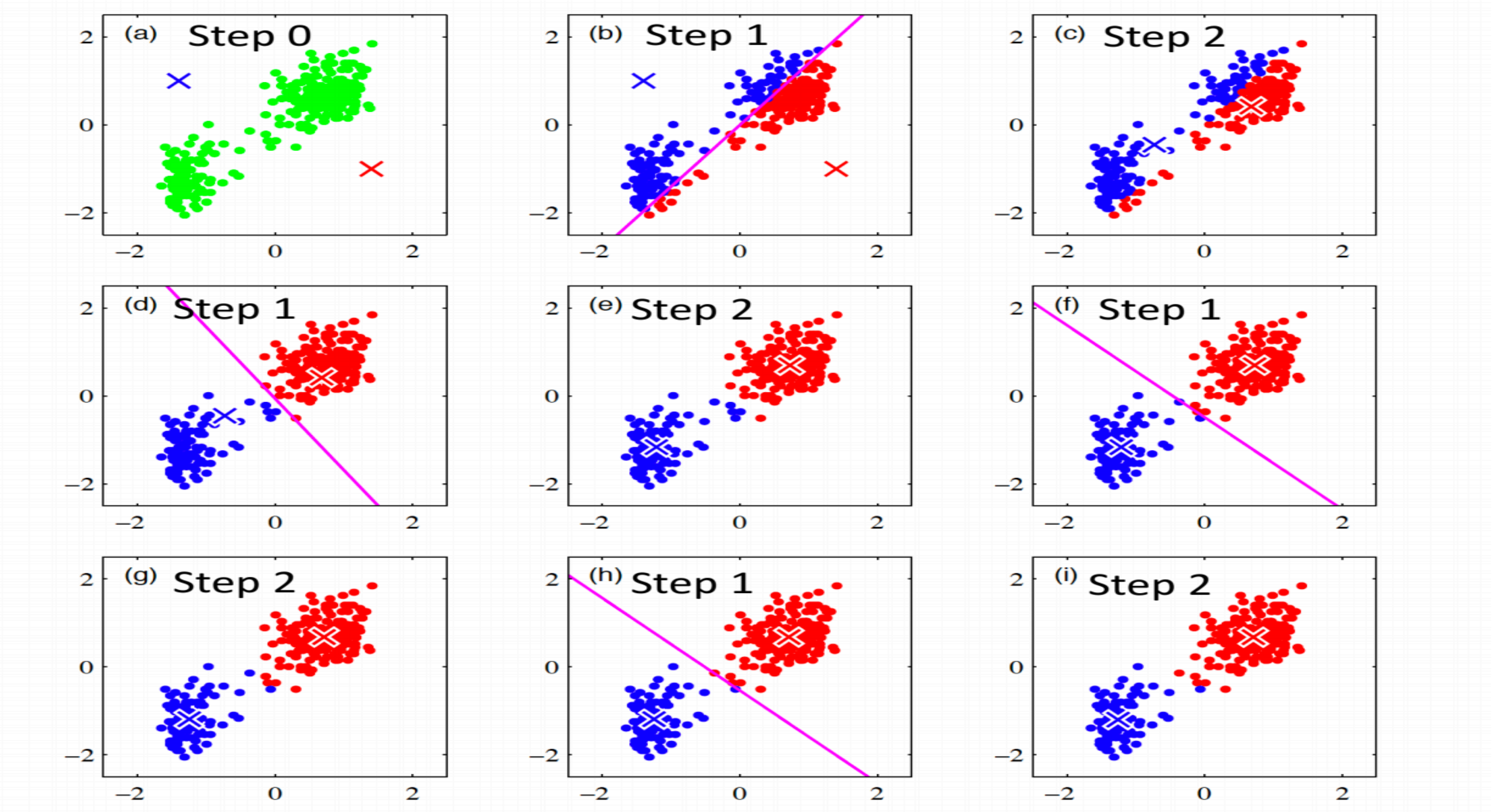
K-Means Clustering

Iterative K-means Algorithm

Example



Data points assigned to cluster k should be close to center of "mass" of the k-th cluster

Cost Function

- We have N data points x_n and we want to have K clusters
- Cost function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} || x_n - \mu_k ||^2$$

where μ_k is the k-th cluster center, r_{nk} = 1 if x_n is in the k-th cluster
0, otherwise

- Our goal is to determine μ_k
- It is not a convex function

K-means Clustering Algorithm

K-means algorithm
a.k.a Llyod’s algorithm

- Step 0: randomly assign the cluster centers $\{\mu_k\}$
- Step 1: Minimize J over $\{r_{nk}\}$ -- Assign every point to the closest cluster center

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Step 2: Minimize J over $\{\mu_k\}$ -- update the cluster centers

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\boldsymbol{x}_n}{\sum_n r_{nk}}$$

- Loop until it converges

K-means Implementation

- Given a vector x
- Calculate $k^* = \arg\min_k || x - \mu_k ||^2$
- Calculate the distances of x to the cluster centers and determine the closest center μ_{k*}

Remarks

- Prototype μ_k is the mean of data points assigned to the cluster k, hence `K-means'
- μ_k may not in the training set
- k need to be pre-defined
  - There are some other approaches for the case k is unknown – not cover in class
- The procedure reduces J in both Step 1 and Step 2 and thus makes improvements on each iteration

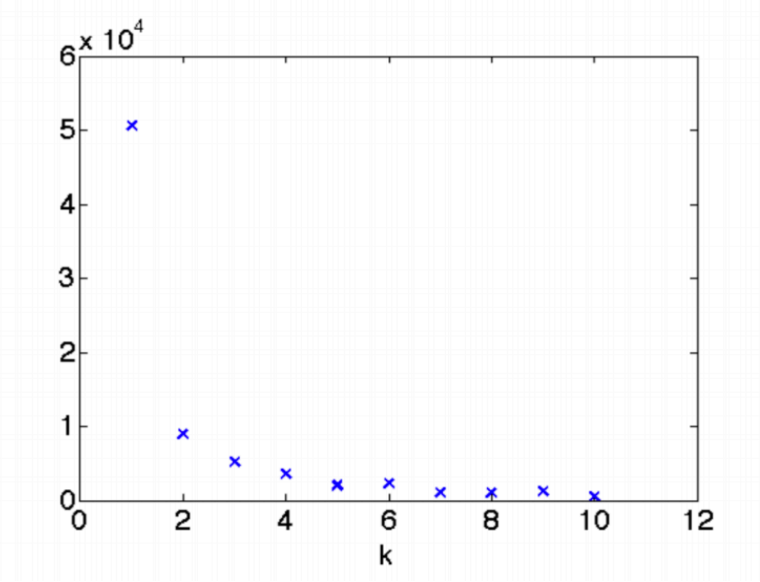
Properties of the K-means algorithm

- Does the K-means algorithm converge
  - Yes
- How long does it take to converge?
  - In the worst case, exponential in the number of data points
  - In practice, very quick
- How good is its solution?
  - Local minimum (depends on the initialzation)

- Try the iterations with several initial conditions and pick the solution with the lowest cost value J
- It can be sensitive to outliers!

Choosing K

- Increasing K will always decrease the optimal value of the K-means objective.
  - Analogous to overfitting in supervised learning.



The cost function $J = 0$ when $K = N$ (Nearest – Neighbor Classifiers)

K-Medoids

- K-means is sensitive to outliers.
- In some applications we want the prototypes to be one of the points.
- Leads to K-medoids.
- Medoids are representative objects of a data set or a cluster with a data set whose average dissimilarity to all the objects in the cluster is minimal. Medoids are similar in concept to means or centroids, but medoids are always restricted to be members of the data set.

K-medoids algorithm

- Step 0: randomly selecting K points as the cluster centers $\{\mu_k\}$
- Step 1: Minimize J over $\{r_{nk}\}$ -- Assign every point to the closest cluster center

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Step 2: Update the cluster centers– the prototype for a cluster is the data that is closest to all other data points in the cluster

$$k* = \arg\min_{m: r_{mk}=1} \sum_n r_{nk} \|\boldsymbol{x}_n - \boldsymbol{x}_m\|_2^2$$

$$\boldsymbol{\mu}_k = \boldsymbol{x}_{k*}$$

- Loop until it converges