K-Mean

Ngô Minh Nhựt

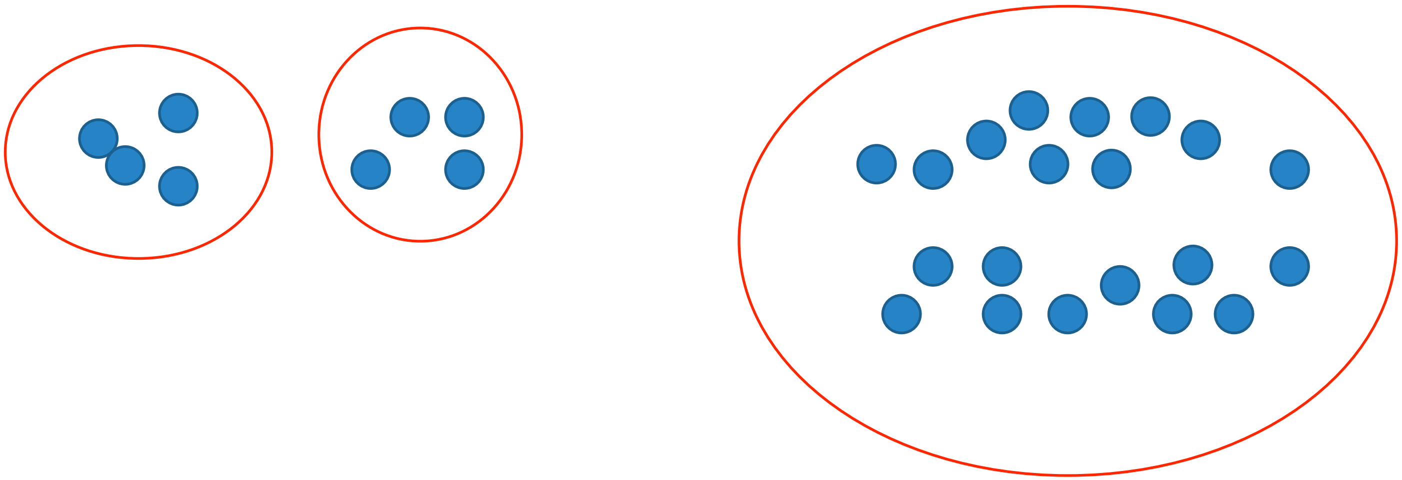
2021

Clustering

- Clustering is an unsupervised learning algorithm
- Dataset for training does not need to be labeled
- Used to recognized similar samples. For example:
  - Searching results,
  - Shopping habits, …
- Clustering is useful when there is not much information about data

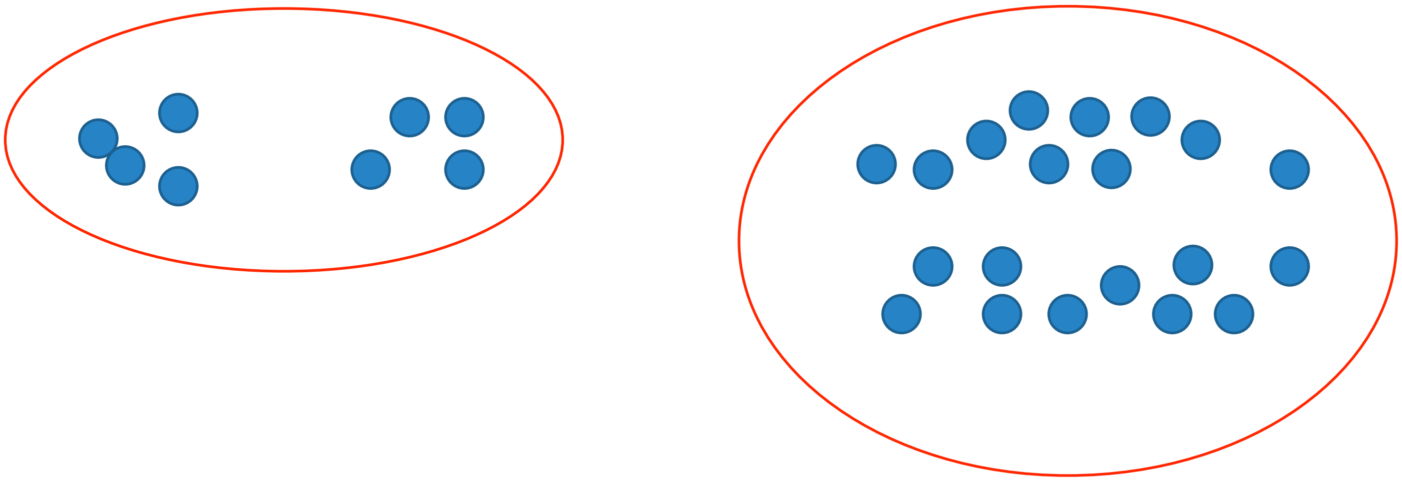
Introduction

- Idea about clustering:
  - Gather similar samples into one group
  - Example: given two dimension samples



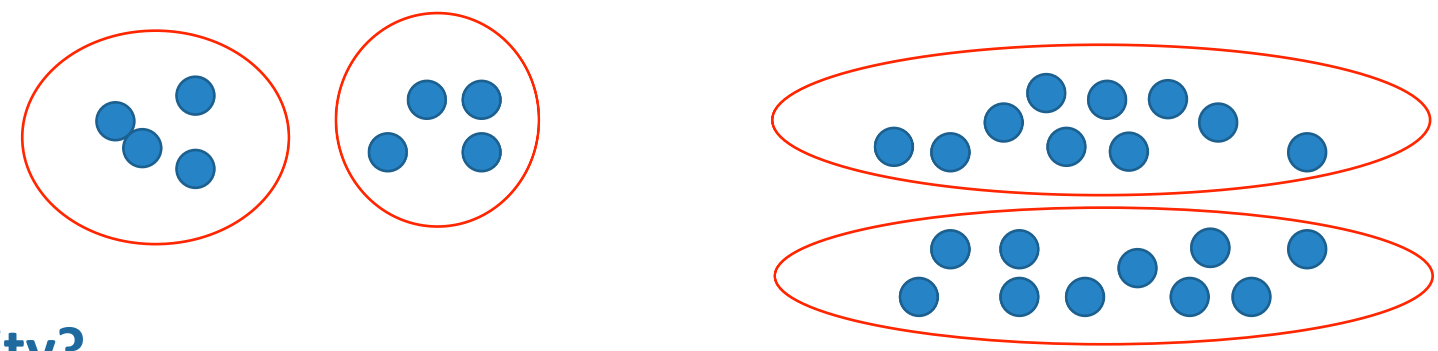
Introduction

- Idea about clustering:
  - Gather similar samples into one group
  - Example: given two dimension samples



Introduction

- Idea about clustering:
  - Gather similar samples into one group
  - Example: given two dimension samples



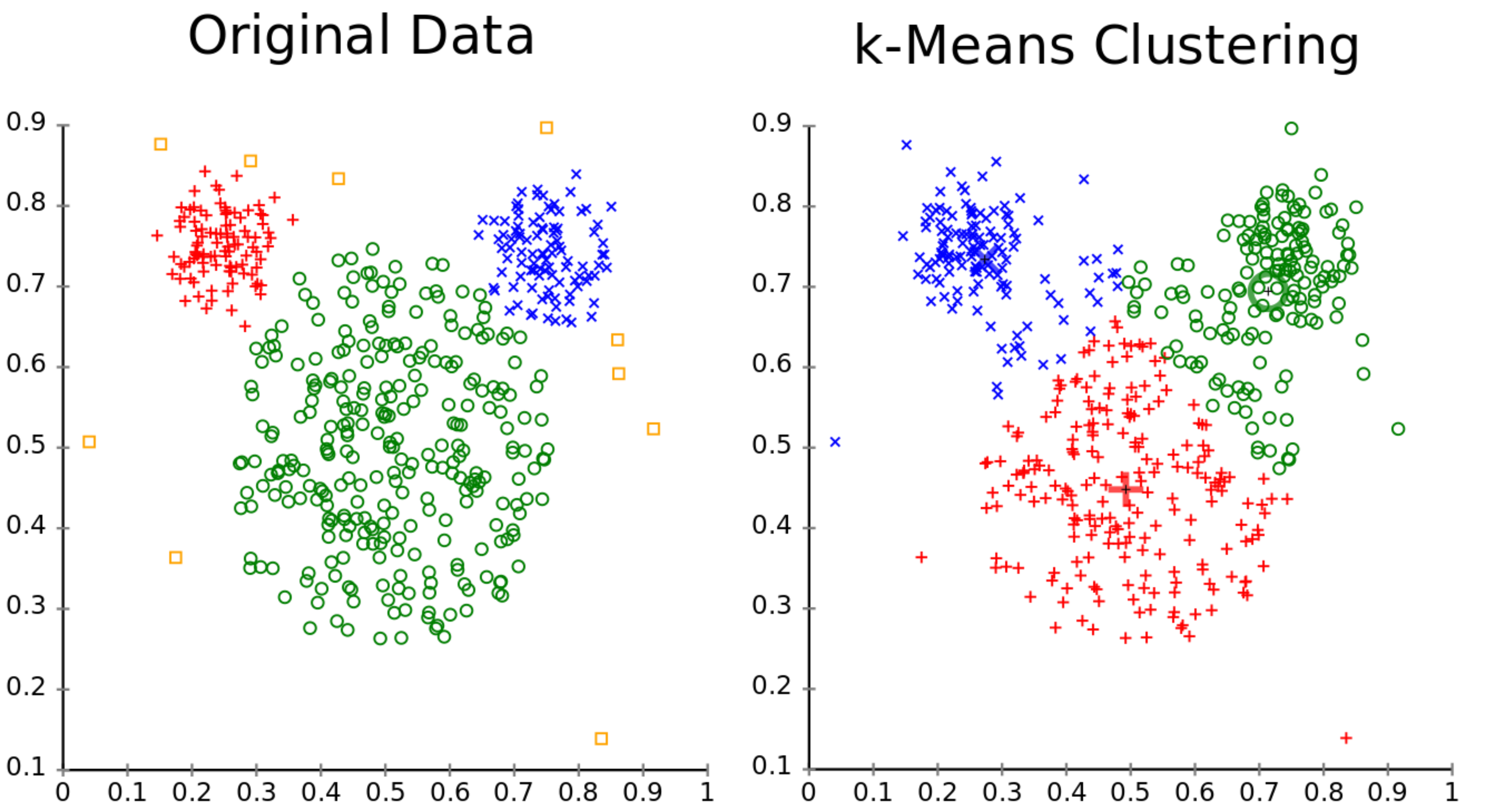
Similarity?

- Example: Euclide distance
- Clustering outcome depends on similarity caculation

K-mean

- K-mean is an unsupervised learning algorithm
- Used to cluster data: learn structure
- Based on Euclide distance: two samples have small distance will belong to one cluster

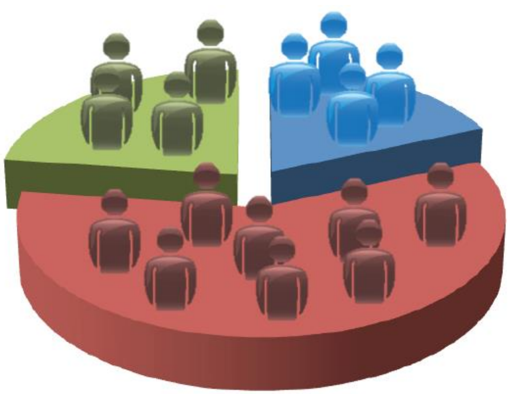
Introduction



Source: Wikipedia

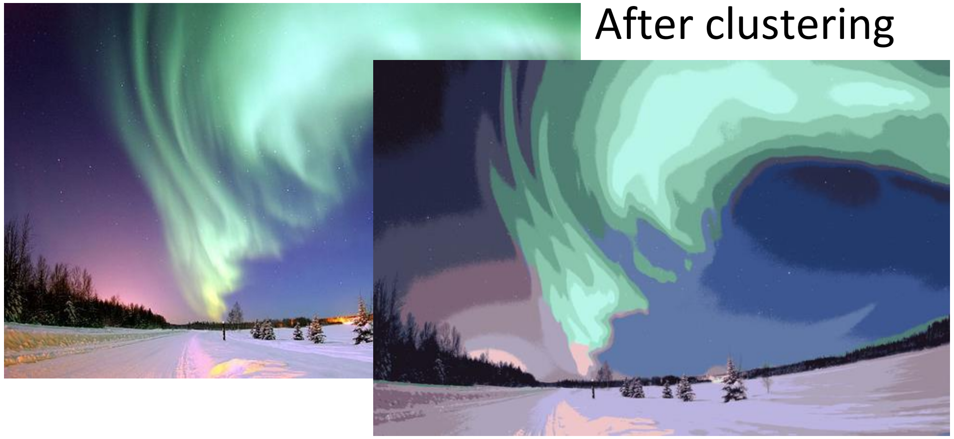
Application of clustering

- Computer science: image segmentation, recommender system, anomaly detection
- Social network analysis: clustering community, search result grouping
- Business marketing: dividing consumers into market segments



Original

After clustering



Source: Andrew Ng, Wikipedia

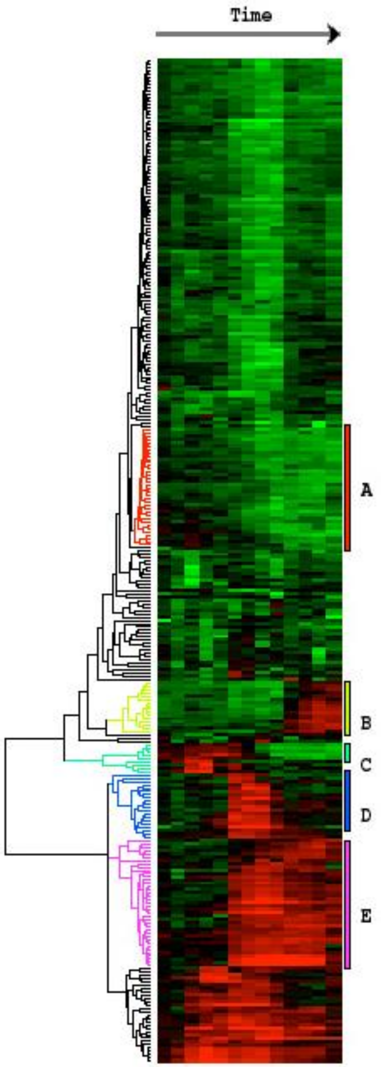
Application of clustering

- Image segmentation
  - Goal: segment image into regions meaningful or similar in term of visual perception

Source: James Hayes

Application of clustering

- Cluster data representing gene
- Goal: figure out similar gene samples



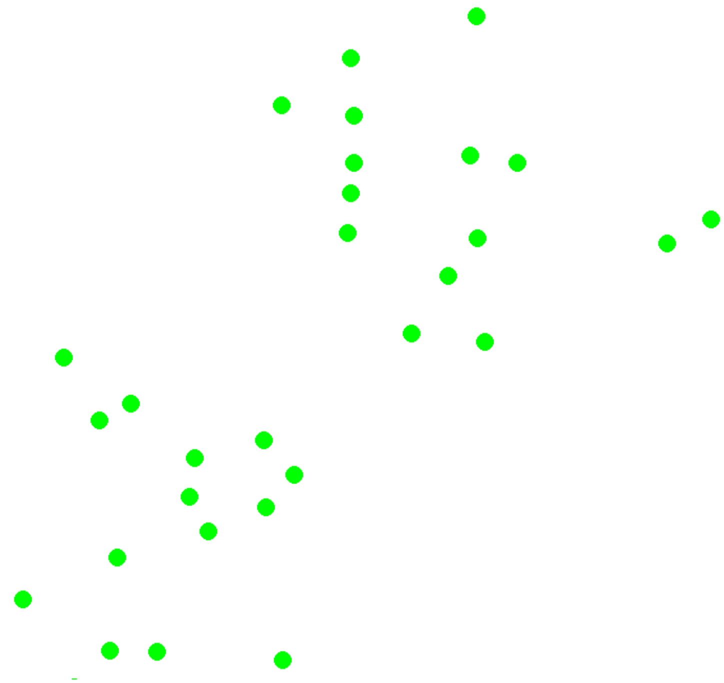
Source: Eisen et al, PNAS 1998

K-mean algorithm

❑ Input: number of clusters K, m data samples

❑ Goal: figure out clusters so that distance between samples and centroid is smallest

- Step 1: initialize K centroids
- Step 2: distribute samples into the nearest cluster
- Step 3: recalculate centroids
- Loop until convergence



Algorithm

- Initialize randomly K centroids: $\mu_1, \mu_2, \ldots, \mu_K$
- Loop until centroids do not change:
  - Loop i = 1 to m
    - $c^{(i)}$ = index of centroid which sample $x^{(i)}$ is nearest to
  - Loop k = 1 to K
    - μ_k = mean of samples clustered into cluster k

Cost function

- Given:
  - $c^{(i)}$: cluster of sample $x^{(i)}$
  - μ_k: centroid of cluster k
  - $\mu_{c^{(i)}}$: centroid which $x^{(i)}$ is assigned to
- Cost function:

$$J(c^{(1)}, ..., c^{(m)}, \mu_1, ..., \mu_K) = \frac{1}{m}\sum_{i=1}^{m}\|x^{(i)} - \mu_{c^{(i)}}\|^2$$

- Goal:

$$\min_{c^{(1)},...,c^{(m)},\mu_1,...,\mu_K} J(c^{(1)}, ..., c^{(m)}, \mu_1, ..., \mu_K)$$

Algorithm

- Initialize randomly K centroids: $\mu_1, \mu_2, \ldots, \mu_K$
- Loop until centroids do not change:
  - Loop i = 1 to m $\quad \min_{c^{(i)}} J(\ldots)$
    - $c^{(i)}$ = index of centroid which sample $x^{(i)}$ is nearest to
  - Loop k = 1 to K $\quad \min_{\mu_k} J(\ldots)$
    - μ_k = mean of samples assigned to cluster k

Centroid initialization

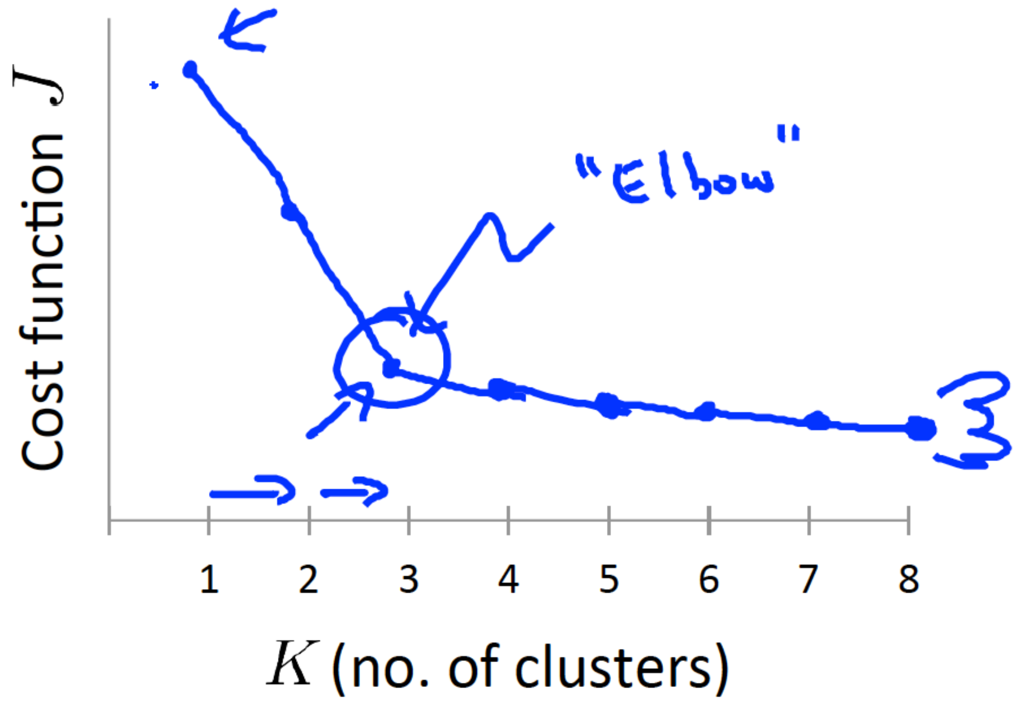
- Loop i = 1 to 100
  - Initialize randomly K centroid
  - Run K-mean algorithm
  - Calculate cost

$$J(c^{(1)}, ..., c^{(m)}, \mu_1, ..., \mu_K)$$

- Choose clusters having smallest cost

Choose number of centroids K

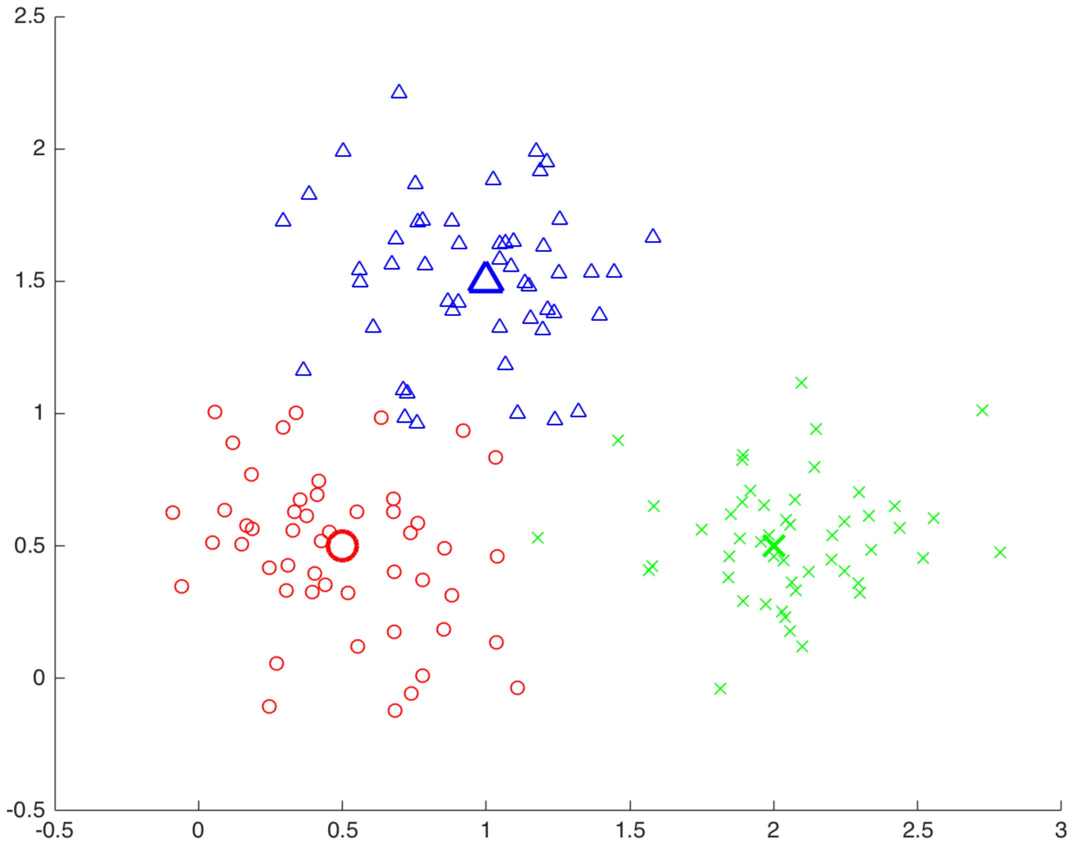
- Elbow method: choose K at the point which cost does not change from

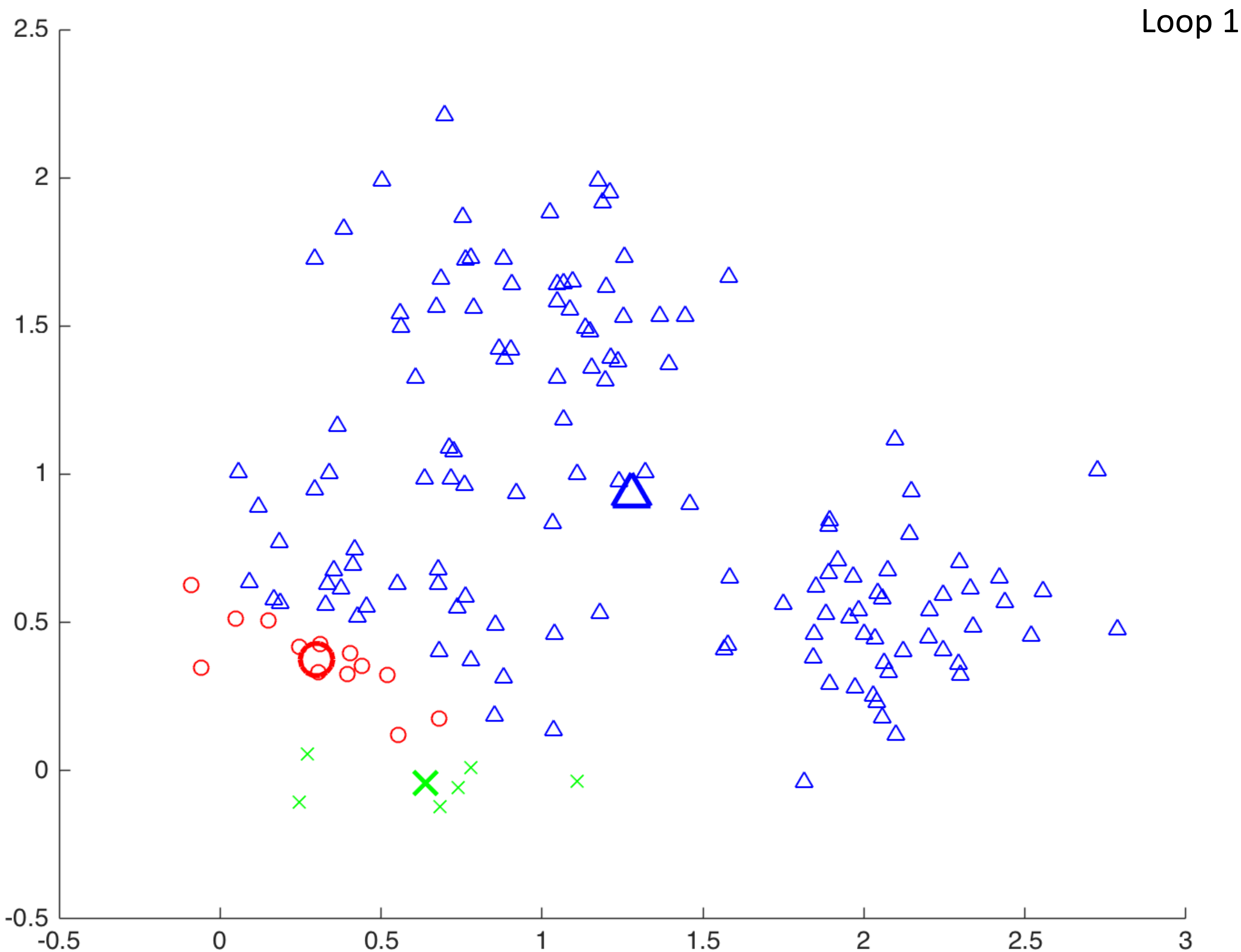


Other distances

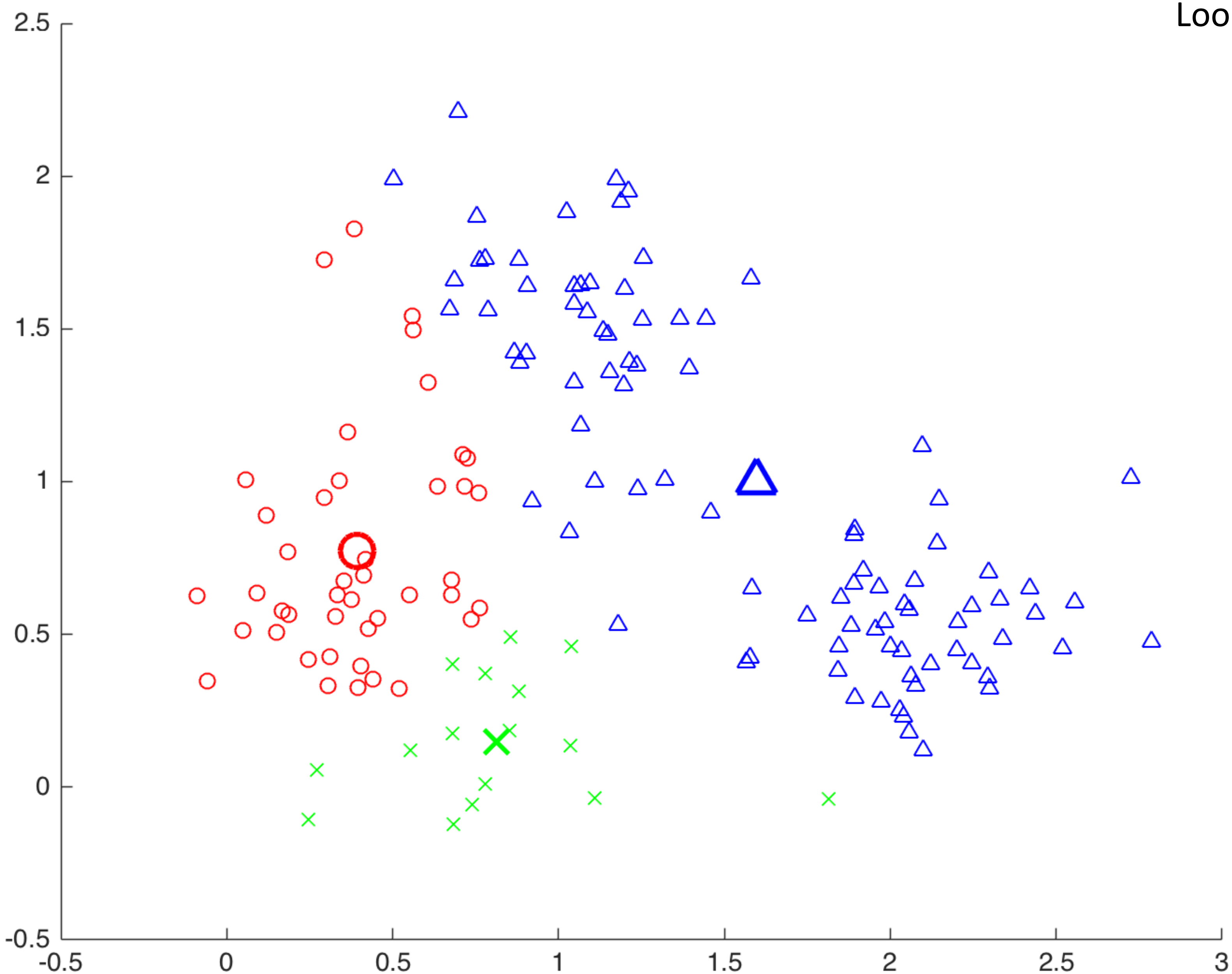
- Euclide distance
  - $d(x, y) = \sqrt{\sum_{i=i}^{n} |x_i - y_i|^2}$
- Manhattan distance
  - $d(x, y) = \sum_{i=i}^{n} |x_i - y_i|$, n: number of features
- Maximum norm
  - $d(x, y) = \max\limits_{1 \le i \le n} |x_i - y_i|$, n: number of features
- Cosine distance
  - $d(x, y) = 1 - \frac{x^T y}{\|x\| \|y\|}$, d is from 0 to 2
- Hamming distance
  - Number of different components between vectors x and y
  - Example: two vectors (0, 1, **1**) and (0, 1, **0**) have Hamming distance of 1

Example

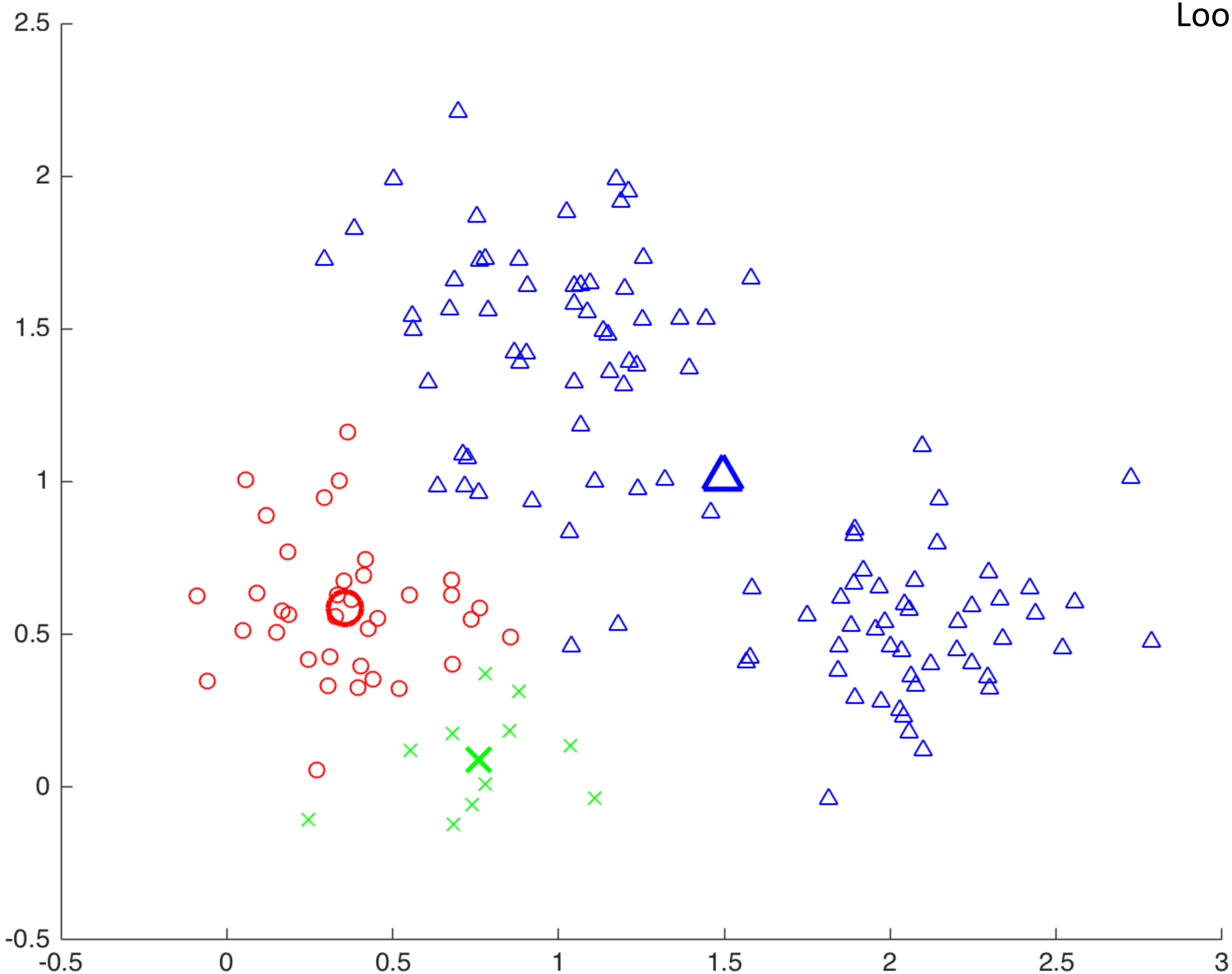



Loop 1
2.5
2
1.5
1
0.5
0
-0.5
-0.5
0
0.5
1
1.5
2
2.5
3

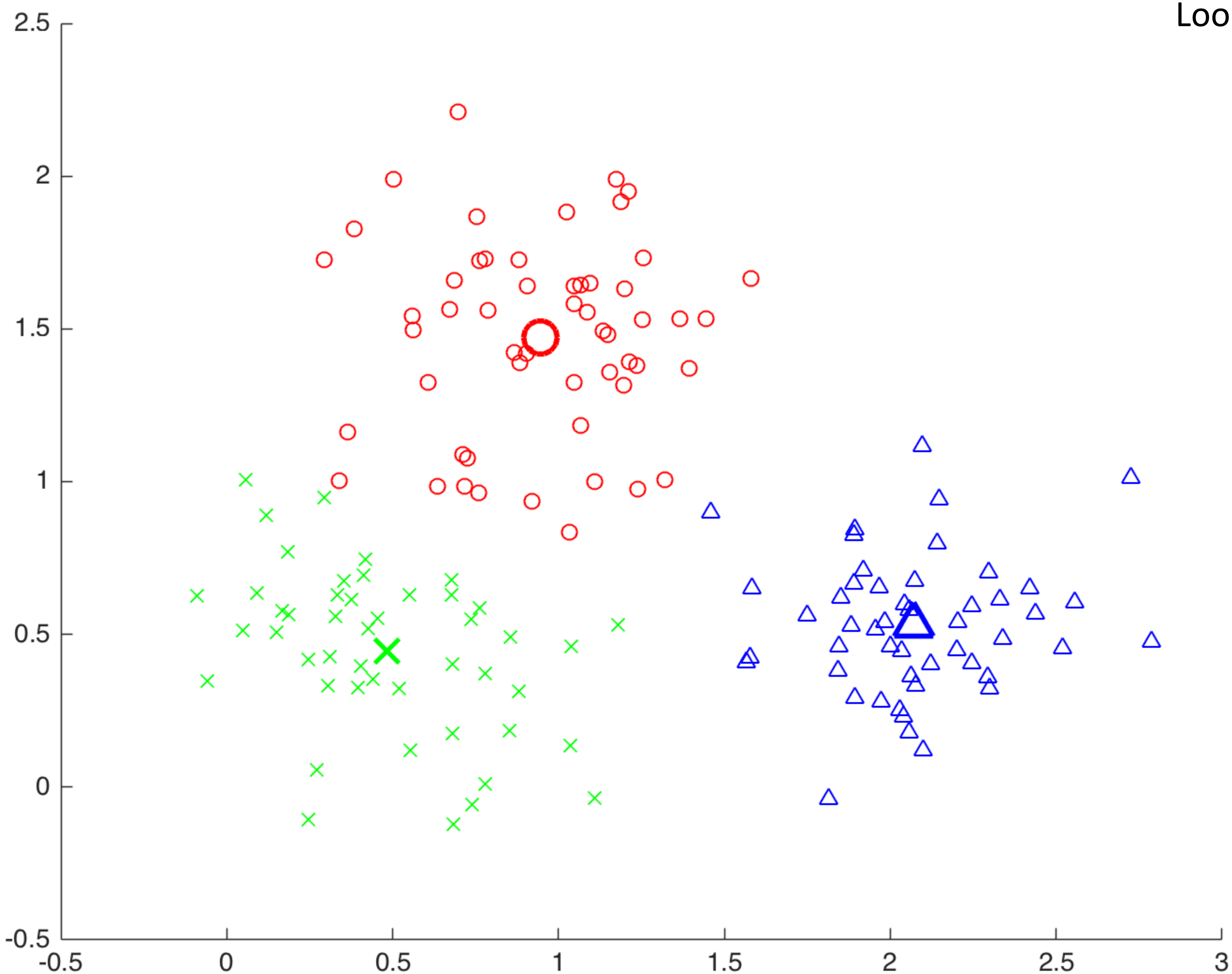
Loop 3



Loop 5



Loop 10



Advantages of k-mean

- Find out clusters having small variance
- Simple and fast
- Easy to implement

Disadvantages of k-mean

- Need to identify parameter K
- Affected by outliers
- Prone to local minimals
- Highly dependent on clusters initialization
- Could be slow. Time complexity of each iteration is: O(Kmn), m is number of samples, n is number of features