Assignment-based Clustering

Feb 8, 2018

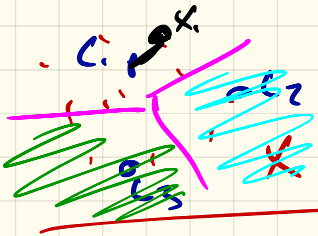
- k-center
- k-means
- k-median / mediod
- k-means++
- Mixture of Gaussians (EM)

Homework #1

Largest setting d	Q1D birthday	Q2D coupon coll
min	0.45	18
mean	300 ~5 minutes	3000 ~1 hour
median	30	1100
max	15,000 ~4 hours	80,000 ~22 hours

(seconds)

$\|c_1 - x_i\|$



Input

- $X \subset \mathbb{R}^d = \{x_1, x_2, \ldots, x_n\}$
- distance $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

  $d(x_1, x_2) = \|x_1 - x_2\|$
- k = # clusters

Outputs : Centers $C = \{c_1, c_2, \ldots, c_k\} \in \mathbb{R}^d$

Mapping $\phi_C : \mathbb{R}^d \to C$

$$\phi_C(x) = \arg\min_{c_i \in C} \|x - c_i\|$$

- **k-means** formulation

$$\text{Cost}_2(X, C) = \sum_{x \in X} d(x, \phi_C(x))^2$$

- **k-median** formulation

$$\text{Cost}_1(X, C) = \sum_{x \in X} d(x, \phi_C(x))$$

k-mediod $\leftarrow$ same but $C \subset X$

k-center

$$\text{Cost}_\infty(X, C) = \max_{x \in X} d(x, \phi_C(x))$$

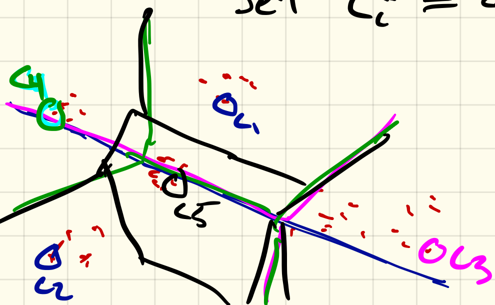
Gonzalez Alg. for k-center Clustering

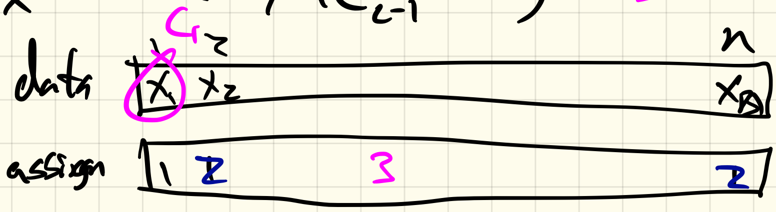
- NP-hard to solve w/in factor 2 of OPT
- Gonz Alg: 2-apx OPT, in metric d.

1. Choose center $c_1 \in X$ arbitrarily
   Let $C_1 = \{c_1\}$ $\quad C_i = \{c_1, c_2, \dots c_i\}$

2. for $i = 2$ to k do
   Set $c_i = \arg\max_{x \in X} d(x, \phi_{C_{i-1}}(x))$

$[c_1\ c_2\ \dots c_k]$





Lloyd's Alg. for k-means

$D =$ Euclidean $\quad X \subset \mathbb{R}^d$

1. Choose k centers $C \subset X$ (arbitrarily)

2. repeat

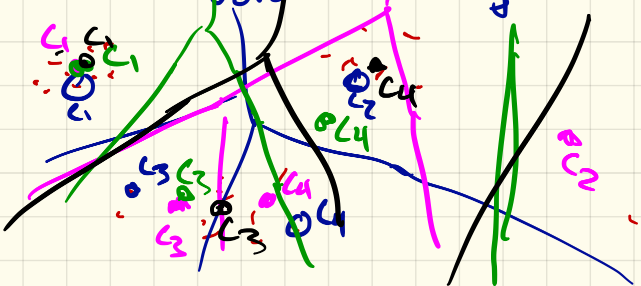
both a, b cost$_z$ decreases

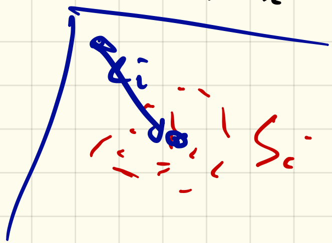
   a. For all $x \in X$, set $x \to c_i = \phi_C(x) = \arg\min_{c_i \in C} \|x - c_i\|$

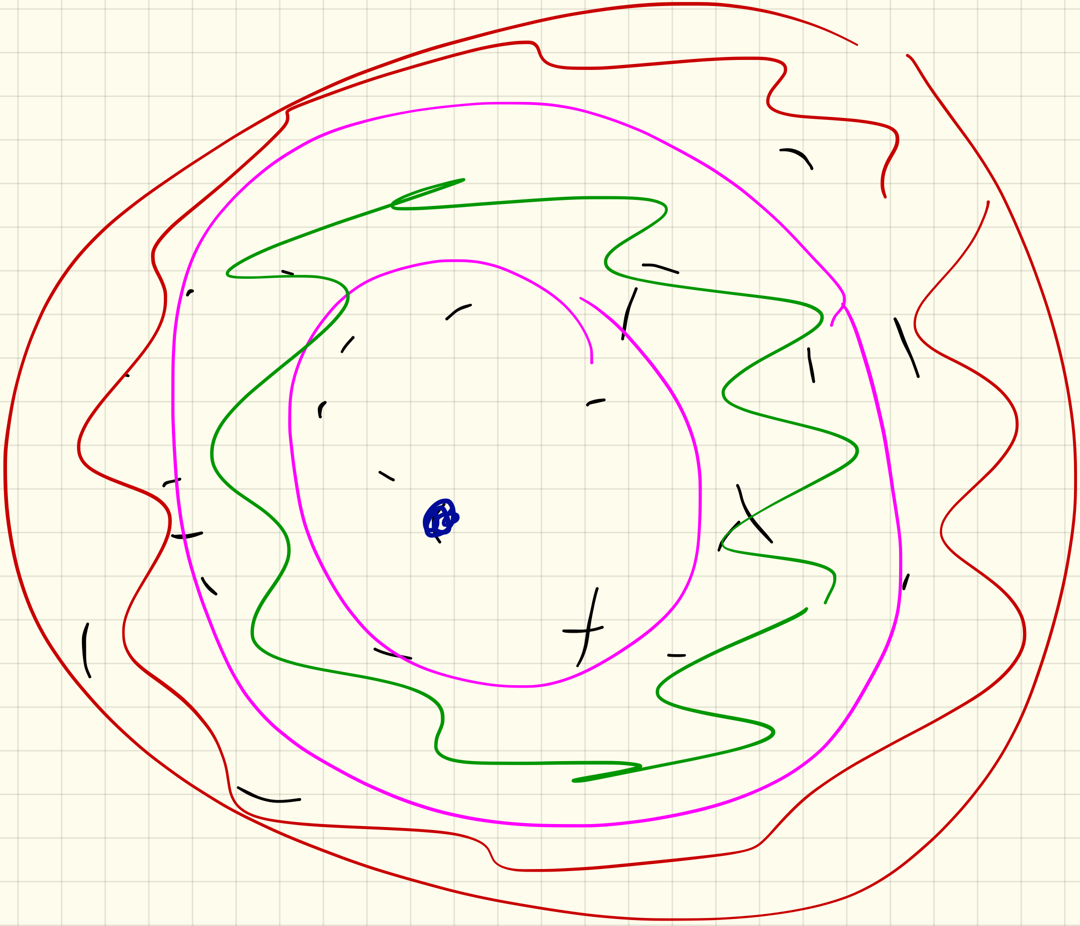
   b. For all $c_i \in C$, set $c_i = \text{average}\{x \in X \mid \phi_C(x) = c_i\} = S_i$ ("such that")

   $c_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$

until "converged"

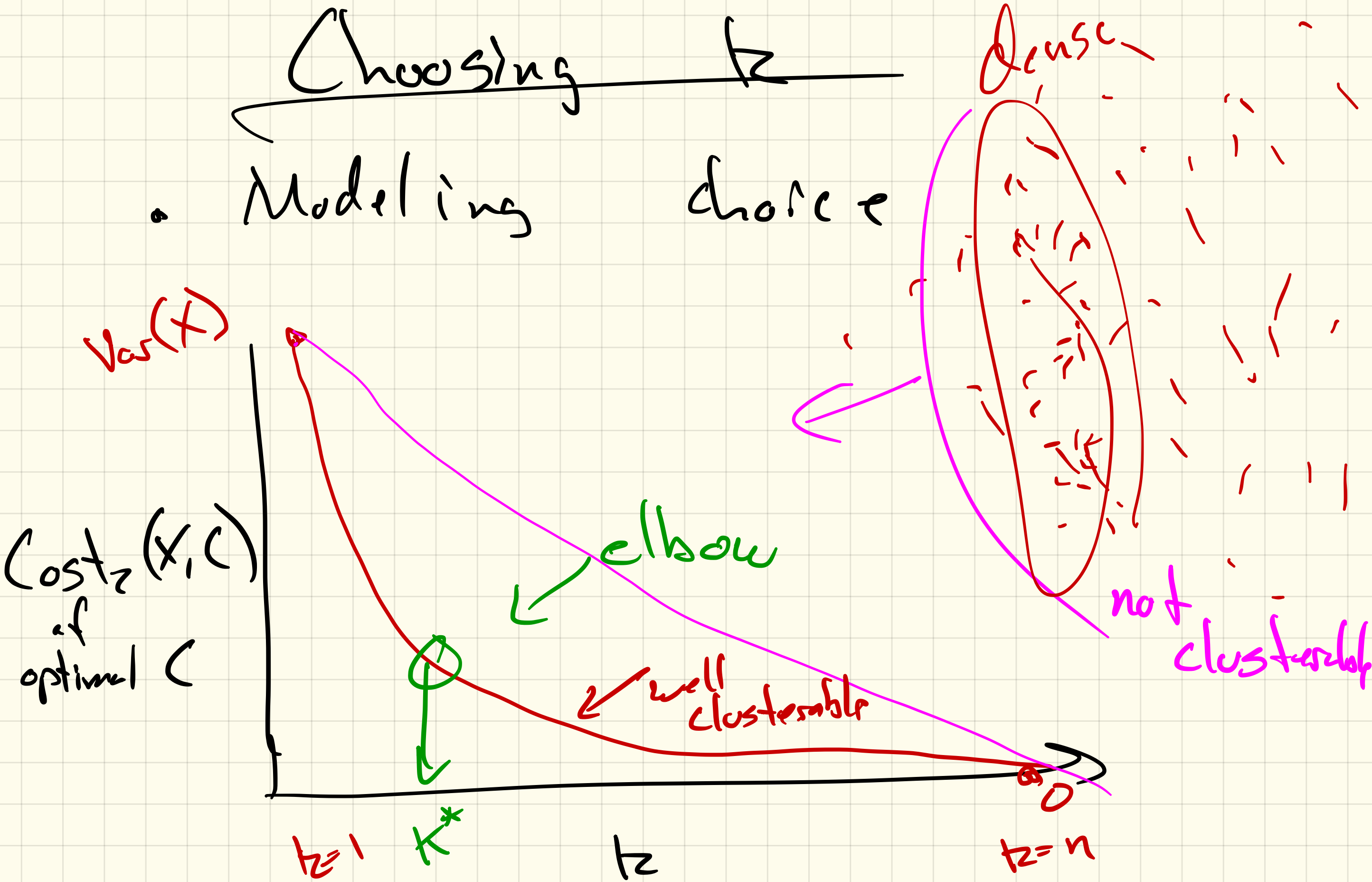


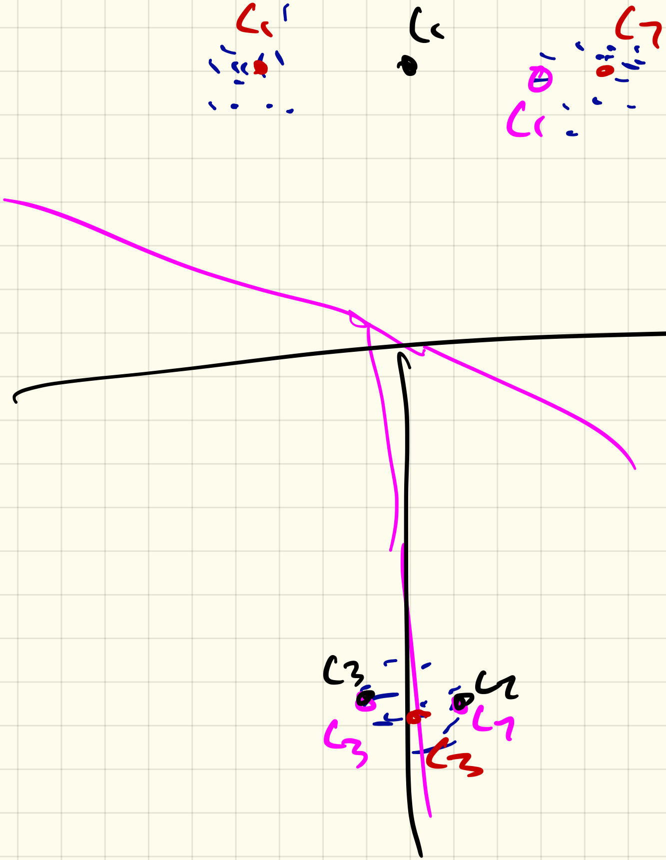




Choosing k

- Modeling choice





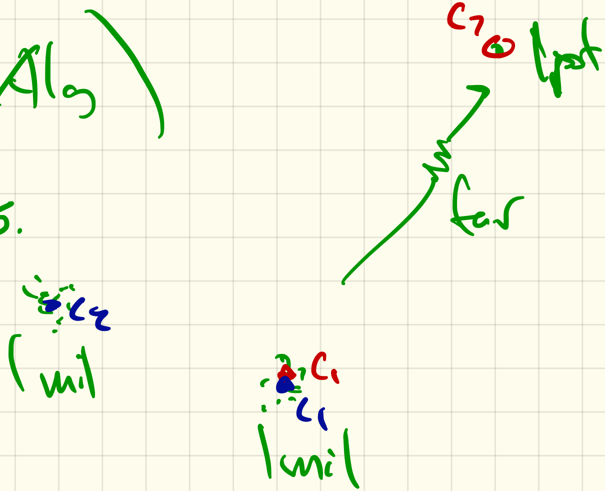
local minimum of cost

if happens

↳ random restart.

How to choose initial centers?

- k-center (Gonz Alg)
- Choose $O(k \log k)$ centers.
  "coupon collectors"
  ↳ then cluster
  then merge.



- k-means ++

k-means++ Algorithm "D^2-sampling"

1. Choosing $c_1 \in X$ arbitrarily

   $C_i = \{c_1, c_2, \ldots c_i\}$

2. for $i = 2$ to k

   Choose $c_i \in X$ w/ probability proportional

   $$w_x = D(x, \phi_{C_i}(x))^2$$

$$W = \sum_{x \in X} w_x$$

$$P_x = \frac{w_x}{W}$$