Clustering (K-means)

9/21

Given dataset $D = \{\vec{x}_1, \cdots, \vec{x}_n\}$ $\vec{x}_i \in \mathbb{R}^d$
$k \in \mathbb{Z}^+$

Partition D in k partions (clusters) $\mathcal{C} = \{C_1, \cdots, C_k\}$
($\mathcal{C}$: script)

Representative clustering

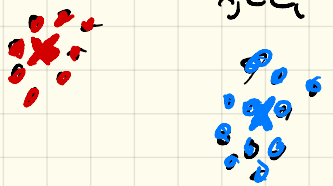
$\forall C_i \; \exists$ rep point to summarize the cluster

commonly mean/center

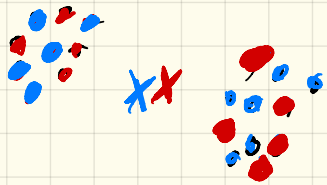
e.g. for $C_i \ni |C_i| = n_i$

$$\vec{\mu}_i = \frac{1}{n_i} \sum_{\vec{x}_j \in C_i} \vec{x}_j$$

Good!



Bad!



How to get started:

1. opt criterion
2. generate all k-partions of points
3. pick the partion w/ highest (lowest) score

How many ways to create k-partions of n points

$O(\frac{k^n}{k!})$ ... even for tiny n too much work

K-means algo

1. pick opt criterion — Sum of squared error

$$SSE(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\vec{x}_j \in C_i} \| x_j - \mu_i \|^2$$

Goal: find $\mathcal{C}^* = \underset{\mathcal{C}}{\operatorname{argmin}} \{ SSE(\mathcal{C}) \}$

↑ clustering that minimizes SSE

Idea of k-means
- greedy
- iterative

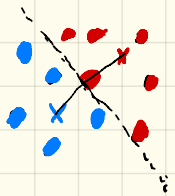
Note: we can get stuck in a local opt

High level

1. init each μ_i
   (typically, randomly sample from range of each dim or bounding box of data if range unknown)

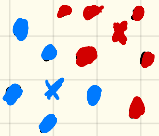
2. at each iteration
   a. cluster assignment

assign each point to closest center

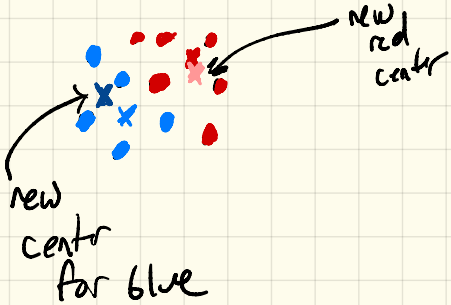


assign each $\vec{x}_j$ to cluster C_{i^*} w/ $i^* = \underset{i=1}{\overset{k}{\operatorname{argmin}}} \left\{ \| x_j - \mu_i \|^2 \right\}$

   b. centroid update



update center to mean of points in cluster



3. continue until centers don't move too much
   e.g. μ_i^t is center for cluster i at iteration t

   $\varepsilon \geq 0$ (user specified)

   stop when $\sum_{i=1}^{k} \| \mu_i^t - \mu_i^{t-1} \| < \varepsilon$

Algorithm 13.1: K-means Algorithm

K-MEANS $(\mathbf{D}, k, \epsilon)$**:**

$t = 0$

Randomly initialize k centroids: $\boldsymbol{\mu}_1^t, \boldsymbol{\mu}_2^t, \ldots, \boldsymbol{\mu}_k^t \in \mathbb{R}^d$

repeat

  $t \leftarrow t + 1$

  $C_i \leftarrow \emptyset$ for all $i = 1, \cdots, k$

  `// Cluster Assignment Step`

  foreach $\mathbf{x}_j \in \mathbf{D}$ **do**

    $i^* \leftarrow \arg\min_i \left\{ \|\mathbf{x}_j - \boldsymbol{\mu}_i^{t-1}\|^2 \right\}$

    $C_{i^*} \leftarrow C_{i^*} \cup \{\mathbf{x}_j\}$ `// Assign` $\mathbf{x}_j$ `to closest centroid`

  `// Centroid Update Step`

  foreach $i = 1, \cdots, k$ **do**

    $\boldsymbol{\mu}_i^t \leftarrow \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j$

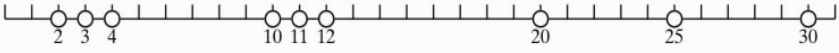
until $\sum_{i=1}^k \|\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1}\|^2 \leq \epsilon$

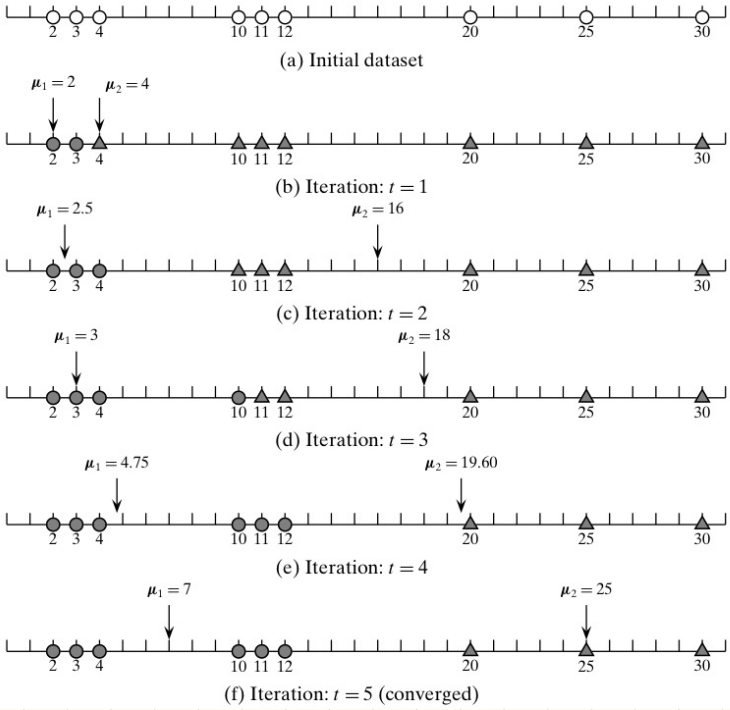
Things to note:

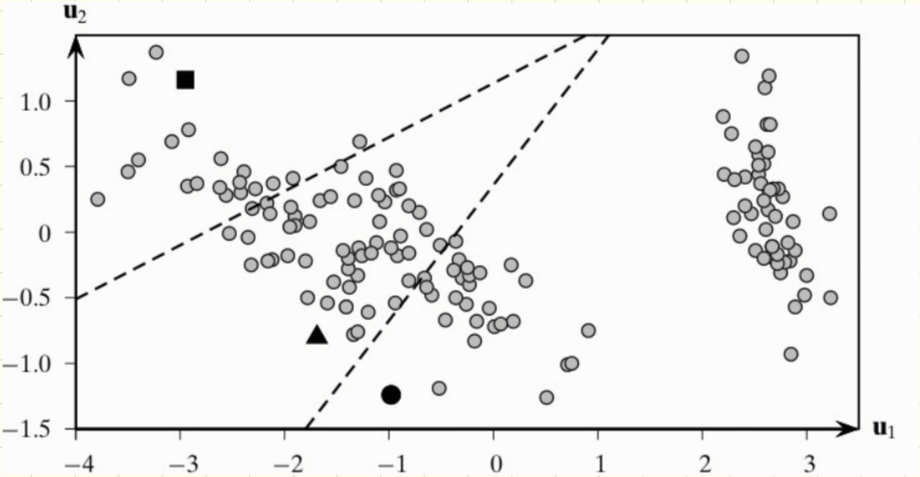
- usually run the algo a few times w/ diff initial centers report the clustering w/ min SSE (helps avoid really bad initial guesses)

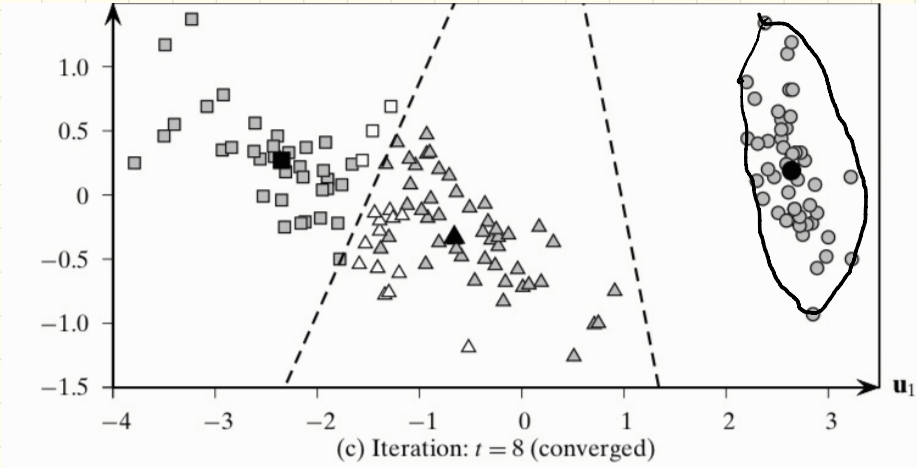
- Cluster are convex

each color is not convex which is a problem for k-means (come back to later)


2 3 4
10 11 12
20
25
30


2 3 4
10 11 12
20
25
30
(a) Initial dataset
$\mu_1 = 2$
$\mu_2 = 4$
(b) Iteration: $t = 1$
$\mu_1 = 2.5$
$\mu_2 = 16$
(c) Iteration: $t = 2$
$\mu_1 = 3$
$\mu_2 = 18$
(d) Iteration: $t = 3$
$\mu_1 = 4.75$
$\mu_2 = 19.60$
(e) Iteration: $t = 4$
$\mu_1 = 7$
$\mu_2 = 25$
(f) Iteration: $t = 5$ (converged)





(c) Iteration: $t = 8$ (converged)

Computational Complexity for K-means

a. Cluster assignment
   for each point
      for each cluster
         compute dist

time proportal to
n
k
d
$O(nkd)$

b. Recompute Center
   for each cluster
      recompute center

$d * \#$ of points in the cluster (R)
$\Rightarrow O(dn)$

Note each point is in exactly 1 cluster

Work at each iteration
$O(nkd + dn) = O(nkd)$

Assume t iteration
total work is $O(tnkd)$

I/O complexity

	# of DB passes proptal to
a. Cluster assignment for each point for each cluster compute dist	1
b. Recompute Center for each cluster recompute center	1

I/O measued in # of DB passes

each iter $O(1)$ db scans

Total I/O for t iterations is $O(t)$