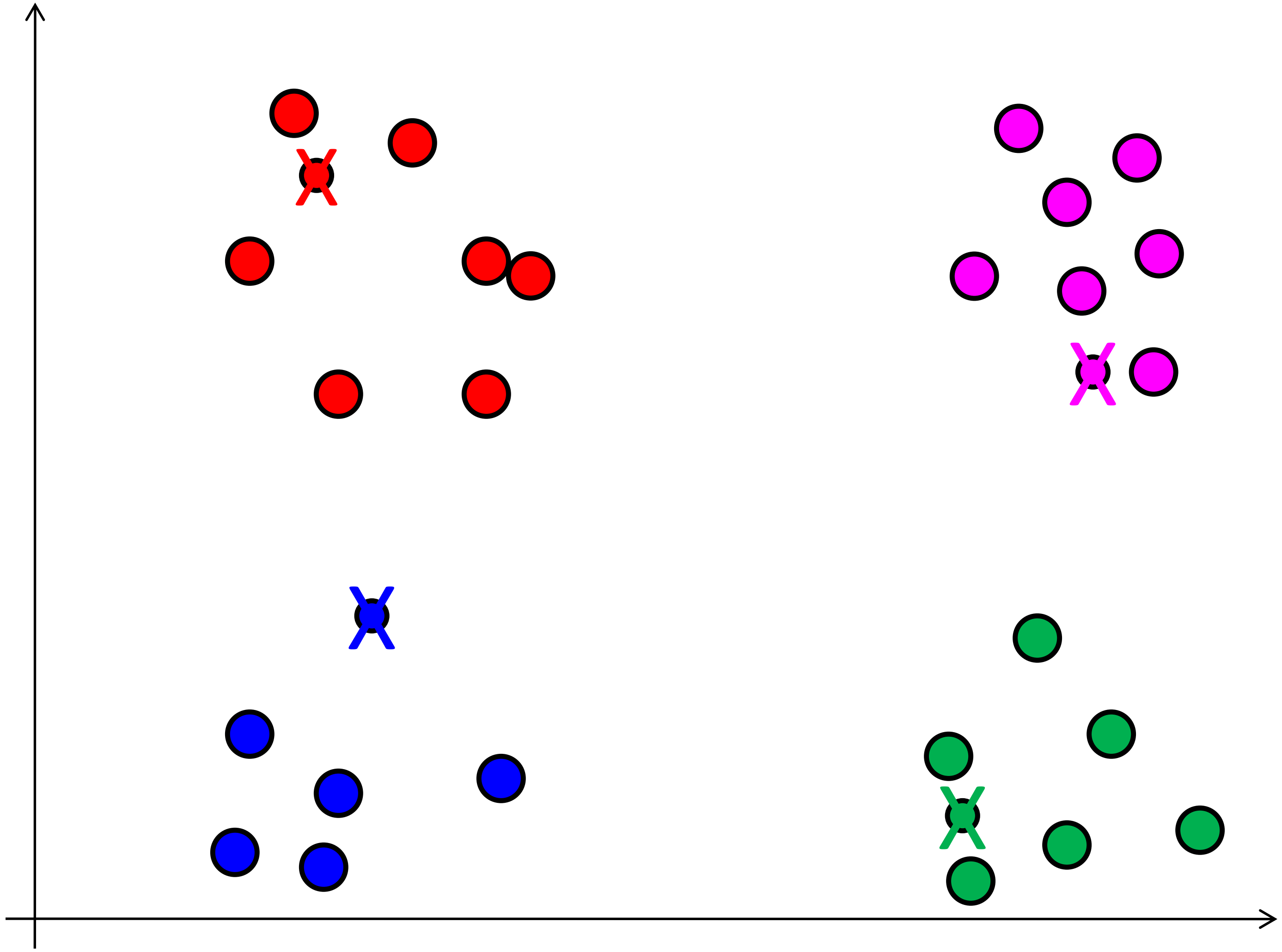
1. Importance of choosing initial centroids: point assignments



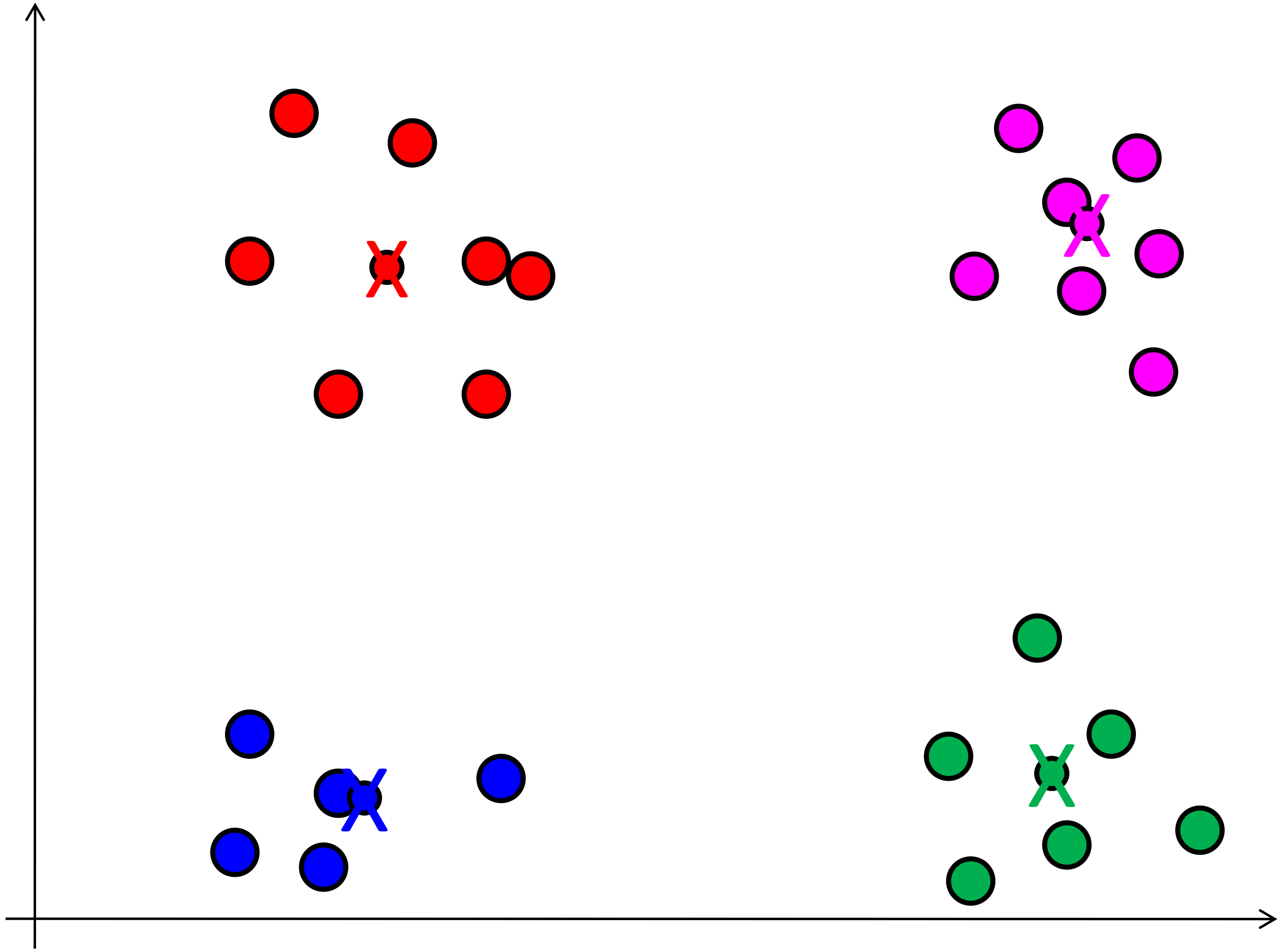
1. Importance of choosing initial centroids: recalculate centroids



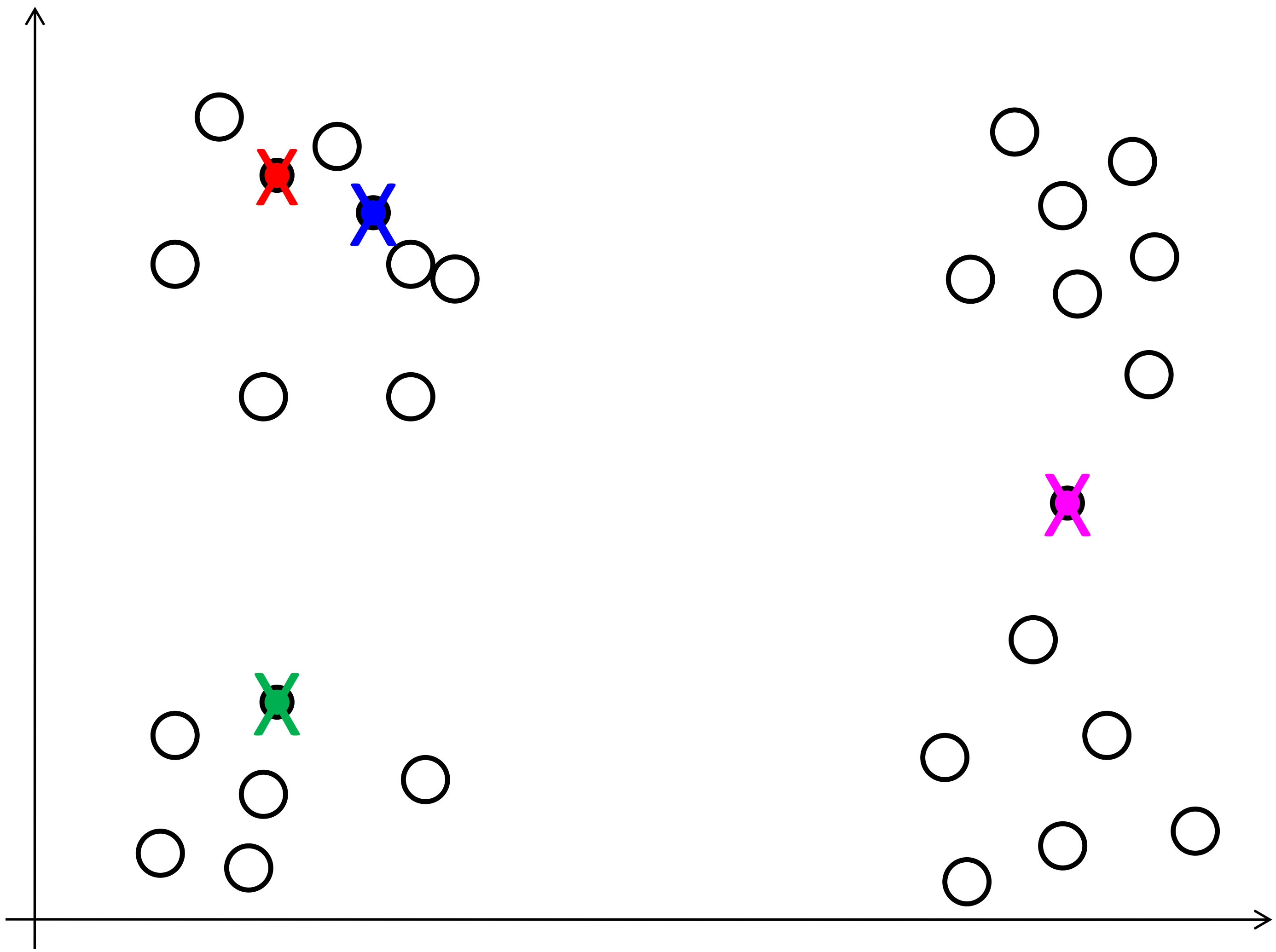
1. Importance of choosing initial centroids: points re-assignments



1. Importance of choosing initial centroids: success – correct clusters



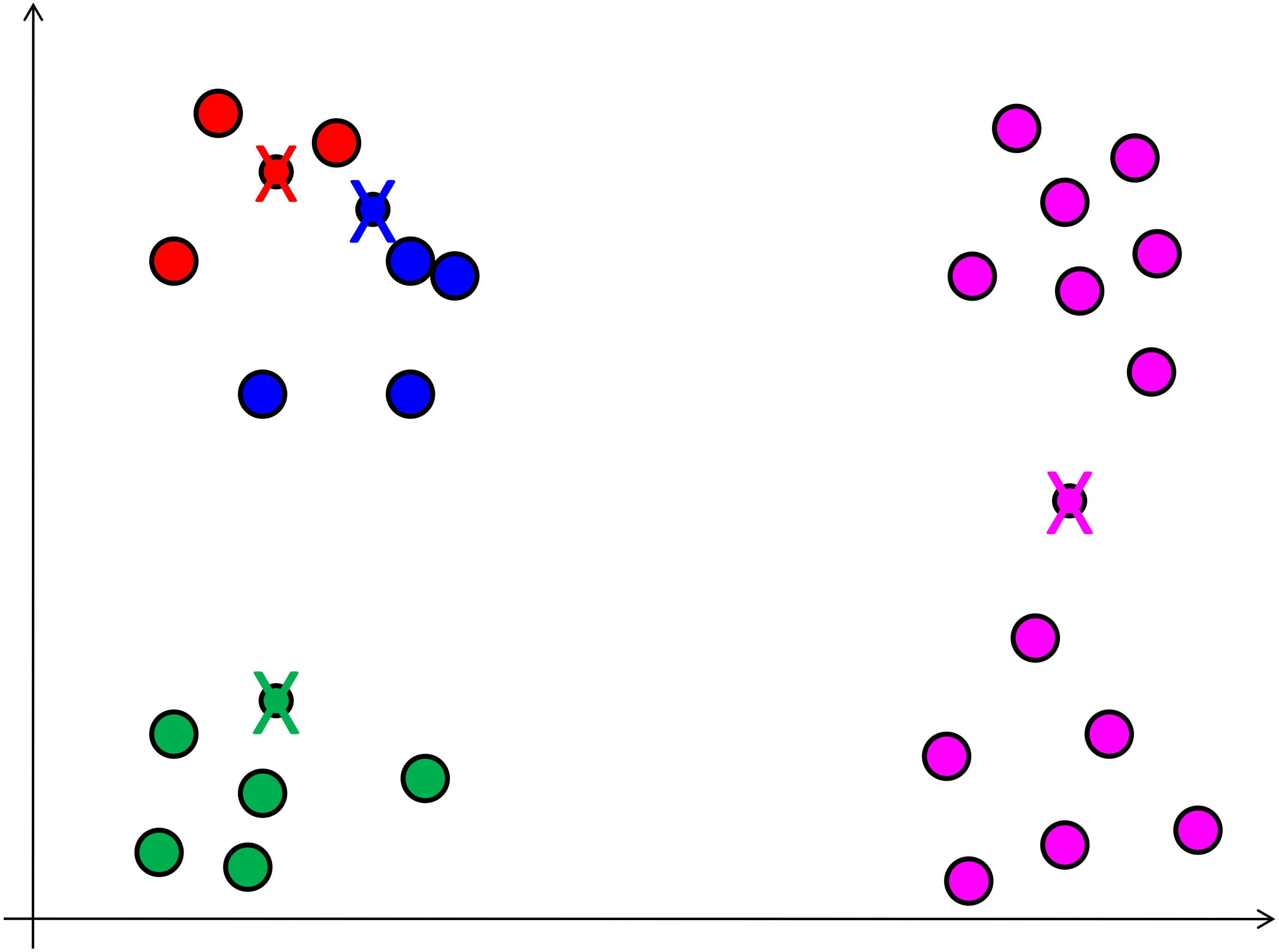
2. Importance of choosing initial centroids: K=4



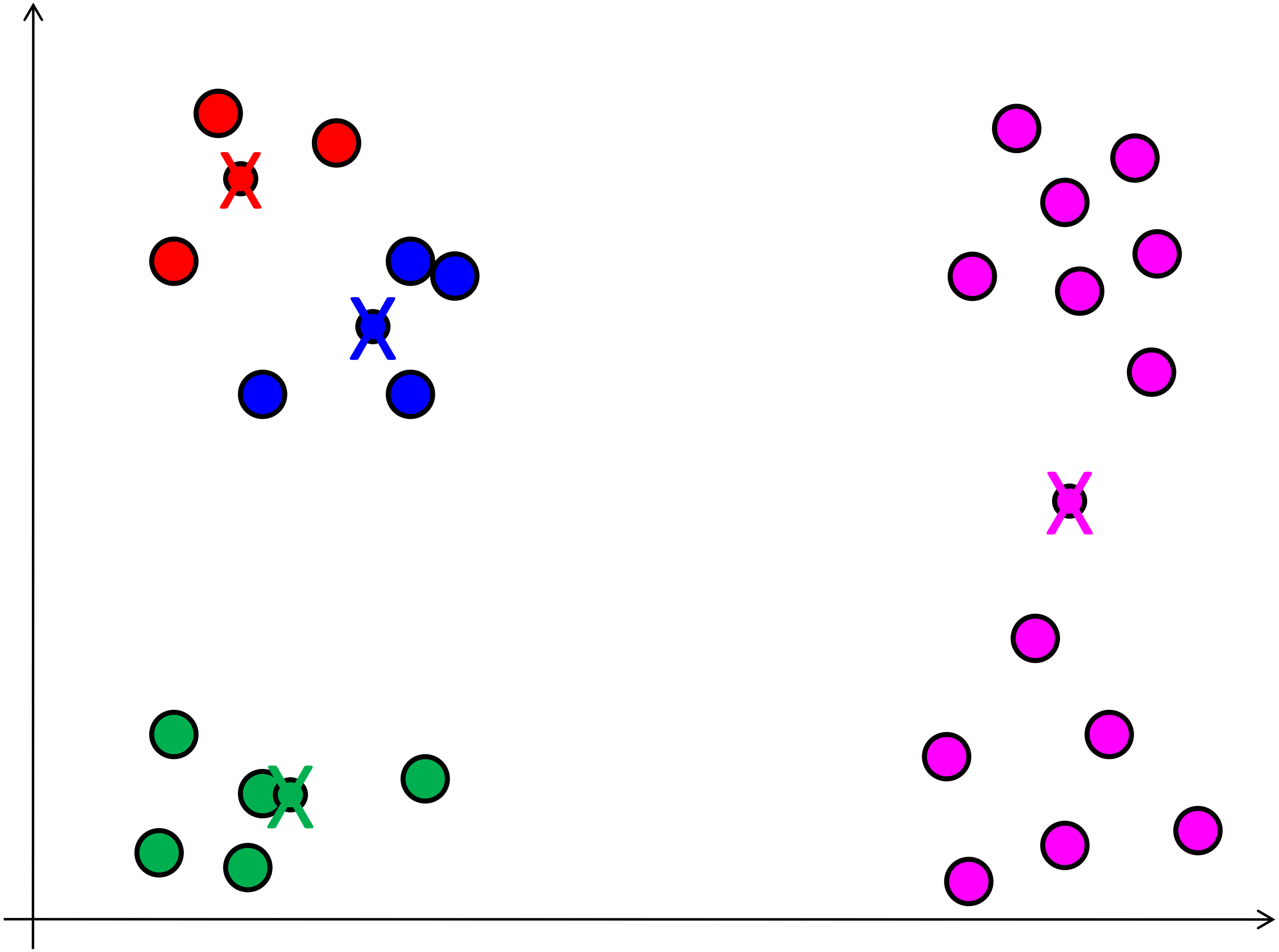
2 pairs of clusters

Initial seeds are chosen: 3 seeds in one pair

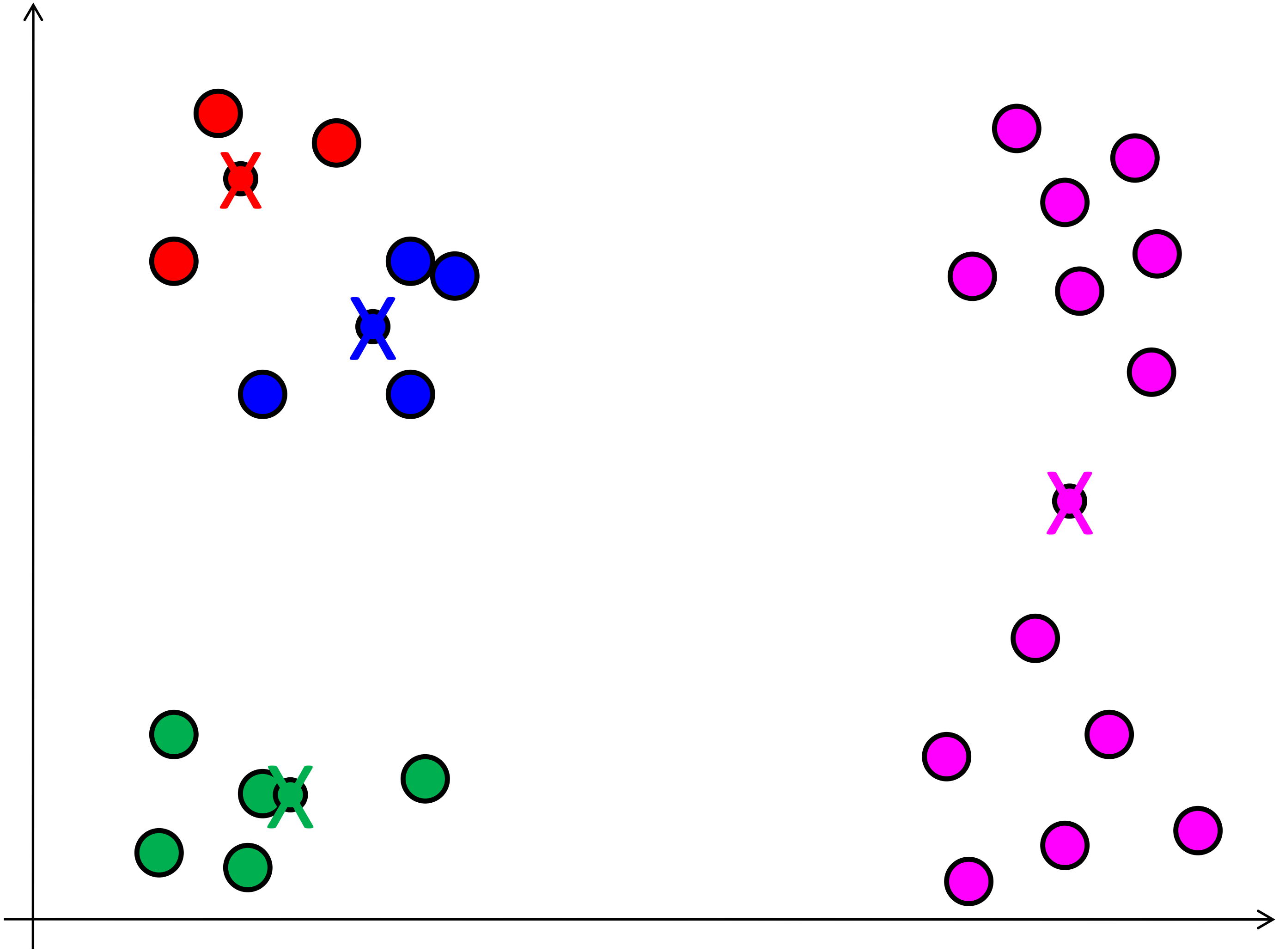
2. Importance of choosing initial centroids: assign points



2. Importance of choosing initial centroids: re-compute centroids



2. Importance of choosing initial centroids: found 4 clusters - incorrect



Problems with Selecting Initial Centroids

- Of course, the ideal would be to choose initial centroids, one from each true cluster.
- 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
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

- For example, if $K = 10$, then *probability* = $10!/10^{10} = 0.00036$
- Sometimes the initial centroids readjust themselves in the 'right' way, and sometimes they don't.

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Bisecting K-means
  - Not as susceptible to initialization issues

Bisecting *K*means

- Straightforward extension of the basic *K*means algorithm. Simple idea:

  To obtain K clusters, split the set of points into two clusters, select one of these clusters to split, and so on, until *K* clusters have been produced.

Bisecting Kmeans

Initialize the list of clusters with the cluster consisting of all points.

Do

Remove a cluster from the list of clusters.

//Perform several "trial" bisections of the chosen cluster.

for *i* = 1 **to** number of trials **do**

Bisect the selected cluster using basic *K*-means (i.e. 2-means).

end for

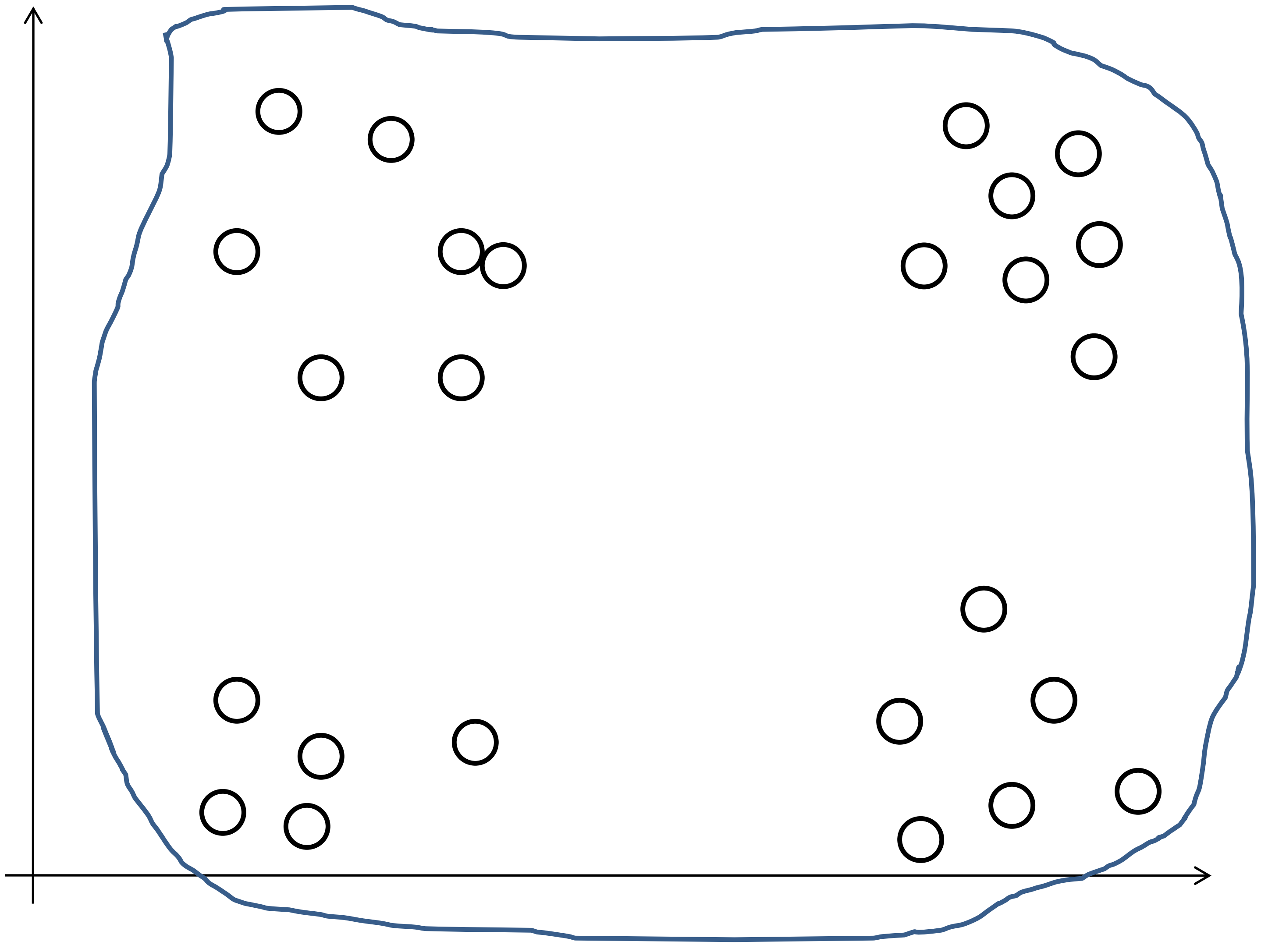
Select the two clusters from the bisection

with the lowest intra-cluster distances (SSE)

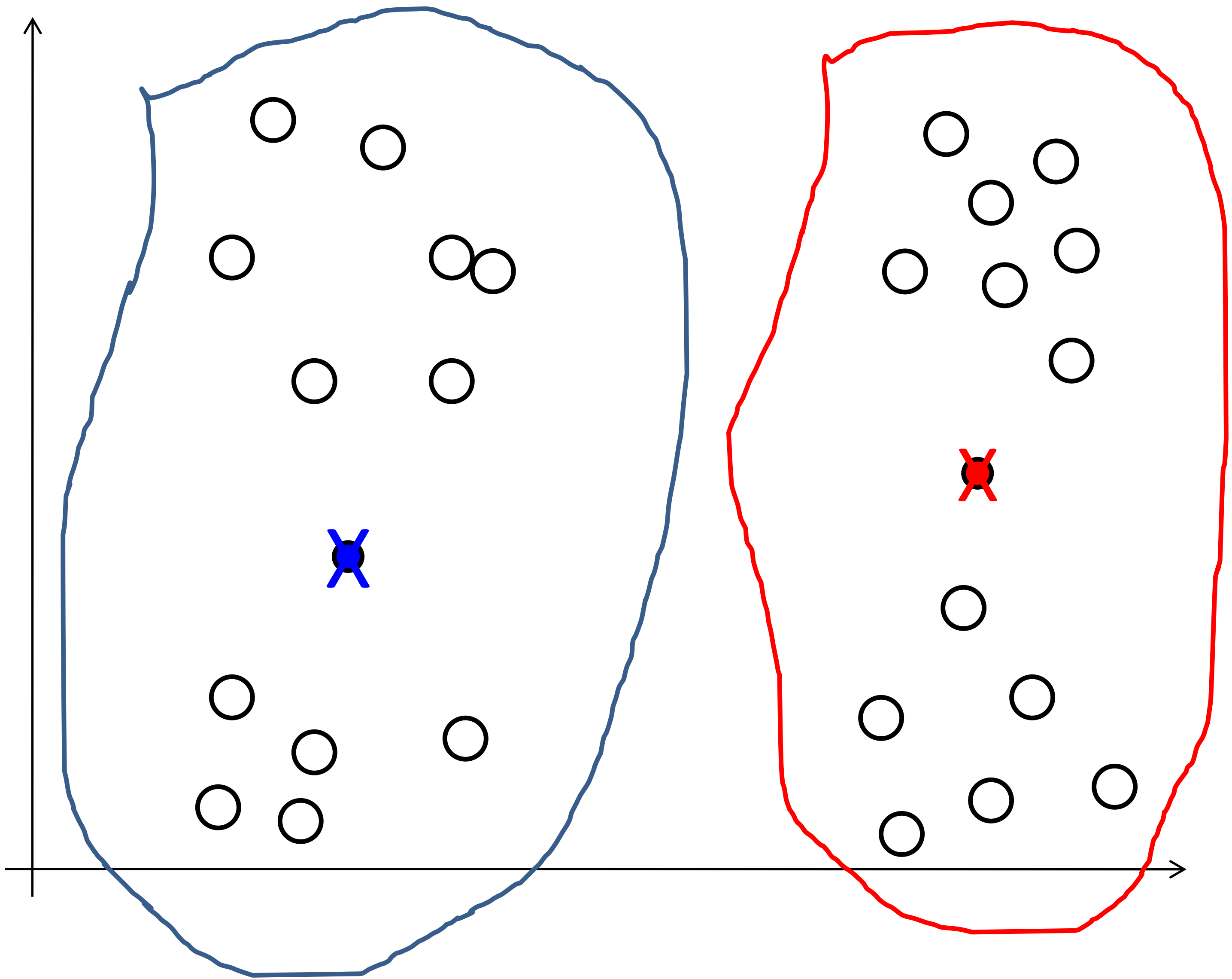
Add these two clusters to the list of clusters.

Until the list of clusters contains *K* clusters.

Bisecting K-means example: one initial cluster



Bisecting K-means example: bisecting initial cluster

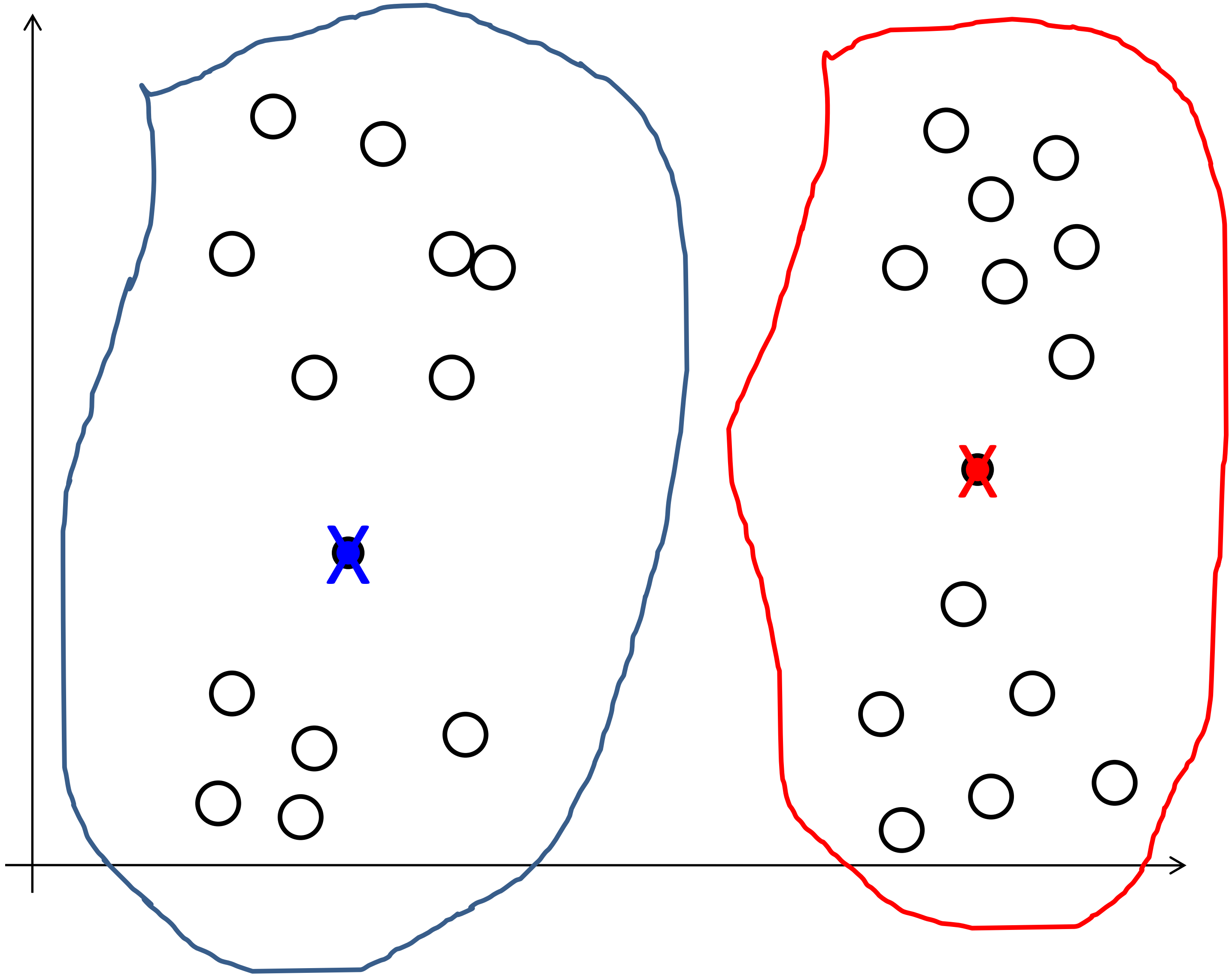


Perform K-means algorithm for K=2

Discovered 2 clusters

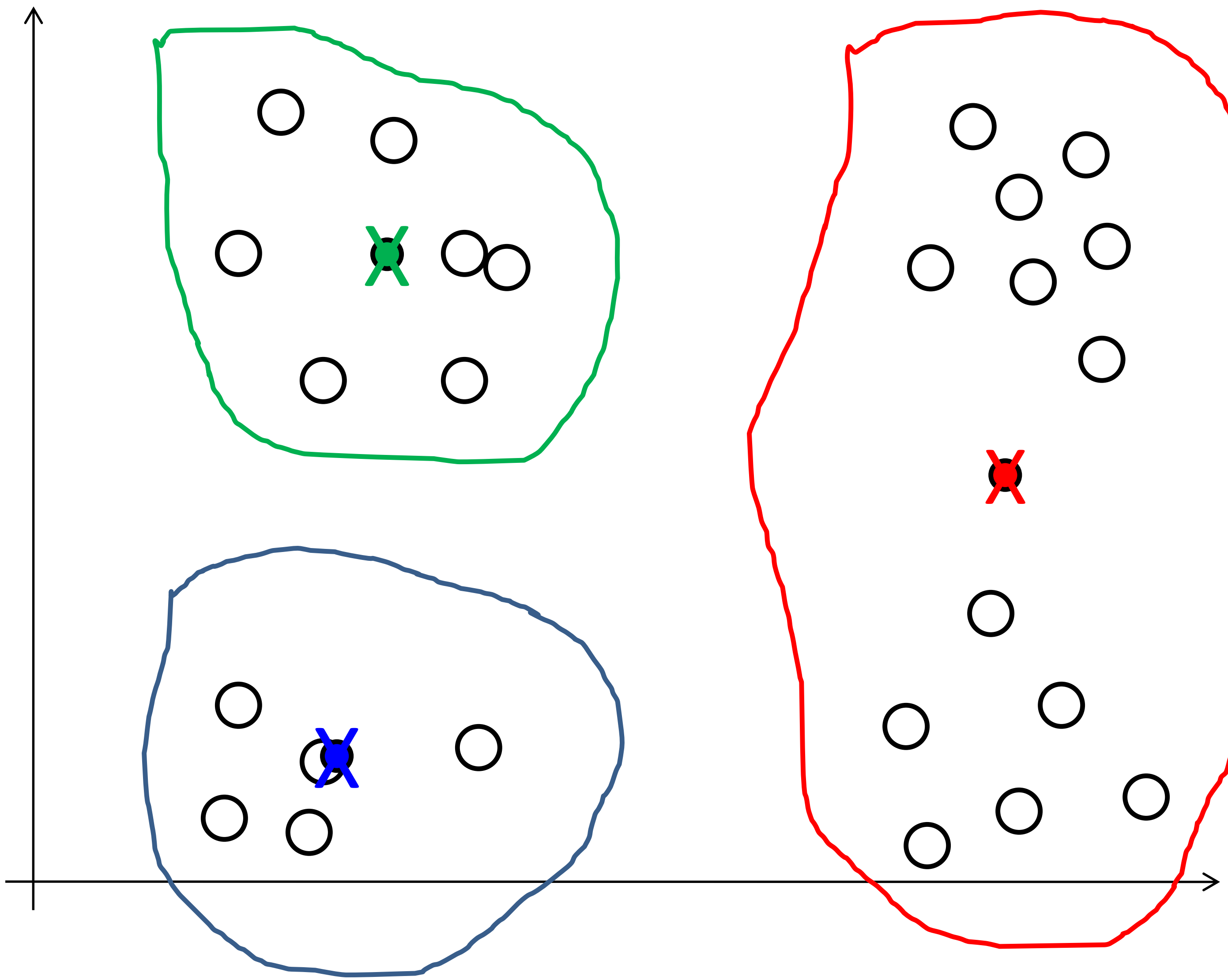
Blue cluster has larger SSE

Bisecting K-means example: bisecting blue cluster



Perform K-means algorithm for K=2 on a blue cluster

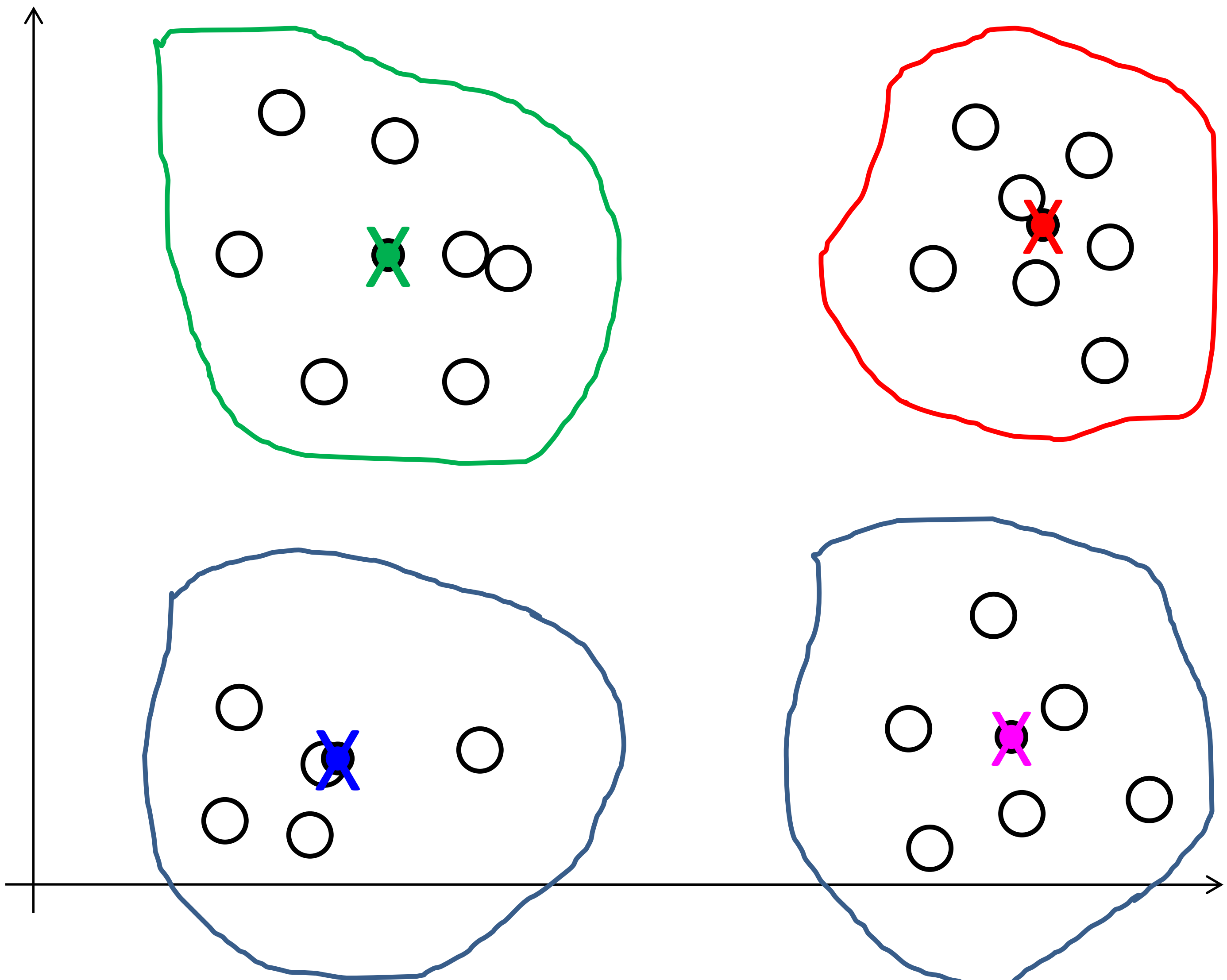
Bisecting K-means example: 3 clusters



Found 2 clusters.

Now red cluster has the largest SSE.

Bisecting K-means example: bisecting red cluster



Process red cluster.

Found 4 clusters.

Stop.

Bisecting K-means Example