Center-Based Clustering

- Setting: N observations $\vec{x}_1, \ldots \vec{x}_N \in \mathbb{R}^d$
- Goal: Find clusters (subsets/groups) of similar points
- Output:
  - No dendrogram
    - unlike Hierarchical Clustering
      - # of clusters must be specified in advance
      - $(k+1)$ clustering of data (likely) does not result from splitting a cluster in (k)-clustering result
        - cannot be constructed as an iterative algorithm
          - specify an objective and (try to) optimize
  - Cluster centers $\{c_1, \ldots, c_k\} \subset \mathbb{R}^d$, $k = \#$ desired clusters and partition $\mathcal{C} = \{C_1, \ldots C_k\}$, $C_i \subset \{\vec{x}_1, \ldots, \vec{x}_N\}$ gives samples in cluster i.

- Methods:

  k-means:

  Centers: $c_i = \frac{1}{N_i} \sum_{x_j \in C_i} \vec{x}_j$, $N_i = |C_i|$ (assuming Euclidean distance!)

  Objective:

$$\Phi(\mathcal{C}) = \sum_{j=1}^{k} \sum_{x_i \in \mathcal{C}_j} d^2(x_i, \vec{c}_j)$$

typically Euclidean distance

* minimize Sum of Squares

squared distance places greater emphasis on large distances!

k- CENTER

OBJECTIVE:

$$\Phi_{CENTER}(C) = \max_{j=1..k} \max_{x_i \in C_j} d(x_i, c_j)$$

* MINIMIZE THE MAXIMUM DISTANCE BETWEEN ANY DATA POINT AND ITS CENTER

* CENTERS $c_1 \dots c_k$ DO NOT HAVE EXPLICIT REPRESENTATIONS
  - FOR FIXED k THIS CAN BE EXPRESSED AS A GRAPH THEORY PROBLEM, NP-COMPLETE ALGORITHMS
  - DETERMINING k IS NP-HARD

* CENTERS $c_1, \dots c_k$ GENERATE VORONOI PARTITIONS

* REQUIRE APPROXIMATE ALGORITHMS

  NO GLOBAL OPTIMAL SOLUTION W/O A LOT OF LUCK!

k-MEDIAN / k-MEDOIDS

- CENTERS:

$$c_i = \underset{\vec{x} \in C_i}{\operatorname{argmin}} \sum_{x_j \in C_i \setminus \{x\}} d(\vec{x}, x_j)$$

- OBJECTIVE:

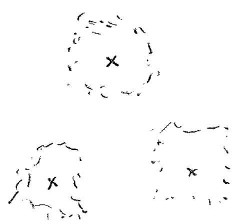
$$\Phi_{k\,MED}(C) = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(\vec{x}_i, c_j)$$

USES DISTANCE (NOT SQUARED) AS OPPOSED TO k-MEANS

* CENTERS ARE GIVEN BY ACTUAL DATA POINTS (REPRESENTATIVES!)

* A VARIANT CALLED PARTITIONING AROUND MEDOIDS (PAM) ALSO EXISTS.

Comparison

k-center	k-medoid	k-means
Need to compute centers from data - Need $\vec{x}_1, \dots, x_N$	Uses only original distance at each step - only requires D - don't need quant. data	Need to compute centers from data - Need $\vec{x}_1, \dots, \vec{x}_N$ - Need metric & quantitative data
Uses $d(x_i, c_j)$ - less sensitive to outliers	Uses $d(x_i, c_j)$ - less sensitive to outliers	Uses $d^2(x_i, c_j)$ - greater emphasis on large distances - sensitive to outliers
Centers estimated from data - centers need not be representative 	Centers chosen from data "representative objects"	Centers estimated from data - centers need not be representative

Minimizing Objectives

- Simple (Combinatorial) Algorithm

For fixed K, we could try all possible clustering assignments

- \# of distinct assignments

$$S(N,k) = \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^N$$

label switching

- $S(10,4) = 34{,}105$
- $S(19,4) \approx 10^{10}$

exponential in N!

* must use iterative/approx./greedy algorithms

Ex: k-MEANS (LLOYD'S ALGORITHM)

1) START W/ k CENTERS

2) CLUSTER EACH POINT TO ITS NEAREST CENTER

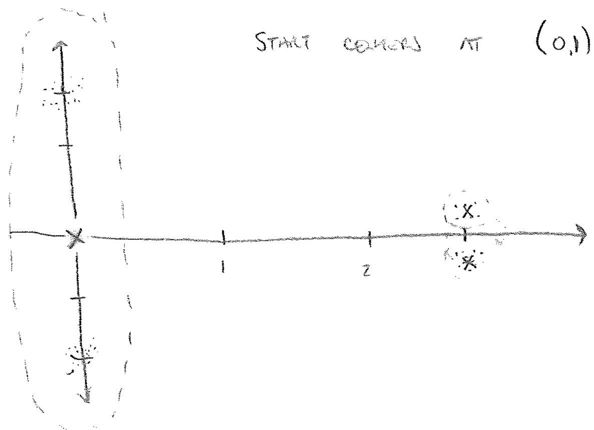
3) FIND THE NEW CLUSTER CENTERS

$$* \quad c_i = \frac{1}{N_i} \sum_{x_j \in C_i} \vec{x}_j \quad \text{IN CASE OF EUCLIDEAN DISTANCE}$$

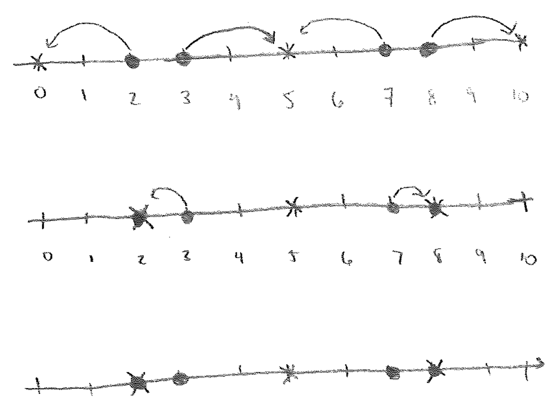
4) REPEAT 2, 3 TO CONVERGENCE !

a)

START CENTERS AT $(0,1)$ $(3,+\varepsilon)$ $(3-\varepsilon)$



b)



CENTERS A 0, 5, 10

CENTER	MEMBERS
2	2, 3
5	ϕ
8	7, 8

* COLLAPSE OF ALGORITHM IF TOO MANY CLUSTERS CHOSEN (TRY MANY k IF UNKNOWN)

* POOR FIT IF BAD INITIAL CONDITIONS CHOSEN (TRY MANY ICs!)

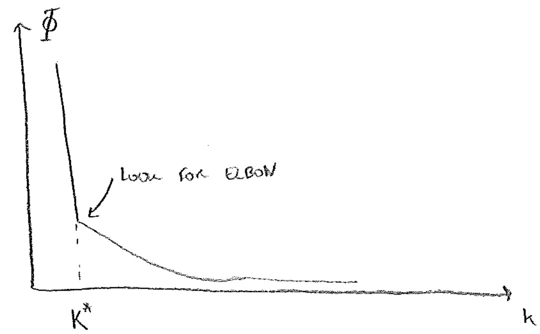
Choosing the Number of Clusters

- A lot of different methods
- Informal (Scree plot) option

  Assuming optimal solutions are found

  $$\Phi_{kmeans}, \; \Phi_{kcenter}, \; \Phi_{kmed}$$

  are decreasing in the # of clusters



- Covariance Decomposition for k-means w/ Euclidean distance
  - Given clustering assignments $C(1), \ldots C(N) \in \{1, \ldots k\}$

$$N\hat{\Sigma} = \sum_{j=1}^{N} (x_j - \bar{x})(x_j - \bar{x})^T = \sum_{j=1}^{k} \sum_{x_i \in C_j} (\vec{x}_i - \bar{x})(\vec{x}_i - \bar{x})^T$$

$$= \sum_{j=1}^{k} \sum_{x_i \in C(j)} \left((\vec{x}_i - \vec{c}_j)(\vec{x}_i - \vec{c}_j)^T + (c_j - \bar{x})(c_j - \bar{x})^T + \cancel{(x_i - c_j)(c_j - \bar{x})^T} + \cancel{(c_j - \bar{x})(x_i - c_j)^T} \right)$$

$$\sum_{x_i \in C(j)} (x_i - c_j) = \vec{0}$$

$$= \underbrace{\sum_{j=1}^{k} \sum_{x_i \in C(j)} (x_i - c_j)(x_i - c_j)^T}_{\text{TOTAL WITHIN GROUP COVARIANCE}} + \underbrace{\sum_{j=1}^{k} N_j (c_j - \bar{x})(c_j - \bar{x})^T}_{\text{BETWEEN GROUP COVARIANCE}}$$

$$= \Phi_{kmeans}(C) + B(C)$$

CALINSKI & HARABAZ (1974)

CHOOSE k MAXIMIZING

$$C(k) = \frac{tr\left(B(\mathcal{C})\right)/_{(k-1)}}{tr\left(\Phi(\mathcal{C})\right)/_{(N-k)}}$$

MEAN BETWEEN GROUP VARIANCE

MEAN WITHIN GROUP VARIANCE

* NOTE CONNECTION/SIMILARITY TO ANOVA

- GAP STATISTIC (TIBSHIRANI ET AL 2001)

$$\mathrm{GAP}_N(k) = \underbrace{E_N^*\left(\log \Phi_k(\mathcal{C})\right)} - \log \Phi_k(\mathcal{C})$$

GENERATE MULTIPLE SAMPLES FROM A SUITABLY CHOSEN NULL DISTRIBUTION, MINIMIZE $\Phi_k(\mathcal{C})$ W/ k CLUSTERS TO ESTIMATE

CHOOSE k MAXIMIZING $\mathrm{GAP}_N(k)$

* ALSO RELATED TO HYPOTHESIS TESTING

- NOTE: THE REFERENCE DISTRIBUTION IS GENERATED USING PCA