Approximation Algorithms for Clustering

08 March 2021 10:00

Given 'n' objects, dissimilarity function

$$d(\cdot) : [n] \to \mathbb{R}_{\geq 0} ,$$

group these into 'clusters' so that similar points are more likely in the same cluster.

Example Use Cases :-

1. Categorizing data (websites/documents) based on content.
2. Clustering songs into ragas.
3. cluster a city into neighborhoods to place utilities.
4. Ensuring diversity in a committee (first find clustering & then pick the committee by choosing from each cluster).

OUR MODELING in this COURSE

GIV

- n points, in a generic 'metric space'

way of formalizing a 'dis'-similarity function.

I
V
E
N

a dis-similarity function.

- Target 'k'.
- Objective Function f

Goal: Partition $[n]$ to clusters $S_1 \uplus S_2 \uplus \cdots S_k$

$$S_i \cap S_j = \phi$$
$$\bigcup_{i=1}^{k} S_i = [n].$$

to minimize $f(S_1, S_2, \ldots, S_k)$

Find efficient (approximation) algos for this problem

WHAT IS A METRIC SPACE?

Given n points,

a distance function $d(i,j)$ is a metric if it satisfies

(i) $d(i,j) \geq 0 \quad \forall \; i,j \in [n]$

(ii) $d(i,i) = 0 \quad \forall \; i \in [n]$

(iii) $d(i,j) = d(j,i) \quad \forall i,j \in [n]$

(iv) $d(i,j) \leq d(i,k) + d(k,j)$
$\forall i,j,k \in [n]$.

EXAMPLES OF METRIC SPACES :-

① - n points can be vectors in $\mathbb{R}^d$
- $d(i,j) = \|v_i - v_j\|_2$
or $\|v_i - v_j\|_p$ for any p.
✓ IS A METRIC

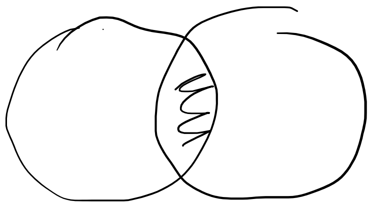
② n points can be vertices of $G=(V,E)$
$d(i,j)$ = shortest path b/w i & j
✓ is a METRIC

③ If G is directed, its not a metric.
[Δ^{le} inequality is fine, but symmetry doesn't hold]

④ Have a collection of documents, each of which contains english words.
Let document i contain S_i (set of words)

then $J(i,j)$ = JACCARD SIMILARITY

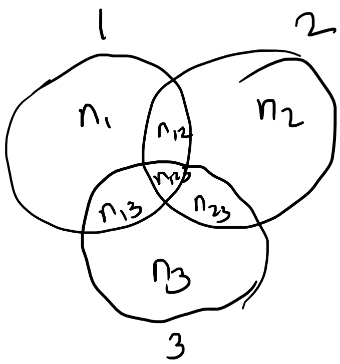
$$= \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$$



then $d(i,j) = 1 - J(i,j)$

is a metric

Is Δ^{le} inequality easy to see?



$$d(1,2) = 1 - \frac{n_{12} + n_{123}}{T - n_3}$$

$$d(1,3) = 1 - \frac{n_{13} + n_{123}}{T - n_2}$$

$$d(2,3) = 1 - \frac{n_{23} + n_{123}}{T - n_1}$$

Need to show

$$d(1,2) \leq d(1,3) + d(2,3)$$

$$1 - \frac{n_{12}+n_{123}}{T-n_3} \leq 1 - \frac{n_{13}+n_{123}}{T-n_2} + 1 - \frac{n_{23}+n_{123}}{T-n_1}$$

$$\frac{n_{13}+n_{123}}{T-n_2} + \frac{n_{23}+n_{123}}{T-n_1} \leq 1 + \frac{n_{12}+n_{123}}{T-n_3}$$

[Homework, Check if true]

Common Objective Functions :-

① k-Center Objective Function

(useful for placing police stations, etc)

First used in the 1950s, One of the earliest uses of approx Algos.

Basically, trying to min $\max_{k'=1}^{k} \max_{i,j \in S_{k'}} d(i,j)$

(ie) Min Max Diameter of each cluster

Useful for Police Stations, Petrol bunks, Hospitals, etc.

FORMAL

Given a cluster S,

Given a cluster S,

def. center of cluster $= \arg\min_{i \in S} \max_{j \in S} d(i,j)$

& radius of cluster $= \min_{i \in S} \max_{j \in S} d(i,j)$

k-Center Problem

Given [n] points, find a clustering of [n] into $S_1, S_2, \dots, S_k$ s.t. we minimize maximum radius over all S_i.

- Prefers all clusters are smallish over a clustering where one is very big & many are very small.

Possible Algorithms :-

(1) Can we use set cover?

Elements = [n].

sets correspond to

$$S_i = \{ j : d(i,j) \le r \}$$

for some 'r'. Then choose a min set cover.

- Keep increasing 'r' & stop when we can find a set cover of size 'k'.

---

(2) Idea: Obj fn. tries to avoid any point which is 'far' from all the k centers.

Greedy like Algo :-

1. Pick C_1 as an arbitrary point from $[n]$.
2. For $t = 2, \dots k$
3. Pick $C_t = \underset{i \in [n]}{\arg\max} \underbrace{\min_{1 \le t' \le t-1} d(i, C_{t'})}$
4. Form the clusters with centers $\{C_1, C_2, \dots C_k\}$ by assigning each point to nearest center.

Idea: In step (3), we are checking if there is any point which hurts the obj. Fn. for the current set of centers.

we pick the 'worst' such point & make it a new cluster center.

THM

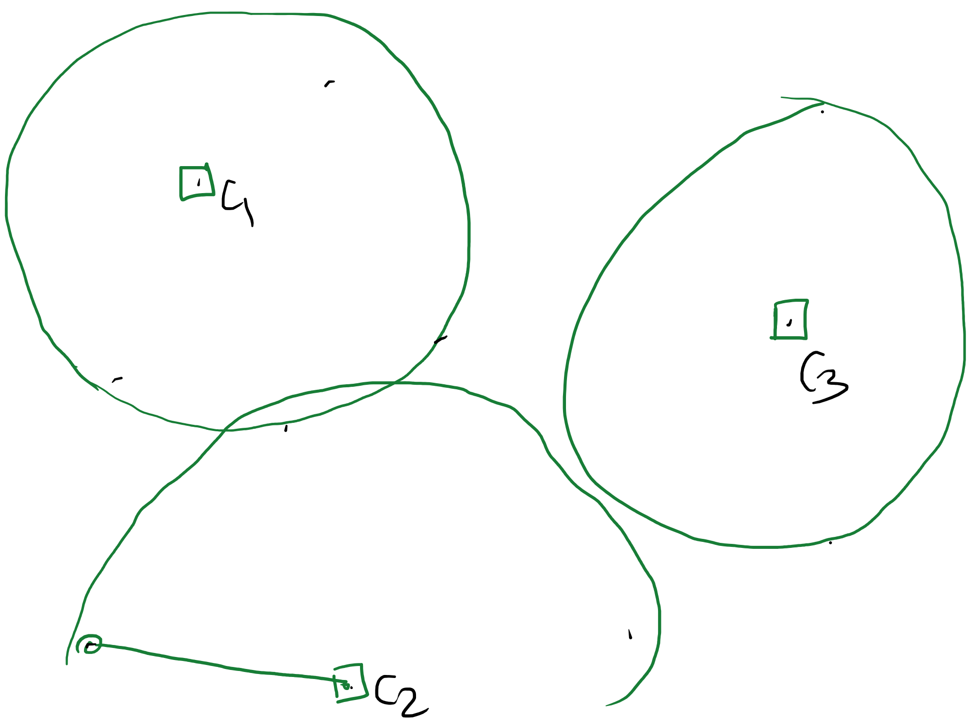
ALGO is 2-approx for k-center problem.

The k-Center Problem

09 March 2021 09:00

Given set P of 'n' points in a metric space, choose a subset of k centers $\mathcal{C} = \{c_1, c_2, \ldots, c_k\} \subseteq P$ such that we can cluster the points in P with minimum radius clusters.

(ie) minimize $\max_{p \in P} \min_{c_i \in \mathcal{C}} d(p, c_i)$.



Once we choose the centers, the clustering to minimize the max-radius is easy - each point is assigned to its nearest center.

Moreover, Max radius of all clusters

$$= \max_{p \in P} \min_{c_i \in \mathcal{C}} d(p, c_i)$$

Let's focus on the distance of each point to its nearest center

Once we fix the k centers

	dist-to-closest-center
p_1	5
p_2	10
⋮	$\boxed{12}$
	2
	0
p_n	7.

Restating the problem:

Choose k centers $\{c_1, c_2, \ldots c_k\} \subseteq P$

to minimize

$$\boxed{\max_{p \in P} \min_{c_i \in \mathcal{C}} d(p, c_i)}$$

- Motivates the greedy algorithm

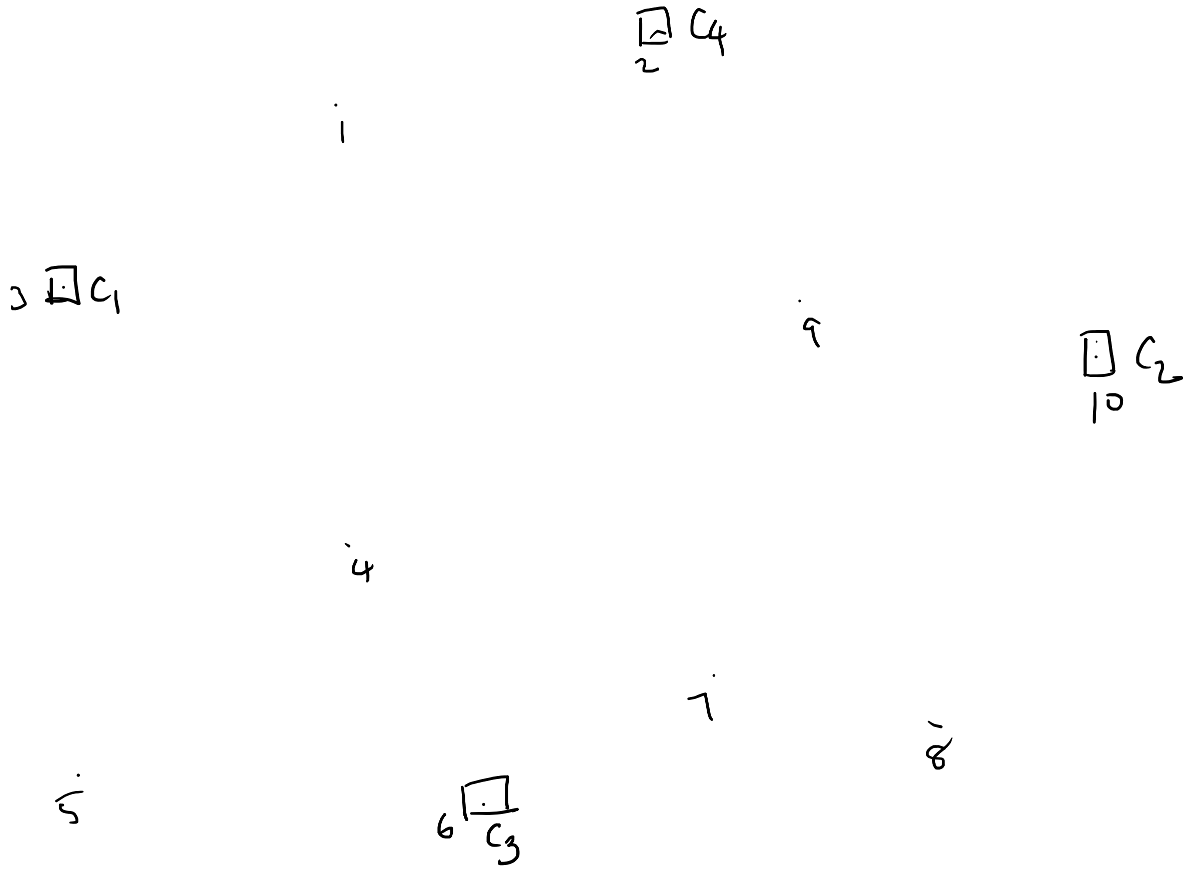
① choose c_1 arbitrarily from P

② for $t = 2, \dots, k$

③ ↳ let $c_t = \underset{p \in P}{\arg\max} \min_{1 \le t' \le t-1} d(p, c_{t'})$.

④ Form the Clustering using $c_1, c_2, \dots c_k$ as centers. ∎

Toy Example



Suppose $k = 4$

THEOREM

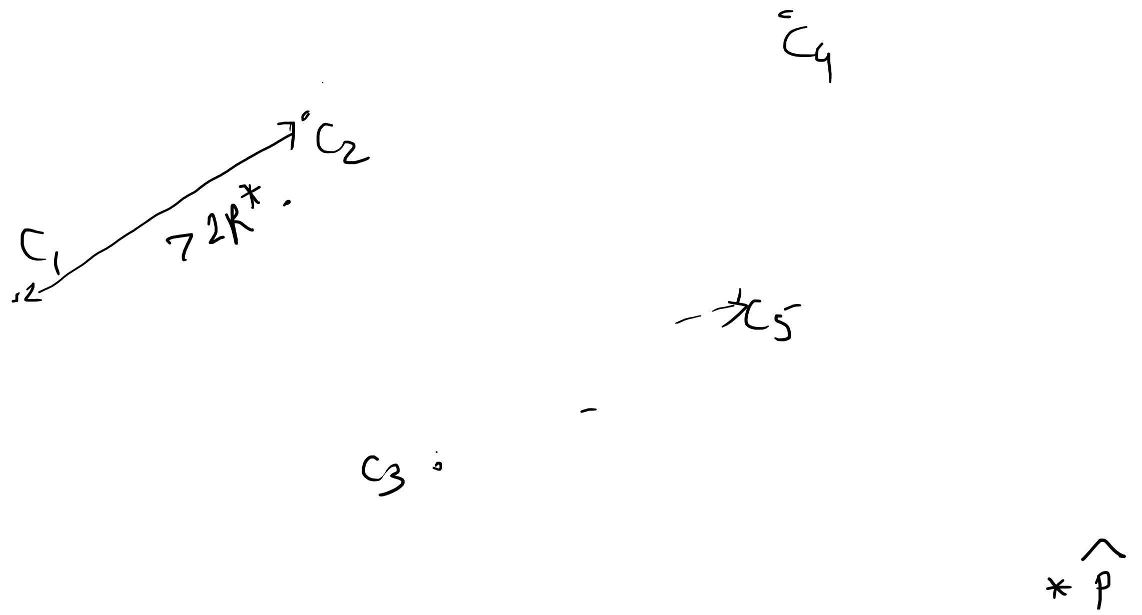
If Optimal clustering has objective value R^*,
Our clustering has objective value $\leq 2R^*$.
($\Rightarrow$ 2-approximation).

Proof :-
Let $c_1, c_2, \ldots c_k$ be algorithms centers.
Let $c_1^*, c_2^*, \ldots, c_k^*$ be optimal centers.

Suppose for contradiction, our clustering has objective value $> 2R^*$.

$\Rightarrow \exists$ a point $\hat{p} \in P$ st it is far from all $c_1, c_2, \ldots c_k$

$\Rightarrow d(\hat{p}, c_i) > 2R^* \quad \forall i = 1, 2, \ldots k$.



$* \hat{p}$

Because greedy algo chose

$$c_t = \underset{p \in P}{\operatorname{argmax}} \min_{1 \le t' \le t-1} d(p, c_{t'})$$

and $\min_{1 \le t' \le t} d(\hat{p}, c_{t'}) > 2R^*$

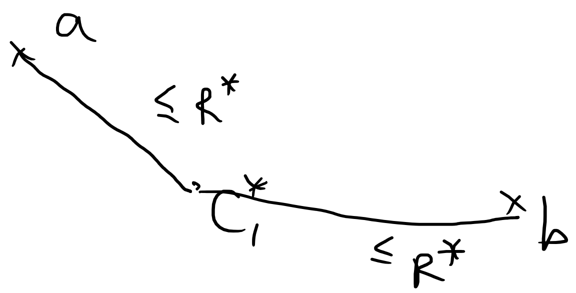
$\Rightarrow$ $\hat{p}$ is a candidate for all of $c_2, \ldots c_k$ but didn't get picked

$\Rightarrow$ $\forall t$, $\min_{1 \le t' \le t-1} d(c_t, c_{t'}) > 2R^*$ as well.

$\Rightarrow$ There exist $k+1$ points in P namely $\{c_1, c_2, \ldots, c_k, \hat{p}\}$ st each pair of points is at $> 2R^*$ distance.

But now how are these points clustered well in OPT?

$\exists$ some cluster, say C_1^* with ≥ 2 points a and b from $\{c_1, c_2, \ldots, c_k, \hat{p}\}$



Because OPT has radius R^*

$$d(a, C_1^*) \leq R^* \quad \text{and}$$
$$d(b, C_1^*) \leq R^*$$

Now, Δ^{le} inequality gives

$$d(a,b) \leq d(a, C_1^*) + d(C_1^*, b)$$
$$\leq 2R^*$$

[CONTRADICTION]

$\boxtimes$

The k-Median Problem

09 March 2021 09:36

Same type of input as k-center, but
a slightly different objective function.

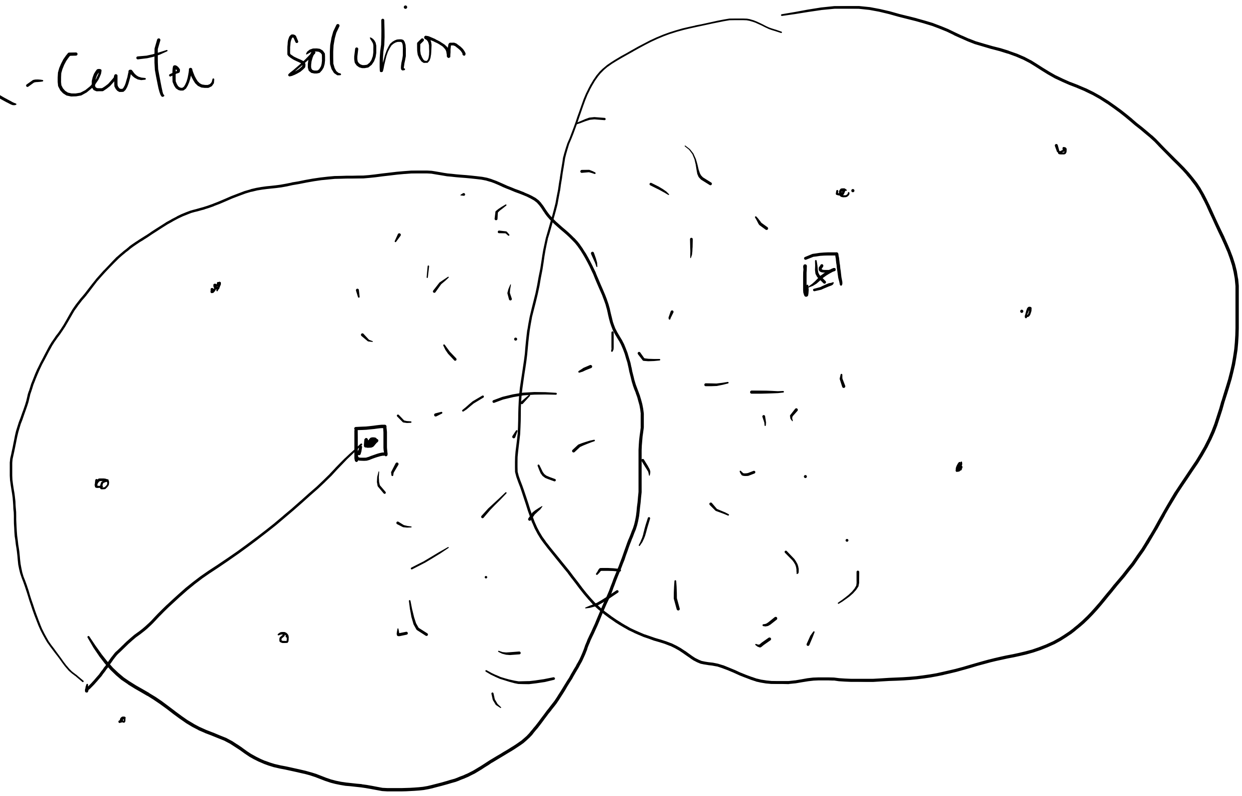
Given n points in a metric space
Find k centers $c_1, c_2, \ldots c_k$
to minimize $\boxed{\sum_{p \in P} \min_{i=1}^{k} d(p, c_i)}$.

Motivation of Objective Function

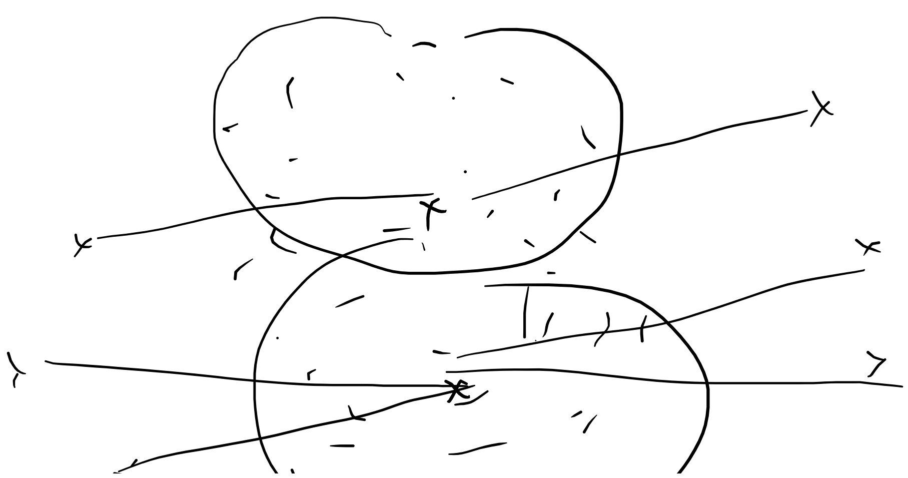
(1) Maybe we are trying to lay <u>cables</u>
from k stations to all
points of the city, where
to place the power stations?

Total Cable length matters more than
max.

k-Center solution



k-Median solution



Try to cover the dense region better while being OK with a few points paying a large cost.

Yet another problem

k-Means Problem.

Same as above, objective is

$$\text{minimize} \sum_{p \in P} \min_{i=1}^{k} d(p, c_i)^2$$

Has lots of applications in ML, unsupervised learning, etc.

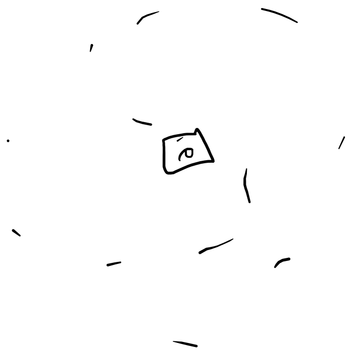
especially when points are vectors in $\mathbb{R}^d$ and

$$d(i, j) = \|v_i - v_j\|_2$$

Has very nice physics connections to concepts such as

Center of gravity, etc.

Point which minimizes sum of squared distances is the 'mean'/centroid of dataset.



k-Means is a natural generalization to k-clusters.

LP-Rounding for k-Median

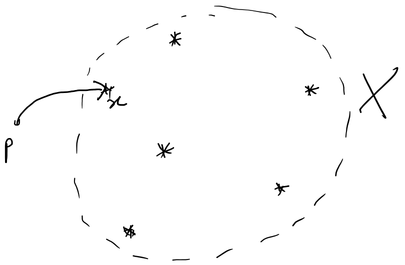
10 March 2021 07:57

Given n points in a metric space P

find k centers $\mathcal{C} = \{c_1, c_2, \dots c_k\}$

to minimize $\sum_{p \in P} d(p, \mathcal{C})$

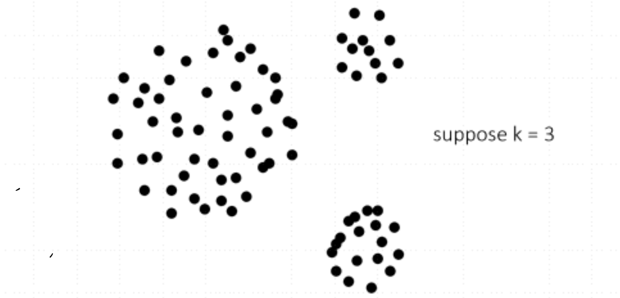
where $\boxed{d(p, X) = \min_{x \in X} d(p, x)}$



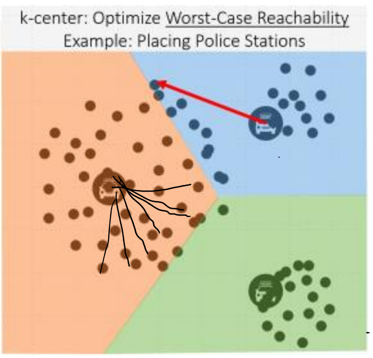
Recall k-Center objective was to

minimize $\max_{p \in P} d(p, \mathcal{C})$

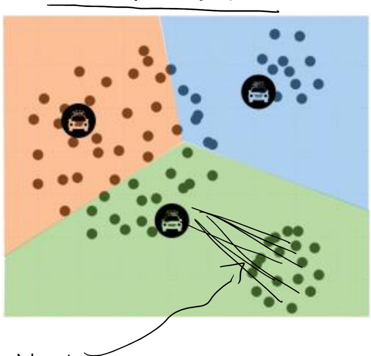
Illustration of k-Median vs k-Center :-



k-Center Solution



k-Median Solution



Maybe its ok for a few points to pay a large cost if many points pay less.

k-Center admits 2-approximation using Greedy algorithm. What about k-Median?

Greedy algorithm. What about k-Median?

Simple greedy-type Algos don't end-up being very good, we can resort to Linear Programming.

Y_i = Variable for whether i is chosen as a center or not.

X_{ij} = variable for whether point j is assigned / clustered to center@i

LINEAR PROGRAM (K-MEDIAN)

$$\text{Min} \sum_j \left(\sum_i d(i,j) \cdot x_{ij} \right)$$

$$\sum_{i=1}^{n} x_{ij} \geq 1 \qquad \forall j \in P$$

$$\sum_{i=1}^{n} y_i \leq k$$

$$x_{ij} \leq y_i \qquad \forall i,j$$

$$x_{ij} \geq 0$$

$$y_i \geq 0$$

Lemma ①

Let (x^*, y^*) be an optimal LP solution

Then $LP^* = \sum_j \sum_i d(i,j)\, x^*_{ij} \leq OPT$

where OPT is the kMedian cost of optimal solution.

Proof

Unknown optimal k-Median solution is feasible for the LP, which can only do better.

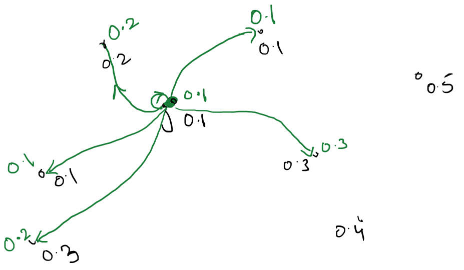
QUESTION

How do we "round" this fractional solution into a good clustering?

Also note :-

In the optimal LP solution, once we know the y^* values, the x^*_{ij} values can be easily derived

Let's look at some point j.



white values are y_i^* values.

green values are x^*_{ij} values

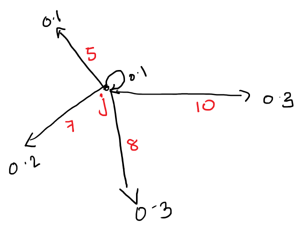
Idea

Infer some basic properties of what the LP optimal is trying to do.

In particular, let

D_j = LP - distance that point j incurs in the optimal sol^n

$$= \sum_i d(i,j) \cdot x^*_{ij}$$



→ Distances are in Red

→ x^*_{ij} values are in white

D_j = LP - distance of point j

$= 0.1*0 + 0.1*5 + 0.2*7 + 0.3*8 + 0.3*10$

$= 0.5 + 1.4 + 2.4 + 3$

$= 7.3$

LP tries to cover point j by connecting it to a center at "distance" 7.3, so we use that as a guide.

LEMMA ②

For any point j, let $B_j = \{ i : d(i,j) \le 2D_j \}$ be the points at distance $\le 2D_j$ from j.

Then $\sum_{i \in B_j} y_i^* \geq \sum_{i \in B_j} x_{ij}^* \geq \frac{1}{2}$.

Proof :-

Suppose not, and $\exists\, j$ st

$$\sum_{i \in B_j} x_{ij}^* < \frac{1}{2}.$$

Then

$$D_j = \sum_i d(i,j)\, x_{ij}^* = \sum_{i \in B_j} x_{ij}^*\, d(i,j) + \sum_{i \notin B_j} x_{ij}^*\, d(i,j)$$

$$\geq 0 + \Big(\sum_{i \notin B_j} x_{ij}^*\Big) \cdot 2D_j$$

$$> 0 + \frac{1}{2} \cdot 2D_j$$

$$= D_j \quad \Rightarrow \Leftarrow .$$

For each j, if we place some center at a point in B_j, then its connection distance $\le 2D_j$

$\Rightarrow$ Overall cost $\le \sum_j 2D_j \le 2\,LP^* \le 2\,OPT$

12/03/2021

Algo ①

① Sort j such that $D_{j_1} \le D_{j_2} \le \cdots \le D_{j_n}$

② Pick the set "near-independent" points J^* as follows

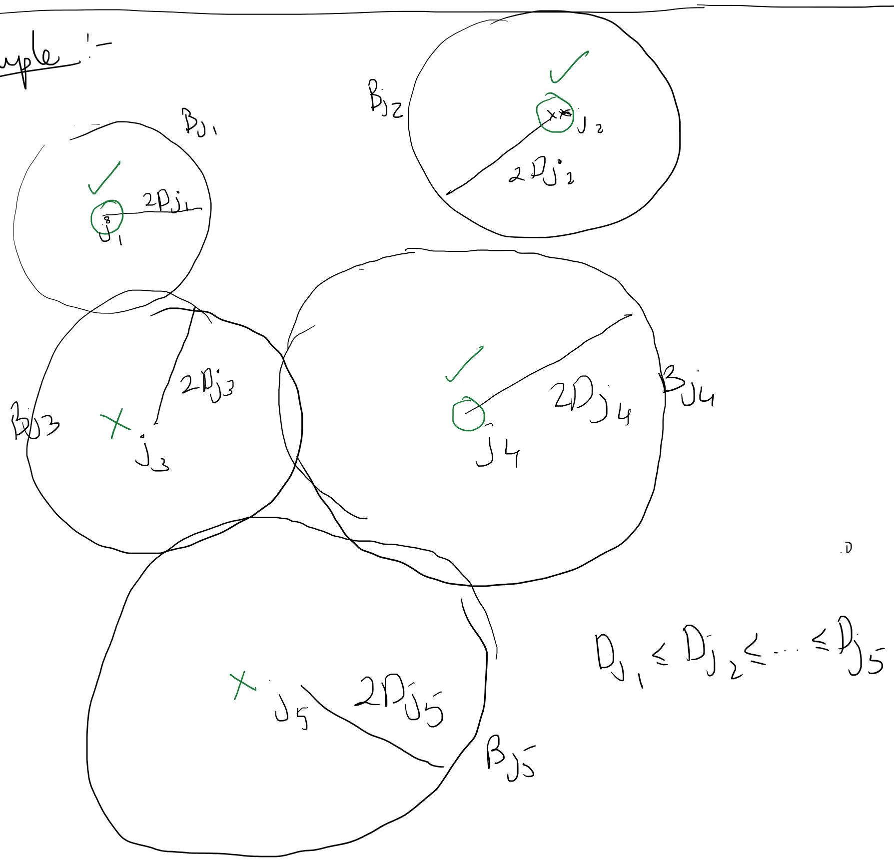
- For $\ell = 1 \ldots n$ (in sorted order)

  If there exists no $j' \in J^*$ s.t $d(j_\ell, j') \le 2D_{j_\ell} + 2D_{j'}$

  then add j_ℓ to J^*.

③ Open a center at each $j \in J^*$

Example :-



$D_{j_1} \le D_{j_2} \le \dots \le D_{j_5}$

Finally, J^* contains j_1, j_2, and j_4.

Lemma ①

J^* is s.t $\forall j', j'' \in J^*$

$$B_{j'} \cap B_{j''} = \phi .$$

Proof

Sps $B_{j'} \cap B_{j''} \neq \phi$ and say i belongs to both.

then $d(j', j'') \le d(j', i) + d(i, j'')$

$\le 2D_{j'} + 2D_{j''}$

(contradiction to whichever got added later)

What did we do?

① Somehow identify points which are

– not too overlapping

(2) Remaining points are "close" to the J^* points ⇒ if we can handle the J^*, we can hope to handle the rest also.

SIMPLE ALGO (1) { Won't exactly give k centers, but might open up to $2k$ centers }.

$\boxed{\text{For each } j \in J^*\text{, open a center at } j.}$

<u>Claim (1)</u>

$\forall j$, distance of j to nearest open center $\leq 4\Delta_j$

<u>Claim (2)</u>

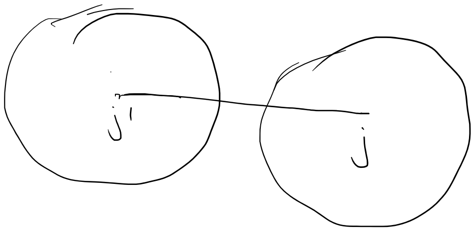
Total # open centers $\leq 2k$.

<u>Proof</u>

Claim (1) :-

If j got added to J^*, then dist of j to nearest center $= 0$
$\leq \Delta_j$ ✓

If j didn't get added,
there must exist a j' st



$$d(j,j') \le 2D_j + 2D_{j'}$$
$$\le 4D_j$$

So dist to nearest open center $\le 4D_j$

Pf of claim 2:

In each B_j, $\sum_{i \in B_j} y_i^* \geq \frac{1}{2}$

Moreover $B_{j_1} \cap B_{j_2} = \phi$ for $j_1, j_2 \in J^*$

So, simply sum over all $j \in J^*$

$$\sum_{j \in J^*} \sum_{i \in B_j} y_i^* \le \sum_{i=1}^{n} y_i^* \le k$$

↑ LP constraint

$$\sum_{j \in J^*} \sum_{i \in B_j} y_i^* \geq \sum_{j \in J^*} \frac{1}{2} = \frac{|J^*|}{2}.$$

$$\Rightarrow \boxed{|J^*| \le 2k}$$

Bi-Criteria Approximation Algorithm

Given an instance of k-Median, with optimal cost $= OPT$, we can efficiently find a solution which opens $2k$ centers and has cost $\le 4\,OPT$

$\leq 4\,OPT$

How do we improve to a pure k-Median solution?

I D E A

Focus on J^*, pair them up in J^*, so that in each pair $\sum y_i^* \geq 1$ and then handle each pair

Recap

1) Solve LP
2) Get (x^*, y^*) as solution
3) Define $D_j = \sum_i d(i,j)\, x^*_{ij}$
4) Define $B_j = \{ i : d(i,j) \leq 2D_j \}$
5) $y^*(B_j) = \sum_{i \in B_j} y^*_i \geq \frac{1}{2} \quad \forall j$
6) J^* is the near-independent far-away points in J.
7) $\forall j \notin J^*$, $\exists j' \in J^*$ st $d(j,j') \leq 4D_j$ and $D_{j'} \leq D_j$
8) $|J^*| \leq 2k$
9) $\forall j, j' \in J^*$, $B_j \cap B_{j'} = \phi$

From $1 \to 8$, we got a 4-approximation
① which opens $2k$ cluster centers

② Today, we'll try to make it a genuine k-clustering which opens $\leq k$ centers while being slightly worse in cost.

Idea
- Pair up points in J^*.
- Open one center in each pair

$|J^*| \leq 2k \implies$ # centers open $\leq k$. ☺

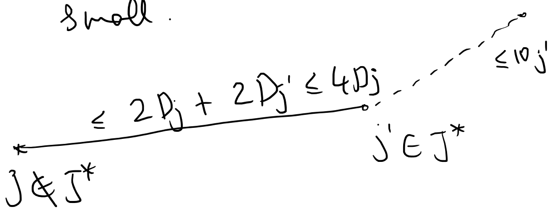
- How do we analyze the cost?
- How to pair points in J^*?
- How to decide which center to open in a pair?

... connection cost ...

a pair ?

We'll try to ensure the connection cost of each point $j \in J^*$ is at most, say, $10 D_j$.

$\Rightarrow$ Connection cost of points not in J^* is also small.



$\Rightarrow$ Conn. cost of $j \le$

$$2D_j + 2D_{j'} + 10D_{j'}$$
$$\le 14D_j$$

$\Rightarrow$ It suffices to show that points in J^* have low connection cost.

Q| How do we choose k centers to ensure that points in J^* have low conn cost ($\propto D_j$)

PAIRING ALGORITHM

- Among J^*, choose the closest pair of points say (j_1, j_2) and match them.
- Remove j_1, j_2 from J^* and repeat

---

pairs + (singleton, If Left) $\le k$.

For now, let's assume no singleton left.

Can handle it easily later.

Can handle it easily later.

We'll come up with a randomized selection procedure which ensures

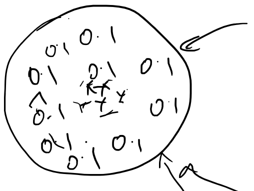
① Each point is chosen as a center with prob y_i^*

② Total # of centers opened $\leq k$

③ For each matched pair (j_1, j_2), we definitely open ≥ 1 center from among the points $B_{j_1} \cup B_{j_2}$.

SATISFYING just ① is easy.
Each i will Independently choose itself as a center w.p y_i^*.

As a sol^n idea, not great because there could be a region in space where we don't open any center



Independent rounding
$\Rightarrow (1-0.1)^{10} \approx 1/e$ prob. not opening any center in this cloud

Very far !!

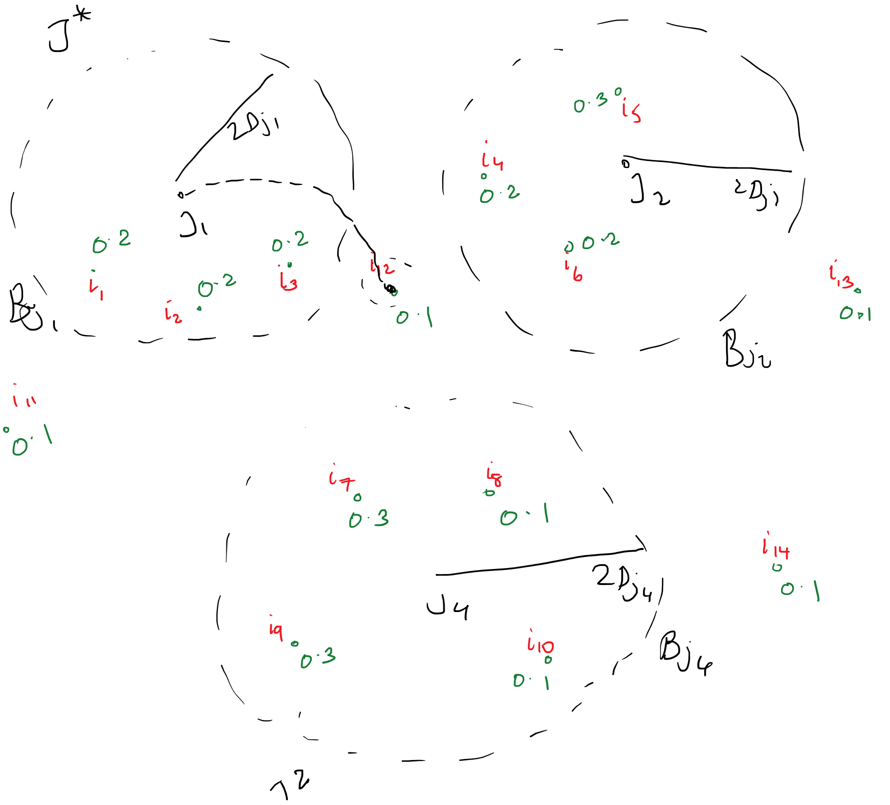
It fails to satisfy ② & ③ also.

In fact ③ precisely tries to address the issue of completely missing local regions in space, which is the drawback of independent rounding.

α - point rounding / dependant rounding
— Nice properties of prob $\propto y_i^*$

along with satisfying extra constraints.

Selection Procedure Illustration

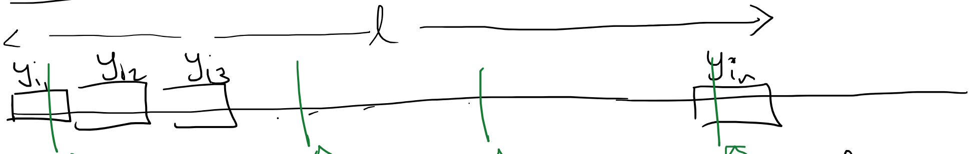


Example in picture above

j_1 connects to $i_1, i_2, i_3, i_{12}, i_{11}, i_4$

$\forall i$, we want to associate y_i length segment on the real line, and put them consecutively

Attempt ①: Order arbitrarily $i_1, i_2, \dots i_n$



and place segments consecutively.

Pick random $\alpha \in [0,1]$ uniformly and

Mark $\alpha, 1+\alpha, 2+\alpha, \dots k-1+\alpha$

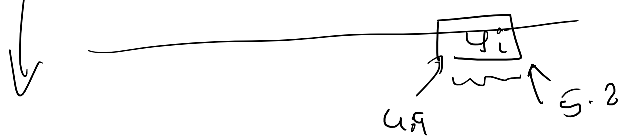
Because $\sum y_i = k$, length of line segment $l = k$.

If a dart intersects a segment y_i, open a center at i.

Nice properties:-

$\forall i$, $\Pr[i$ is opened as center$]$

First dart crosses i, only if $0 \le \alpha \le y_i$, which has probability y_i



This is true for any i,

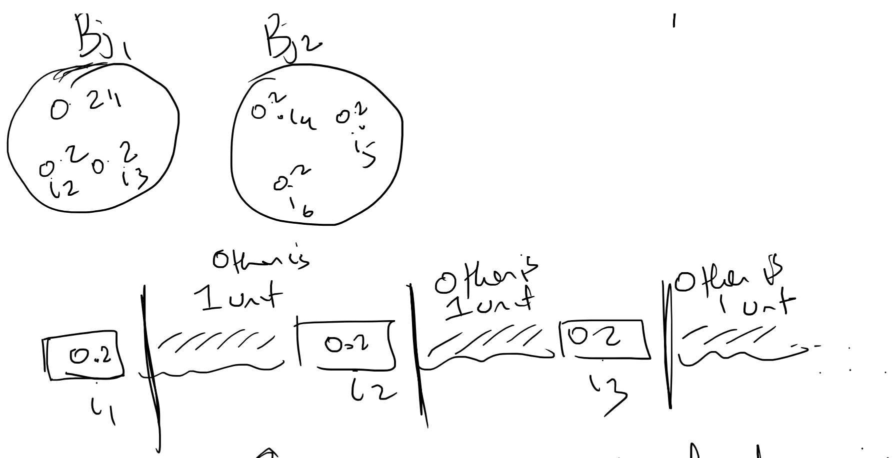
$\Pr[i$ is chosen as center$] = y_i$

(2) ✓ # centers $= k$
because $(k+1)^{th}$ dart will lie outside the system.

(3) ✗ doesn't satisfy the property that in a "local region", there is one open center

(e.g) $\forall (j, j') \in M$, we may not open any center in $B_{j_1} \cup B_{j_2}$

B_{i_1} B_{i_2}



if $\alpha > 0.2$, all darts miss B_{j_1} and B_{j_2}

So, need to preserve locality in some manner.

Goals:

1. Ensure B_j is contiguous for each j
2. Ensure B_{j_1} and B_{j_2} are contiguous for each $(j_1, j_2) \in M$
3. Ensure other close points to j are contiguous for each j

for each i not in any B_j for $j \in J^*$, ensure it is contiguous with its nearest $j \in J^*$

$$O_j = \{ i : d(i,j) \leq d(i,j') \ \forall j' \in J^*, j' \neq j \}$$

No ties in distances

lets assume that we break ties arbitrarily.

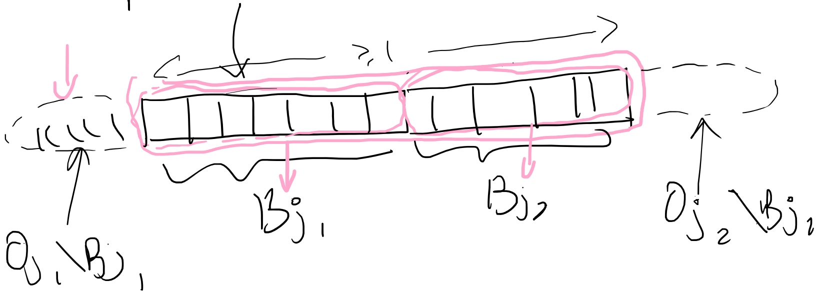
No ties in distances } we break ties arbitrarily.

(ii) $d_1 \neq d_2$ for all pairs

(3) $\forall j$, want O_j appears consecutively

Will this work?

Take pair $(j_1, j_2) \in M$



repeat for all pairs

Now since $y(B_{j_1}) + y(B_{j_2}) \geq 1$

we will definitely open one center in each "pair".

with same dart throwing algorithm.

17/03/2021

Algorithm

- Recall defn of B_j, O_j, J^*, D_j, x^*, y^*
- Recall Pairing M.
- Place the points on a line and do the "α-point rounding"

Ensure that for each pair,

B_{j_1} is contiguous

B_{j_2} is contiguous

$O_{j_1} \setminus B_{j_1}$ contiguous with B_{j_1}

$O_{j_2} \setminus B_{j_2}$ contiguous with B_{j_2}

- Open all centers which are crossed by the darts α.

Lemma 1

Each i is opened with prob y_i^*

Lemma 2

For any pair $(j, j') \in M$, at least one point is opened from $B_{j_1} \cup B_{j_2}$ with probability 1.

$\longleftarrow$ length ≥ 1 $\longrightarrow$

$\underbrace{\qquad\qquad}_{B_{j_1}} \underbrace{\qquad\qquad}_{B_1}$

because $y^*(B_{j_1}) + y^*(B_{j_2}) \geq 1$

and $B_{j_1} \cap B_{j_2} = \phi$

✓ gap b/w any 2 darts $= 1$

ENSURES "LOCALITY PRESERVING ROUNDING".

Lemma 3

For each point $j \in J^*$, let

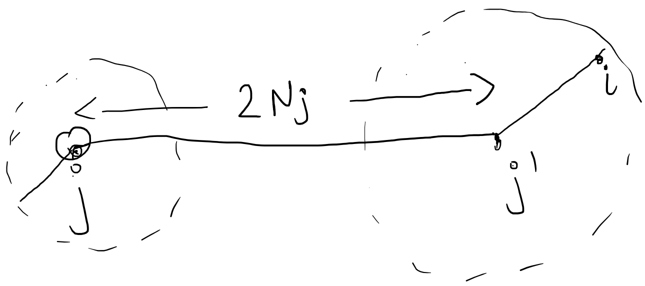
$N_j = \frac{1}{2} \min_{j' \in J^*} d(j, j')$ be half

$N_j = \frac{1}{2} \min_{\substack{j' \in J^* \\ j' \neq j}} d(j, j')$ be half the dist. to nearest other pt in J^*

Then, there is always an open center within $6 N_j$

Proof

Let's focus on j and j' – its nearest other pt from J^*



Either $(j, j') \in M$ or not

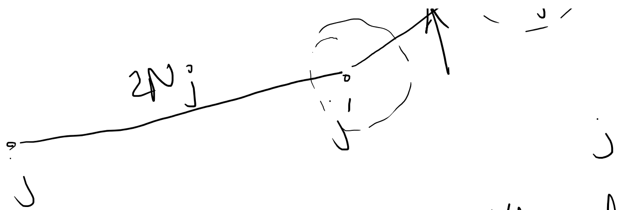
Case ①: If $(j, j') \in M$.

There is open center i among $B_{j_1} \cup B_{j_2}$

$\Rightarrow d(j, i) \leq 4 N_j$

Case ② $(j, j') \notin M$.

$\Rightarrow \exists\, j''$ st $d(j', j'') \leq 2N_j$ and $(j', j'') \in M$.



But again, there will always be an open center in $B_{j'} \cup B_{j''}$

$$d(j, i) \le 2N_j + D + D$$
$$\le 6N_j \quad (\Delta^{le} \text{ inequality}).$$

Rest of Analysis :-

If for some j, $D_j \ge 0.1 N_j$ then ← LP connection cost.

such j's are very happy

(Their real connection cost ≤ 60 their LP connection cost)

$\boxed{\text{Real problem happens if } D_j \text{ is much smaller than } N_j}$

Let's analyze D_j :-

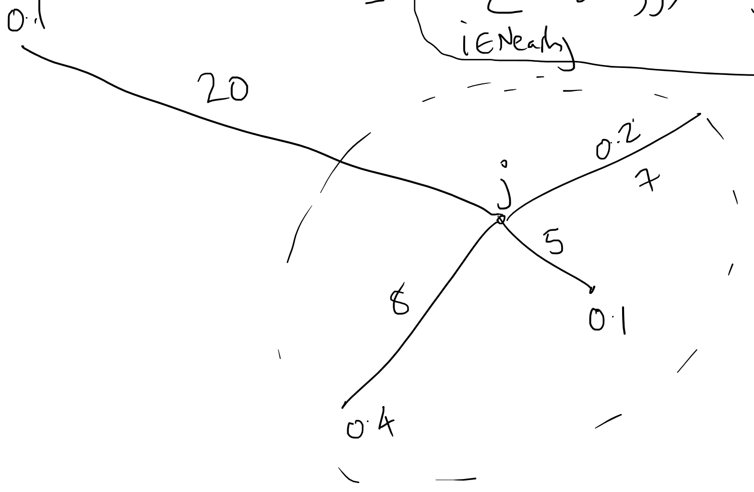
$$D_j = \sum_i d(i, j)\, x^*_{ij}$$

$$= \sum_{i : d(i,j) < N_j} d(i, j)\, x^*_{ij} + \sum_{i : d(i,j) \ge N_j} d(i, j)\, x^*_{ij}$$

$i : d(i,j) < N_j$ $\qquad$ $i : d(i,j) \geq N_j$

$$\boxed{D_j \geq \sum_{\substack{i : d(i,j) \\ < N_j}} d(i,j)\, x^*_{ij} + N_j \sum_{\substack{i : d(i,j) \\ \geq N_j}} x^*_{ij}}$$

$$= \sum_{i \in \text{Nearby}} d(i,j) \cdot x^*_{ij} + N_j \left(1 - \sum_{i \in \text{Nearby}} x^*_{ij}\right)$$

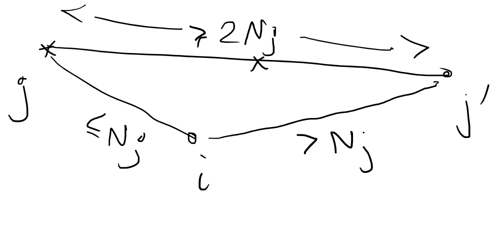


$5 * 0.1 + 7 * 0.2$
$+ 8 * 0.4$

What's the expected connection cost of this point j, in our rounding scheme?

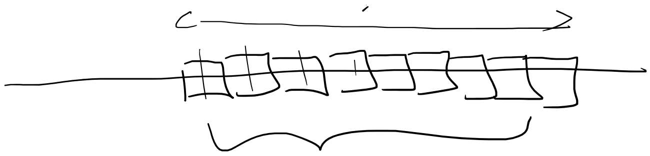
Notice: all the points i s.t. $d(i,j) < N_j$ belong to O_j

for any other j'



Now, we ensured that such points can be placed contiguously in the line.

All points $i : \{ d(i,j) < N_j \}$ are placed contiguously.



$i : d(i,j) < N_j \Leftarrow$ Nearby Points

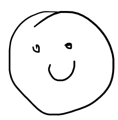
Expected cost of j's connection

$=$ Pr(One of the nearby pts chosen) $\cdot$
E[distance | nearby points chosen]

$+$ Pr(Nearby pt not chosen)
E[distance | Nearby pt not chosen]

$$= \left(\sum_{i : d(i,j) < N_j} y_i^*\right) \left[\frac{\sum_{i : d_{ij} < N_j} d_{ij}\, y_i^*}{\sum_{i : d(i,j) < N_j} y_i^*} \right] + \left(1 - \sum_{i : d_{ij} < N_j} y_i^*\right) \cdot 6 N_j$$

$\leq 6 D_j$

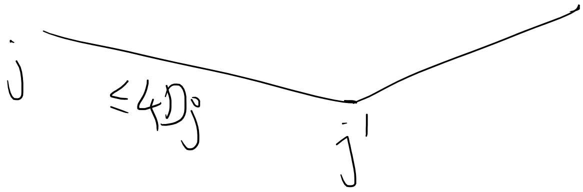


Corollary

$\forall j \in J^*$, E[connection cost of j] $\leq 6 D_j$

Corollary

$\forall j \notin J^*$, E[Conn. cost of j] $\leq 10 D_j$



$\exists\ j'$ st $d(j, j') \le 4D_j$ &

$D_{j'} \le D_j$

$\Rightarrow E[\text{conn. cost of } j] \le 4D_j + E[\text{conn cost of } j']$

$\le 4D_j + 6D_{j'}$

$\le 10 D_j$

Takeaway

1. $O(1)$ for k-Median
2. Respecting hard constraints $\sum y_i \le k$ during rounding is hard, we need "dependant rounding" ideas
3. α-point rounding is a good way to ensure this.

Next 2-3 lectures, yet another algo. for k-Median which gets rid of 2 in a clever way.

k-Median via Lagrangean Relaxation

19 March 2021 12:07

Given metric (X, d) $|X| = n$ points

$d(\cdot)$ is distance function (metric)

- $d(i,j) + d(j,k) \geq d(i,k)$ $\forall i,j,k \in [n]$
- $d(i,j) = d(j,i)$
- $d(i,i) = 0$

Choose k points as centers and assign each point in X to nearest center to minimize total "assignment distance"

(ie) $\sum_{j=1}^{n} d(j,S)$ where $d(j,S) = \min_{i \in S} d(i,j)$

and $|S| = k$

Recall LP

Total Assignment distance $\leftarrow$ $\text{Min} \sum_j \sum_i d(i,j) x_{ij}$

every point is assigned $\leftarrow$ $\sum_i x_{ij} \geq 1$ $\forall j$

Center must be open $\leftarrow$ $x_{ij} \leq y_i$ $\forall i,j$

$\sum y_i \leq k$

$$\sum y_i = k$$

Total k centers

$$\left.\begin{array}{l} x_{ij} \geq 0 \\ y_i \geq 0 \end{array}\right\}$$

Method of Lagrangean Relaxation

<u>Idea</u>:

We had a lot of difficulty in rounding the LP solution to preserve $\sum y_i \leq k$ (allowing violations was much easier)

Instead, let us push this constraint to the objective function

---

<u>New LP</u>: LP2

$$\min_{x,y} \sum_j \sum_i d(i,j)\, x_{ij} + \lambda\left(\underbrace{\sum y_i}\, - k\right)$$

$$\text{LP2} \left\{\begin{array}{ll} \sum_i x_{ij} \geq 1 & \forall j \\ x_{ij} \leq y_i & \forall i,j \\ \left.\begin{array}{l} x_{ij} \geq 0 \\ y_i \geq 0 \end{array}\right\} & \forall i,j \end{array}\right.$$

$\forall \lambda \geq 0,$

$$OPT(LP2) \leq OPT(LP).$$

In LP2, we can effectively ignore the $-\lambda k$ constant in the obj (It is the same for all x, y solutions).

$\boxed{LP3}$ is LP2 without $-\lambda k$ term

LP3

$$\text{Min } \sum_j \sum_i d(i,j) x_{ij} + \lambda \sum y_i$$

$$\sum_i x_{ij} \geq 1 \quad \forall j$$

$$x_{ij} \leq y_i \quad \forall i,j$$

$$x_{ij} \geq 0$$

$$y_i \geq 0$$

$$\boxed{OPT(LP3) \leq OPT(LP) + \lambda k}$$

Called the FACILITY LOCATION PROBLEM.

Open a set S of centers and

assign points to nearest center,

but instead of asking $|S| \le k$, we add a cost of λ for each open center.

obj. fn = $\min \sum_j d(j,S) + \lambda|S|$.

HOPE

① To find an "easy" Apx algo for facility location

② Maybe for suitable λ, the algo actually opens k centers

$\Downarrow$

$\boxed{Q}$ Do ① & ② $\Rightarrow$ Apx Algo is good for k-Median?

$\boxed{Ans}$ This is almost true, need a little more guarantee from the approx algo for Facility Locations.

FACILITY LOCATION OBJ. FN

$\swarrow$ CONN. COST (DISTANCE)

$\searrow$ FACILITY OPENING COST (λ)

2^{nd} cost is what came from the ...

k-Median constraint

Algo A is a C-LAGRANGEAN approximation for FACILITY LOCATION if

CONN. COST(A) + C · FACILITY COST(A)

$\leq$ C · [CONN(OPT) + FAC(OPT)]

without the C in the LHS, it is a traditional C-approximation.

[Recall: these are minimization problems, $C \geq 1$].

WHY does ① & ② along with C-Lagrangean Algo $\Rightarrow$ good algo for k-Median?

CONNCOST(A) + C · λ · k $\leq$ C [CONN COST(k-Median OPT) + λ k]

Plugged in the k-Median OPT as a feasible soln for Facility Location OPT.

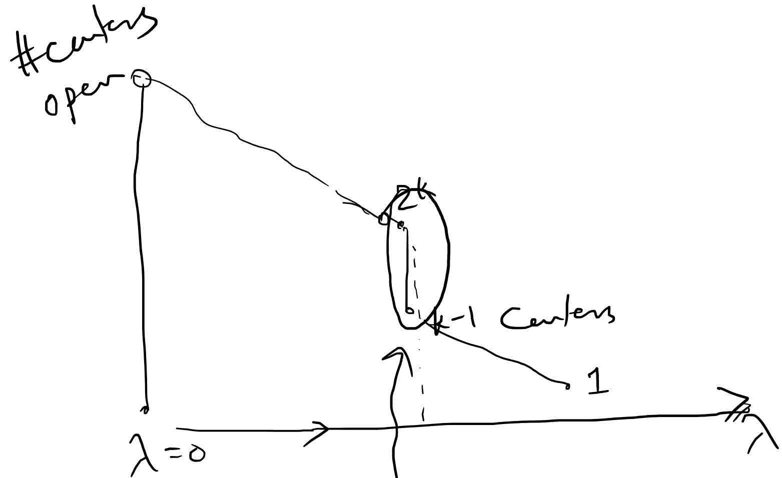
for facility location OPT.

$$\boxed{\text{CONN Cost}(A) \le c \cdot \text{CONN Cost}(k\text{Median OPT}) \text{ and } A \text{ opens } k \text{ centers (from ②)}}$$

$\Downarrow$

For the suitable λ, A is a c-approx. for k-Median

$\Downarrow$



What could happen in reality is that

as we keep increasing λ, there need not exist any point λ where Algo opens exactly k centers.

(i.e) For $\lambda - d\lambda$, it opens $\geq k$ centers and $\lambda + d\lambda$ it opens $< k$ centers

Could very well happen but there's a very nice way to deal with it

Remains to do

Next 1-2 lectures

1. Design C-LAGRANGEAN algo for Facility Location
2. Design the "Combiner Procedure" for $\lambda - d\lambda$ and $\lambda + d\lambda$ problem.

26-03

Any qns about 1 ?

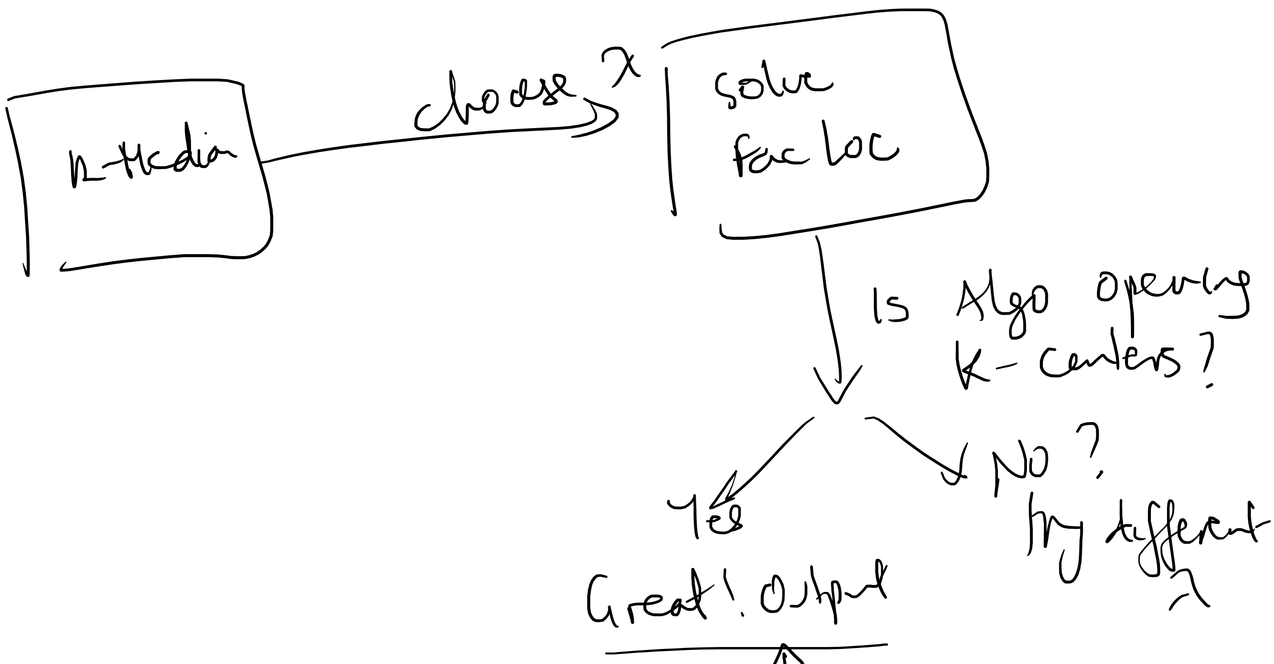
We'll assume black-box access to algorithm which has

CONN-Cost + 3 FAC.COST $\leq$ 3 OPT(FAC.LOC).

How do we use this for k-Median approximation?

Idea

Given a k-Median problem, lets choose an "ideal λ" and solve the Facility location instance.



Why is this sol^n good for k-Median?

Here is where we use the 3-Lagrangean-Approx

If $\exists\ \lambda$ where Algo opens k centers,

then

$$\text{CONN Cost}(\text{Alg}) + 3\lambda k \leq 3(\text{OPT}(\text{FL}))$$

$$\leq 3\left(\text{CONNCost}(\text{k-Median OPT}) + \lambda k\right)$$

$\Rightarrow$

$$\text{CONN Cost}(\text{Alg}) + \cancel{3\lambda k} \le 3\,\text{CONN Cost}(k\text{-Median OPT}) + \cancel{3\lambda k}$$

$$\Rightarrow \quad \text{Conn Cost}(\text{Alg}) \le 3\,\text{Conn Cost}(k\text{-Median OPT})$$

$\boxtimes$

Great, but maybe no λ is "good" where we open exactly k centers

As we keep increasing λ, Algo opens fewer centers.

Maybe discrete jump occurs

$k_1 > k$

$k_2 < k$.

$\lambda_- < \lambda < \lambda_+$

$\lambda_- \approx \lambda \approx \lambda_+$ but algo jumps from opening k_1 centers $(> k)$

Think of

to k_2 centers $(< k)$.

$\lambda_- = \lambda_+ = \lambda$, say.
(limiting case)

$(k_1 \text{ sol}^n) \rightarrow \text{Sol}^n = S_1, \quad \text{Cost} = C_1$

$(k_2 \text{ sol}^n) \rightarrow \text{Sol}^n = S_2, \quad \text{Cost} = C_2.$

$$C_1 + 3\lambda k_1 \leq 3(\text{OPT} + \lambda k) \quad \text{— } (1)$$

$$C_2 + 3\lambda k_2 \leq 3(\text{OPT} + \lambda k). \quad \text{— } (2)$$

COMBINER Procedure

k_1 sol^n is cheap but infeasible
k_2 sol^n is feasible but expensive
{ but, their average is feasible & cheap }

Let $p = \dfrac{k - k_2}{k_1 - k_2}$.

Consider

$p\,(1) + (1-p)\,(2)$, and see what it gives?

$p C_1 + (1-p) C_2 + 3\lambda [\, p k_1 + (1-p) k_2 \,]$

$$pC_1 + (1-p)C_2 + 3\lambda\left[\,pk_1 + (1-p)k_2\,\right]$$
$$\leq 3\,(\mathrm{OPT} + \lambda k).$$

$$\underbrace{pC_1 + (1-p)C_2}\, + 3\lambda\left[\frac{kk_1 - k_1k_2 + k_1k_2 - kk_2}{k_1 - k_2}\right]$$

$$\leq 3(\mathrm{OPT} + \lambda k)$$

$$pC_1 + (1-p)C_2 + 3\cancel{\lambda k} \leq 3\mathrm{OPT} + 3\cancel{\lambda k}$$

So, if we choose $p = \dfrac{k - k_2}{k_1 - k_2}$

(*) so that

$$p \cdot k_1 + (1-p)k_2 = k \quad [\text{wtd Avg is } k]$$

Then Corr. wt Avg of cost

$$pC_1 + (1-p)C_2 \leq 3 \cdot \mathrm{OPT}$$

In argument above

we used $\lambda_- = \lambda - d\lambda$ for

(1) and $\lambda_+ = \lambda + d\lambda$ for (2)

but think of $\lambda_- = \lambda_+$ since
$d\lambda \to 0$.

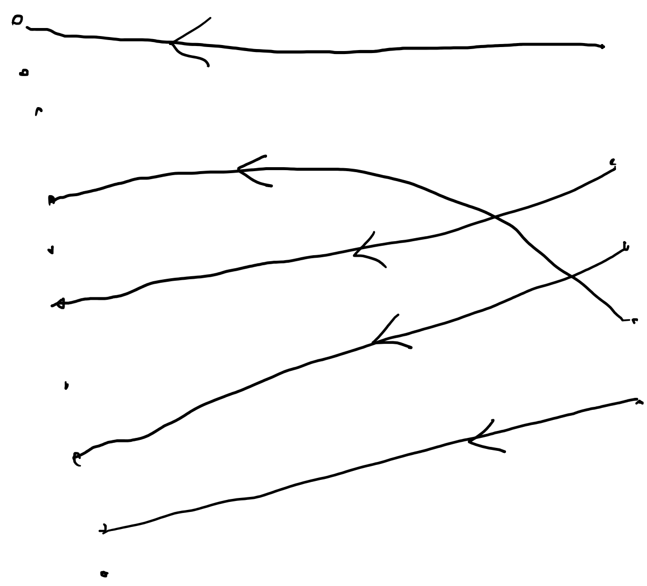
A Randomized Combiner Process :-

Idea :
Open centers in S_1 wp 'p' and
centers in S_2 wp '(1-p)' but
ensure that we open only
k centers wp 1.

If we open S_1 wp p and
S_2 wp (1-p), then we are
infeasible wp p since
$|S_1| > k$.

We 'identify good back-ups' from
S_1 which we open instead
of all of S_1.

$|S_1| > k$



$|S_2| < k$

For each $i \in S_2$, let $\eta(i) \in S_1$ denote the closest point in S_1 to i.

(i.e) $\eta(i) = \underset{i' \in S_1}{\text{argmin}}\ d(i, i')$

Algo

Lets assume $\eta(i_1) \neq \eta(i_2)\ \forall\ i_1, i_2 \in S_1$

$\downarrow$

Proof works even if they collide, this is just to simplify the discussion.

COMBINING PROCEDURE :-

- With probability $(1-p)$, Open all of S_2
- Else, with prob p, we open $D(S_2)$.

Step 1

Only open all mates

After step 1, we open k_2 facilities with prob 1.

From rest of S_1 ($k_1 - k_2$ points)
Choose $k - k_2$ uniformly at random and open them as centers.

⇒ After step 2, we open k centers w.p 1.

Sanity Check

Q: for $i \in S_2$, what is prob of i being selected?

Ans: $(1-p)$

Q: for $i \in S_1$, what is prob of i being selected?

Ans: p

Proof: if $i \in \eta(S_2)$ then it is p.

if $i \notin \eta(S_2)$ then

$$\Pr[i \text{ open}] = \frac{k - k_2}{k_1 - k_2} = p.$$

LEMMA

Expected conn. cost of any pt j

$$\leq 2\left[p \cdot d_1(j) + (1-p)\, d_2(j)\right]$$

where d_1 & d_2 are j's cost in S_1 and S_2.

Corollary

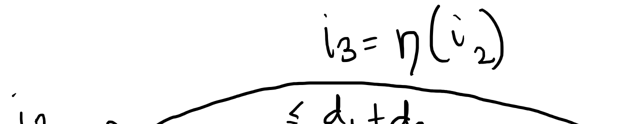
Expected conn. cost of sol^n $\leq 2\left[p \cdot C_1 + (1-p) C_2\right]$

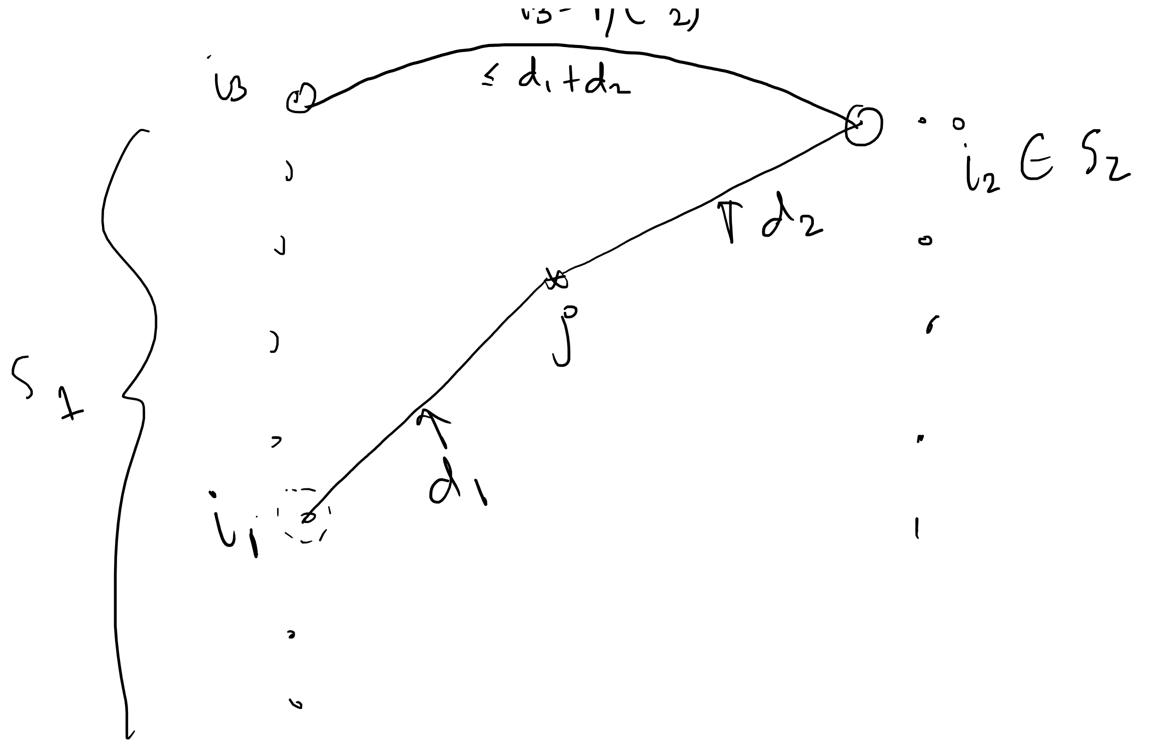
$\leq 2 \cdot 3 \cdot OPT$

$= 6\, OPT$

$\because$ $C_1 = \sum_j d_1(j)$, $C_2 = \sum_j d_2(j)$.

Proof of Lemma





Consider j and suppose its preferred conn. in S_1 is i_1, and S_2 is i_2.

and sps $i_3 = \eta(i_2)$ is the mate.

$$\boxed{d(i_2, i_3) \leq d_1 + d_2}$$

What does j connect to in our "mixed sol^n"?

If i_1 is chosen, j can connect to it. (happens with probability p)

If i_1 is not opened, we can check if i_2 is open.

"Check if i_2 is open"

If i_2 is not open, j can connect to i_3

for this "worst case analysis", lets assume that i_1 is not anybody's mate.

$i_1 \notin \eta(S_2)$.

$E[\text{Conn cost of } j]$

$$= \underset{\substack{\uparrow \\ \text{if } i_1 \text{ is open}}}{p d_1} + (1-p) \cdot \left[(1-p) \cdot \underset{\substack{\uparrow \\ \text{Open all of } S_2}}{d_2} + \underset{\substack{\uparrow \\ \text{Open all of } \eta(S_2)}}{p(d_1 + 2d_2)} \right]$$

$$= p d_1 + (1-p) \left[d_2 + p d_1 + p d_2 \right]$$

$$= p d_1 + (1-p) d_2 + p(1-p)(d_1 + d_2)$$

$$\leq 2 p d_1 + 2(1-p) d_2$$

Next Week

Johnson-Lindenstrauss Lemma.

Given n points X, distance function d and opening cost of $\lambda \geq 0$, choose a set S of centers to open and connect each point to nearest open center (incurring a cost of

$$d(j,S) = \min_{i \in S} d(i,j) \;)$$

to minimize

$$\lambda |S| + \sum_j d(j,S)$$

PRIMAL DUAL 3-(Lagrangean) Approximation.

If OPT sol^n has facility opening cost O^* and connection cost C^*,

and our solution has connection cost $\hat{C}$, and facility opening cost $\hat{O} = \lambda |\hat{S}|$ if we open centers at $\hat{S}$

α-approximation requires that

$$\hat{C} + \hat{O} \leq \alpha (C^* + O^*).$$

α-Lagrangean Approximation requires that

$$\boxed{\hat{C} + \alpha \hat{O} \leq \alpha (C^* + O^*)}$$

Useful to move from FL to k-Median by choosing λ parameter carefully.

<u>LP (FL)</u>

$$\text{Min} \sum_j \sum_i d(i,j) x_{ij} + \lambda \sum_i y_i$$

$\alpha_j)$ $\quad \sum_i x_{ij} \geq 1 \quad \forall j$

$\beta_{ij})$ $\quad y_i - x_{ij} \geq 0 \quad \forall (i,j)$

$\quad x_{ij}, y_i \geq 0$

<u>Dual (FL)</u>

$$\text{Max} \sum_j 1 \alpha_j + \sum_{ij} 0 \beta_{ij}$$

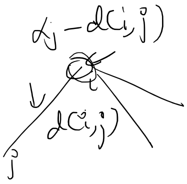
$\alpha_j - \beta_{ij} \leq d(i,j)$

$\sum_j \beta_{ij} \leq \lambda$

$\alpha_i, \beta_{ij} \geq 0$

Recall: "take ≥ 0 linear combination, to maximize RHS, while ensuring all coefficients of primal variables are dominated by obj fn".

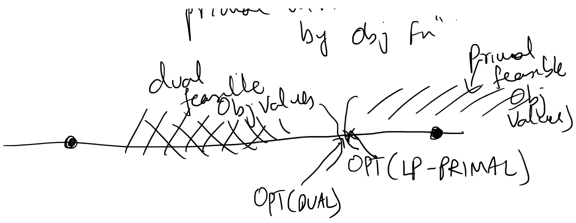
dual feasible values ... Primal feasible Obj



Think of α_j as amount of money j is raising for being connected.

β_{ij} = Amt of Money j is willing to put to opening a center at location i

$$\alpha_j = \beta_{ij} + \underline{d(i,j)}$$



Weak Duality THM

if (α, β) is dual feasible
and (x, y) is primal feasible,

then
$\text{Dual Cost}(\alpha,\beta) \le \text{Primal cost}(x,y)$.

(ie) $\sum_j \alpha_j \le \sum_i \sum_j d(i,j)\, x_{ij} + \lambda \sum_i y_i$

In particular,
(α,β) dual feasible $\Rightarrow$ $\boxed{\sum \alpha_j \le OPT}$ (*)

$\Rightarrow$
if we find some good solution
with cost $\le 3 \cdot \sum \alpha_j$, then
it will be a 3-approximation
due to (*)

PRIMAL - DUAL ALGORITHM ; STEP 1

Initialize $\hat{T} = \phi$ (no open facility)
Initialize $\alpha, \beta = 0$, and all clients are "unfrozen".

While ($\exists$ unfrozen clients)
- Increase $\alpha_j = \alpha_j + \varepsilon$ for suitably small ε for all unfrozen clients.
- If some $\alpha_j - \beta_{ij} = d(i,j)$ is tight for some $i \notin \hat{T}$, then also increase $\beta_{ij} = \beta_{ij} + \varepsilon$ to ensure ① remains feasible
- if some facility constraint ② becomes tight, (ie), $\sum_j \beta_{ij} = \lambda$, then add i to $\hat{T}$ (ie) open i temporarily) and freeze all clients j for which $\underbrace{\alpha_j - \beta_{ij} = d(i,j)}$ is tight.

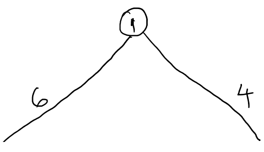
Recall Dual

Max $\sum_j \alpha_j$

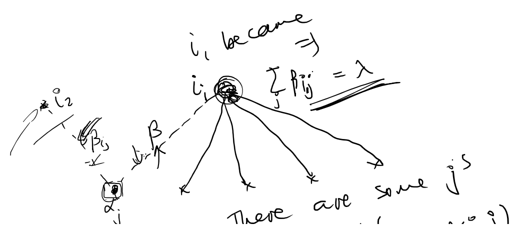
① $\alpha_j - \beta_{ij} \le d(i,j)$ $\forall i,j$
② $\sum_j \beta_{ij} \le \lambda$ $\forall i$
$\alpha_i, \beta_{ij} \ge 0$

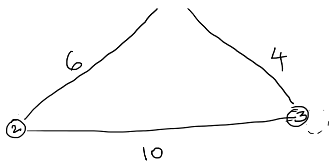
tight

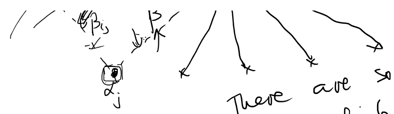
Toy Example
3 points, distances marked in figure



Suppose $\lambda = 12$







There are some j's for which $\alpha_j - \beta_{i,j} = d(i,j)$

None of these j's can $\uparrow \alpha_j$ (all frozen)

$\hat{I} = \phi$, all 3 βs are unfrozen.
all increase their α slowly @ same rate

At first step itself, $\alpha_i - \beta_{ii} = d(i,i)$ is tight because $d(i,i) = 0$

$\Downarrow$

$\beta_{i,i}$ increases jointly with α_i

Max $\Sigma \alpha_i$

$\alpha_1 - \beta_{11} \le 0 \leftarrow$

$\alpha_1 - \beta_{21} \le 6$

$\alpha_1 - \beta_{31} \le 4 \leftarrow$ tight @ $t=4$

$\alpha_2 - \beta_{12} \le 6$

tight @ $t=6$

$\alpha_2 - \beta_{22} \le 0 \leftarrow$

$\alpha_2 - \beta_{32} \le 10$

$\alpha_3 - \beta_{13} \le 4 \leftarrow$ tight @ $t=4$

$\alpha_3 - \beta_{23} \le 10$

$\alpha_3 - \beta_{33} \le 0 \leftarrow$

$\beta_{11} + \beta_{12} + \beta_{13} \le 12 \leftarrow$

$\beta_{31} + \beta_{32} + \beta_{33} \le 12$

Right at first step, β_{ii} grows like α_i

@ $t=7$, $\beta_{11} = 7$, $\beta_{12} = 1$, $\beta_{13} = 3$

$\beta_{21} = 1$
$\beta_{22} = 7$
$\beta_{23} = 0$

$\beta_{31} = 3$
$\beta_{32} = 0$
$\beta_{33} = 7$

Think of α_j as money j is willing to raise to be connected

β_{ij} as the money j is willing to contribute to opening a facility at location i.

Lets think of $\varepsilon = 1$

$t=0$: All $\alpha = 0$, $\beta = 0$, $\hat{T} = \phi$

$t=1$: $\alpha_1 = 1$ $\alpha_2 = 1$ $\alpha_3 = 1$ $\beta_{ii} = 1$, $\hat{T} = \phi$

$t=2$: $\alpha_i = 2$, $\beta_{ii} = 2$, $\hat{T} = \phi$

$t=3$: $\alpha_i = 3$, $\beta_{ii} = 3$, $\hat{T} = \phi$

$t=4$: $\alpha_i = 4$, $\beta_{ii} = 4$, $\hat{T} = \phi$

$t=5$, $\alpha_i = 5$, $\beta_{ii} = 5$, $\beta_{13} = 1$, $\beta_{31} = 1$, $\hat{T} = \phi$

$t=6$: $\alpha_i = 6$, $\beta_{ii} = 6$, $\beta_{13} = 2$, $\beta_{31} = 2$, $\hat{T} = \phi$

$t=7$: $\alpha_i = 7$, $\beta_{ii} = 7$, $\beta_{13} = 3$, $\beta_{31} = 3$, $\beta_{21} = 1$, $\beta_{12} = 1$
$\hat{T} = \phi$

$t = 7\frac{1}{3}$: $\alpha_i = 7\frac{1}{3}$, $\beta_{ii} = 7\frac{1}{3}$, $\beta_{13} = \beta_{31} = 3\frac{1}{3}$, $\beta_{21} = \beta_{12} = 1\frac{1}{3}$

$\swarrow$

facility 1 is tight

$\sum_j \beta_{ij} = 12$

$\Rightarrow$ can't increase any β_{ij} for all j

$\Rightarrow$ can't increase α_j for all j st $\beta_{ij} = d_{ij}$

$\Rightarrow$ freeze all such α_j.

(In this example, all 3 clients freeze at this point)

23/03

- Think of it as a continuous process (can be discretized easily).

- <u>Few</u> <u>observations</u>

- if $\beta_{ij} > 0$ for some (i, j) pair then i becoming frozen / added to $\hat{T}$ [tight constraint] $\sum \beta_{ij} = \lambda$. gets frozen due to $\alpha_j =$ (ie) $\beta_{ij} + d(i, j)$

(algo increases β_{ij} only when constraint becomes tight)

<u>Next</u>

if j

Then $d(i, j) \leq \alpha_j$

Only those clients who can't increase their α anymore due to i freezing are frozen.

(ie) these clients have

$\alpha_j - \beta_{ij} = d(i, j)$ is tight

$\Rightarrow$ $\boxed{d(i, j) \leq \alpha_j}$ since $\beta_{ij} \geq 0$

$\Rightarrow$ If we open all the facilities at $\hat{T}$ then the connection cost of all points (at the end)

Total Conn Cost to $\hat{T}$ $\leq \sum \alpha_j \leq$ dual OPT $\leq$ OPT.

<u>But</u> What about total facility cost of opening $\hat{T}$?

Individually, each facility in $\hat{T}$ is a reasonable choice to open b/c $\sum_j \beta_{ij} = \lambda$

and so there are enough clients willing to share some money to open it.

But issue is collective,

· Same client could have $\beta_{ij} > 0$ to multiple facilities in $\hat{T}$.

GOOD CASE

Sps $\forall$ points j, there is atmost 1 facility $i \in \hat{T}$ for which $\beta_{ij} > 0$

Then I claim that overall it is a great solution

(ie) CONN Cost + $\lambda |\hat{T}| \leq$ OPT

from assumption: For each j, there is only one $\beta_{ij} > 0$ and it is the one which freezes j.

CONN Cost$(j, \hat{T})$ + $\lambda |\hat{T}|$

$$\sum_j \alpha_j = \sum_{\text{edges in above graph}} (d(i,j) + \beta_{ij})$$

$$= \text{CONN cost}(j, \hat{T}) + \sum_{i \in \hat{T}} \left(\sum_{j \to i} \beta_{ij} \right)$$

Hence, CONN Cost + opening Cost $\leq \sum \alpha_j \leq$ OPT

How to handle general case when some j is willing to put $\beta_{ij} > 0$ to multiple i's among $\hat{T}$

- Form a graph with vertices in $\hat{T}$
- Edges (i, i') iff $\exists j$ st $\beta_{ij} > 0$

& $\beta_{i'j} > 0$

- Pick a maximal independent set in this graph. $T_{ind} \subseteq \hat{T}$

Means No edges amongst T_{ind}

and $\forall \; i \in \hat{T} \setminus T_{ind}$,

$\exists \; i' \in T_{ind} \; (i, i')$ is edge.

$\hat{T}$ is all frozen centers for which $\sum_j \beta_{ij} = \lambda$

edge iff $\exists \; j$ st $\beta_{ij} > 0$ & $\beta_{i'j} > 0$

$\hat{T} = \{ i : \sum_j \beta_{ij} = \lambda \}$

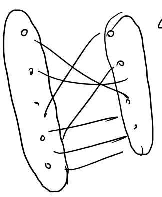
Given graph $G = (V, E)$

a set of vertices $I \subseteq V$

is "INDEPENDENT SET"

if $\forall \; i_1, i_2 \in I$, there

is no edge $(i_1, i_2) \in G$



example, each side of a bipartite graph is an independent set.

In our case

T_{ind} is a "maximal Independent set"

(IE) can't add any other vertex to T_{ind} while preserving independence.

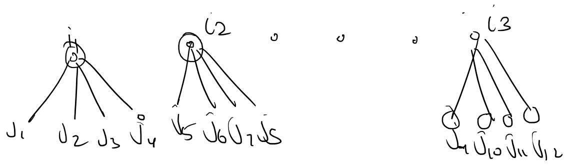
Overall Algo

- Run primal-dual process
- Build graph over $\hat{T}$ and choose maximal independent set
- Open facilities at T_{ind}, and connect all clients to nearest open facility.

ANALYSIS

Firstly, for T_{ind},

the total cost of opening T_{ind} is small.



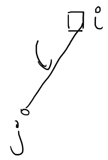
$$\lambda |T_{ind}| = \sum_{i \in T_{ind}} \Big(\sum_{j : \beta_{ij} > 0} \beta_{ij} \Big)$$

j's are disjoint !!

In particular,

$$\lambda |T_{ind}| \leq \sum_j \alpha_j .$$

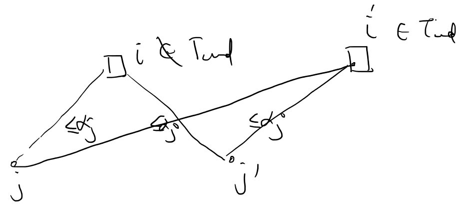
What about connection cost?



Consider client j and suppose it is frozen b/c of i

& $i \in T_{ind}$

Then $d(i,j) \leq \alpha_j$ ✓



<u>Recap</u> 24-03-2021

Facility Location

Given points X, $|X| = n$, distance metric d, and opening cost $\lambda \geq 0$,

Open "centers/facilities" at $S \subseteq X$ to minimize

$$\sum_{j \in X} d(j,S) + \lambda |S| .$$

Formulated Primal-Dual and use dual to infer a solution.

Min — | Max $\sum \alpha_j$

Min

$$\sum_j \sum_i d_{ij} x_{ij} + \sum_i \lambda y_i$$

$\sum x_{ij} \geq 1 \quad \forall j$

$y_i - x_{ij} \geq 0 \quad \forall i, j$

$x_{ij}, y_i \geq 0$

Max

$$\sum \alpha_j$$

$\alpha_j \leq d_{ij} + \beta_{ij} \quad \forall i, j$

$\sum_j \beta_{ij} \leq \lambda \quad \forall i$

Intuition: α_j is money j is willing to invest for its overall happiness.

for any i, α_j breaks up into $d(i,j) + \beta_{ij}$

β_{ij}: Share j is willing to contribute for opening at center i.

Algo:

1. Keep raising all α_j (for unfrozen pts) till all points to nearest open centers in T_{ind} ~~(at) unfrozen~~
2. If necessary, raise β_{ij} @ same rate.
3. If facility $\sum_j \beta_{ij} = \lambda$ is tight for some i, freeze that i, open "Temp facility" (ie) add i to $\hat{T}$.

   and freeze all clients which have tight constraint $\alpha_j = \beta_{ij} + d_{ij}$

4. When all clients frozen, form graph $G = (\hat{T}$, conflict edges$)$

   (ie) (i, i') is an edge iff $\exists$ some j with $\beta_{ij} > 0$ & $\beta_{i'j} > 0$.

5. Pick a maximal independent set ~~and~~ and output that soln (T_{ind})

For ~~Theorem~~ 3-Approx, we need to show: 3-lagrangean Approx.

$\hat{C}$ This Algo $\hat{O}$: $\leq$ 3(OPT + FL)

For lagrangean approximation we need to show

$\hat{C}$ = total conn cost

$\hat{O}$ = total opening cost $\leq \lambda |T_{ind}|$

$$\boxed{\hat{C} + 3\hat{O} \leq 3(OPT - FL).}$$

Proof

Consider T_{ind} and let us break up all the points in X into 2 sets:

G = good pts = $\{ j : \alpha_j - \beta_{ij} = d(ij)$ is tight for some $i \in T_{ind} \}$

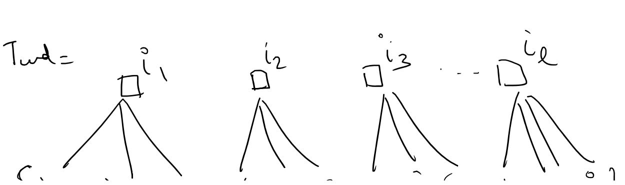
Notice: all pts j with $\beta_{ij} > 0$ for $i \in T_{ind}$ belong to G

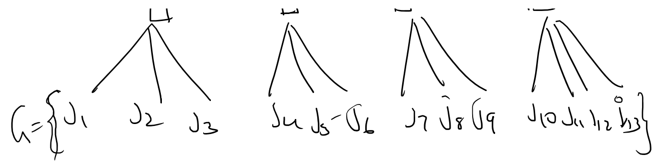
Moreover,

It can't happen that $\beta_{i_1 j} > 0$ & $\beta_{i_2 j} > 0$ for $i_1, i_2 \in T_{ind}$

If not, then there'll be an edge (i_1, i_2) so both can't be in independent set.

$T_{ind} =$





Now,

$\forall j \in G$, let $mate(j)$ be the $i \in T_{ind}$ for which $\beta_{ij} > 0$ (or if no such i exists, then pick any $i \in T_{ind}$ for which the constraint $\alpha_j - \beta_{ij} = d(i,j)$ is tight)

$\forall j$
$$\alpha_j = d(j, mate(j)) + \beta(mate(j), j)$$

Sum up over all $j \in G$

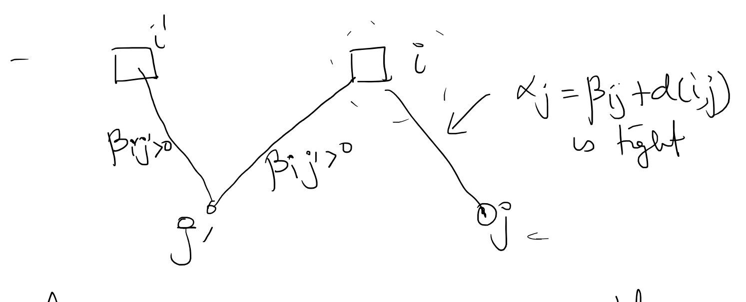
$$\sum_{j \in G} \alpha_j = \text{CONN Cost}(G) + \underbrace{\sum_{j \in G} \beta(mate(j), j)}$$

$$= \quad '' \quad + \lambda |T_{ind}|$$

$$\boxed{3 \sum_{j \in G} \alpha_j = 3\,\text{CONN Cost}(G) + 3 \lambda |T_{ind}|}$$

It remains to bound the bad points.

Lets look @ $j \notin G$ and let the facility which "froze it" be i



Clearly $i \notin T_{ind}$ else j would have been added to T_{ind}

$\Rightarrow \exists\, j', i'$ st
$$\beta_{i'j'} > 0, \quad \beta_{ij'} > 0$$

and $i' \in T_{ind}$

Claim :

① $d(i,j) \le \alpha_j$

② $d(i,j') \le \alpha_j$

③ $d(i',j') \le \alpha_j$

$\Rightarrow \boxed{d(j,i') \le 3\alpha_j}$

---

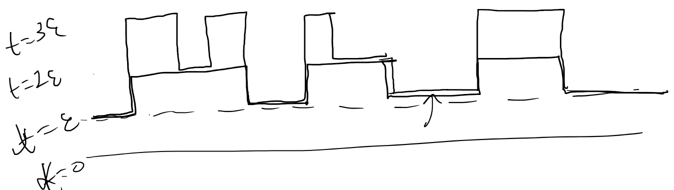
① is clear ($\alpha_j = \beta_{ij} + d(i,j)$ is tight).

For ②

View the dual process as "at time t, α_j for all unfrozen points $= t$"

and view $\beta_{ij} = \max(\alpha_j - d(i,j), 0)$

and if any $\sum_j \beta_{ij} = \lambda$ for some i, freeze all j's for which $\alpha_j = \beta_{ij} + d(i,j)$.



In this view, j froze at time α_j

At this time, i gets frozen or was already frozen.

But definitely, since at this time, $\beta_{ij'} > 0$, j' must be frozen before j did

$\Rightarrow \alpha_{j'} \le \alpha_j$

and because $\beta_{i'j} > 0$ & $\beta_{ij'} > 0$,

we get that

$d(i, j) \leq \alpha_{j'} \leq \alpha_j$

$d(i', j') \leq \alpha_{j'} \leq \alpha_j$

$\Rightarrow$ j has a good conn to T_{ind} of cost $\leq 3\alpha_j$

SUMMARY

$$3\sum_{j \in G} \text{CONN Cost}(j) + 3\lambda |T_{ind}| \leq 3\sum_{j \in G} \alpha_j$$

$$\sum_{j \notin G} \text{CONN Cost}(j) \leq 3\sum_{j \notin G} \alpha_j$$

Overall

$$\text{Total CONN Cost} + 3\lambda |T_{ind}| \leq 3\sum \alpha_j$$

$$\leq 3(\text{OPT-FL})$$

$\because \alpha$ is feasible soln. dual S