



CS 602

Approximation Algorithm

Design algorithm that strictly runs in polynomial time ($n^{O(1)}$)
 Output is allowed to be a "provable" factor away from
 the optimal solution.

Maximization Problems

Ind Let
 Variable $\alpha > 1$ set of vertices such that no two of them are connected

α -approx if we output a solution that is $(\frac{1}{\alpha})$ to the optimal solution

Minimization problems

Hamiltonian Cycle
 Cycle that visits every vertex of G exactly once and returns back α -opt if we output a solution that is at most α of the optimal solution

Polynomial-time approximation solution

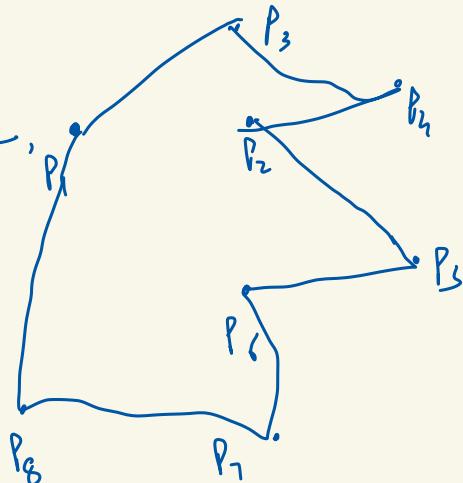
algorithm (of with some parameter $\epsilon > 0$) for any input, output a solⁿ within a factor $(1 + \epsilon)$ of the optimal solution.
 that runs in $n^{f(\frac{1}{\epsilon})}$ for some comparable f^n .
 (Running time is polynomial in $n, \text{some } \epsilon$)

Traveling Salesman Problem (TSP)

Given a list of cities ($P \subseteq \mathbb{R}^2$), and distances between each pair of cities, goal is to compute the shortest possible route that visits every city exactly once.

Decision Version

Given a length L ,
is it possible
to find a solution
of length L .



Graph:

A set of vertices, edges, weights $G = (V, E, w)$
Visit all the vertices without repetition (minimize the sum
of edge weights)

↓
Hamiltonian cycle problem

- * Hamiltonian cycle is NP-complete (Richard Karp)
- * No constant factor abs! is possible

Symmetric

Some edge weights
on both
directions

Asymmetric



Metric Space :

- $d(u, v) \geq 0$
- $d(x, y) = d(y, x)$

- Triangle inequality
 $d(x, y) + d(y, z) \geq d(x, z)$

x . y
. z

$\text{cost}(S) = \text{sum of the weights of the edges (union)}$

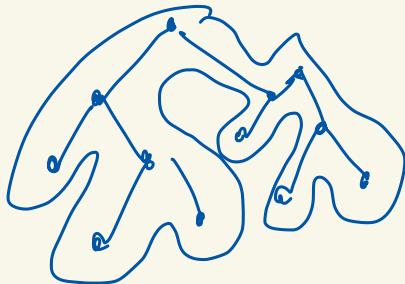
$S = \text{set of edges}$

same as finding the \min of Hamiltonian path
path cycle

Base Structure

- Min Spanning Tree [kruskal]

Due to triangular inequality, we can remove duplicates and cost is reduced



Do the DFS traversal and delete duplicates

Analysis

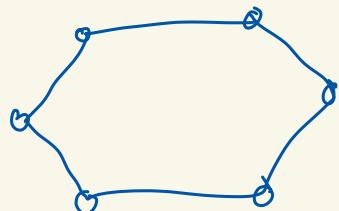
Every edge of the MST is traversed twice

$$\text{cost}(C) \leq 2 \times \text{cost}(\text{MST})$$

valid cycle

$$\text{cost}(\text{MST}) \leq \text{cost}(\text{opt})$$

$$\text{cost}(C) \leq 2 \times \text{cost}(\text{opt})$$



Q Can we do better?

$$I/O = G = (V, E) \xrightarrow{\quad} OPT_G$$

(i) Take a subset

Induced subgroup

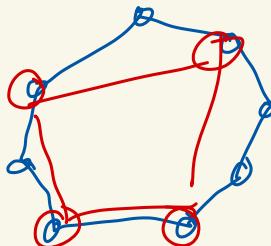
$$S \subseteq V$$

$$G_i(S)$$

$$\xrightarrow{\quad} OPT_S$$

$$OPT_S \leq OPT_G$$

Triangle
Inequality



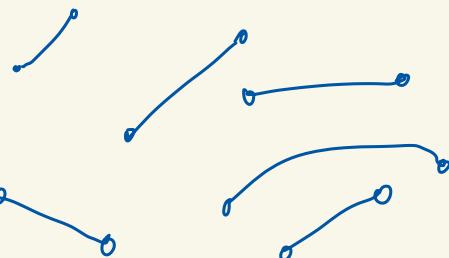
Property-2

Perfect Matching (can be computed in polynomial time)

Min cost AM

Perfect Matching with
smallest cost

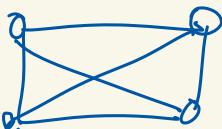
(Polynomial Time
 $\sim O(n^2)$)



Eulerian Tour (Circuit)

vertices can be repeated

each node has
even degree than
this is possible



$$\sum d(v_i) = 2 \times [\epsilon]$$

↑
every edge connected twice

Q How many odd degree vertices we have?
even

We will add another edge for perfect matching

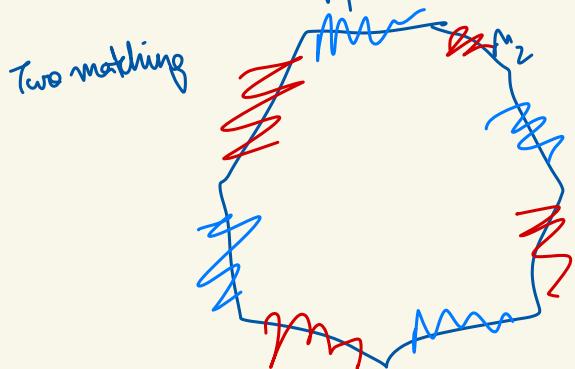
Compute Eulerian circuit

→ Vertex repetition is allowed, but we can delete the duplicates.

Analysis

$$\begin{aligned} \text{Cost } (C) &= \underbrace{\text{Cost } (\text{MST})}_{\leq \text{Cost } (\text{optimal})} + \text{Cost } (\text{Matching}) \\ \text{Cost } (C') &\leq \frac{1}{2} \text{Cost } (\text{optimal}) \end{aligned}$$

$\text{Cost } (C') \leq \frac{3}{2} \text{Cost } (\text{optimal})$



$$\begin{aligned} \text{Cost } (M_1) &\leq \text{Cost } (\text{optimal}) \\ \text{Cost } (M_2) &\leq \text{Cost } (\text{optimal}) \\ \text{Cost } (M') &\leq \frac{1}{2} \text{Cost } (\text{optimal}) \\ \text{Matching we choose because it is minimum} \end{aligned}$$

Metric TSP

- 2- α_{PR} [MST doubling]
- 1.5- α_{PR} (Christofides algorithm, 1976)

$$\downarrow \\ 1.5 - \varepsilon$$

$$\varepsilon = 10^{-30} \quad (2021)$$

No α_{PR} is possible if the distances are arbitrary.

$$G = (V, E, W)$$

- Determine if there is a hamiltonian cycle & some length t .

G' - complete graph

TSP in G' of wt n .

Does there exist in PTAS $(1+\varepsilon)-\alpha_{\text{PR}}$ for metric TSP
No $n^{o(1)}$ time

Theorem: There can't be a PTAS $(220/219) - \alpha_{\text{PR}}$, unless $P = NP$.

Restrict the metric

$\overline{\mathcal{I}}$

Euclidean metric

A set of points in \mathbb{R}^2 , with euclidean distances

$$d(x, y) = \|x - y\|_2$$

Find the shortest route that covers all the points.

The Traveling Salesman Problem

$$\text{Cities} = \{1, 2, \dots, n\}$$

$C(n \times n)$ matrix \rightarrow Cost of traveling between pairs of cities

\downarrow
symmetric

$\underbrace{\quad}_{\text{If we view this as undirected complete graph}}$
then the problem is $\underbrace{\text{Hamiltonian cycle}}_{\text{problem.}}$

Approximation algorithms for the
TSP can be used to solve the Hamiltonian cycle problem

NP-complete

$$G_1 = (V, E)$$

$$C_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ n+2 & \text{otherwise} \end{cases}$$

If Hamiltonian cycle then
tour = n

$$\text{otherwise } \geq (n+2) + (n-1) = 2n+1$$

$\underbrace{\quad}_{\text{Input to TSP-algo}}$

we can detect
hamiltonian cycle $\leftarrow \begin{cases} 2\text{-apx can increase the cost to } 2n \\ \text{for hamiltonian cycle} \end{cases}$

\downarrow
Contradiction! (because HC is NP-complete)

Assumption: Restrict attention to metric space (metric TSP)

Algo(1): A spanning tree of a connected graph $G_1 = (V, E)$ is a minimal subset of edges $F \subseteq E$ such that each pair of nodes in G is connected by a path using edges only in F .

minimum spanning tree: Total edge cost minimized.

* Cost (optimal tour of TSP) \geq cost (MST)

Take this tour and remove one edge

(We will get a spanning tree whose cost $>$ cost (MST))

Algo(1) = nearest addition algorithm \rightarrow 2-apx algo

\downarrow

$F = \{(i_2, j_2), \dots, (i_n, j_n)\}$ \rightarrow edges obtained

$OPT > \sum_{l=2}^n c_{i_l j_l}$

\downarrow
Minimum spanning tree

Cost of the first
two nodes (i_2, j_2)

\downarrow
 $2c_{i_2 j_2}$ (traversed
two times)

j is inserted between (i, k)

marked

$$c_{ij} + \underbrace{c_{jk} - c_{ik}}_{\leq c_{ij}} \leq 2c_j$$

$$\text{cost (nearest-addition algo)} \leq 2 \sum_{l=2}^n c_{i_l j_l} \leq 2(OPT)$$

* Eulerian graph \rightarrow traversal of edges (each edge exactly once)

A graph is eulerian iff it is connected and each node has even degree

Algo(II) \rightarrow Double Tree Algorithm

MST compute \rightarrow replace each edge by two copies of itself

↓
resulting graph is Eulerian and has cost $\leq 2(\text{OPT})$

Eulerian Traversal \rightarrow sequence of edges (but vertices might repeat)

i_0, i_1, \dots, i_k remove all but the first occurrence of each city in this sequence.

↳ Tour of each city once

two consecutive cities (i_1, i_m)

we have removed i_{l+1}, \dots, i_{m-1}

By triangle inequality, cost is decreased,

↳ In total cost is at most the total cost of all the edges in the Eulerian graph
 $\leq 2(\text{OPT})$

double-tree = 2-apx algo.

Christofides Algorithm : MST Comput

↳ $O = \text{set of odd-degree vertices}$

For a tree, sum of degrees = $2 \times |E| = \text{even}$

↳ number of odd degree vertices = $|O| = \text{odd}$

$|O| = 2k$

→ perfect matching $(i_1, i_2), \dots, (i_{2k-1}, i_{2k})$

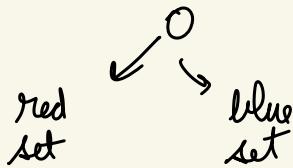
perfect-matching of minimum cost = $O(n^2)$

* Christofides = $\frac{3}{2}$ -apx algo

MST has cost $\leq \text{OPT}$

tour on nodes of $\underbrace{\mathcal{D}}$ has cost $\leq \text{OPT}$
 \downarrow
subset of original graph

Consider the shortest tour on the node set \mathcal{D} . Colour edges
red and blue



$$\text{cost(red)} + \text{cost(blue)} \leq \text{OPT}$$

$$\min(\text{cost(red)}, \text{cost(blue)}) \leq \frac{\text{OPT}}{2}$$

Perfect matching cost \leq \rightarrow Perfect matching + MST $\leq \frac{3}{2} \text{OPT}$

* For any constant $\alpha < \frac{220}{219}$ no α -apx for the metric TSP.

Euclidean TSP

Given n points in \mathbb{R}^2 with Euclidean distances i.e.

$$d(x_i, y) = \|x - y\|_2 \quad \text{shortest tour that visits all points?}$$

Euclidean TSP = NP-hard (do not know NP) length might be irrational

ϵ -nice instance

- (1) Every point has integral coordinates in the interval $[0, O(\frac{m}{\epsilon})]^2$
- (2) Any two different points have distances at least 4.

Consider the smallest bounding box around the points of the input instance. L = longer side of the box

$$L = \left\lceil \frac{8n}{\epsilon} \right\rceil$$

* OPT_I = length of the optimum tour in I . We can transform I into an ϵ -nice instance I' such that $OPT_{I'} \leq (1+\epsilon)OPT_I$

Proof: $I \rightarrow$ smallest bounding box \rightarrow length longer = L

optimal tour $\geq 2L$ \leftarrow $\begin{cases} \text{two points opposite in} \\ \text{the box have distance} \\ \geq L \end{cases}$

Now to obtain $I' \rightarrow$ draw a fine grid with spacing $\frac{\epsilon L}{2n}$ and map each point to the closest grid point.

$$L = \left\lceil \frac{8n}{\epsilon} \right\rceil = O\left(\frac{n}{\epsilon}\right) \quad \frac{\epsilon L}{2n} > \frac{\epsilon}{2n} \times \frac{8n}{\epsilon} = 4$$

$\Rightarrow I' = \epsilon\text{-nice}$ (integer coordinates and $d_{ij} > 4$)

By mapping points of I' to the points in I , we moved each point at most by $\frac{\varepsilon L}{2n}$

→ Edge changed by $\frac{\varepsilon L}{n}$ n edges $\rightarrow \varepsilon L$
 $\leq \text{opt}$

$$\text{OPT}_{I'} \leq \text{OPT}_I + \varepsilon L \leq (1+\varepsilon) \text{OPT}_I$$

VC dimension

Range space $S = (X, R)$

elements of $X \rightarrow$ points
elements of $R \rightarrow$ ranges

↓
ground set
(finite or infinite)

family of subsets of X
(finite or infinite)

x = finite subset of X

measure of a range $\bar{m}(x) = \frac{|x \cap X|}{|x|}$

subset N (might be a multi-set)
of x

estimate of the measure $\bar{m}(x)$
is $\bar{s}(x) = \frac{|x \cap N|}{|N|}$

$Y \subseteq X$ $R_{1Y} = \{x \cap Y \mid x \in R\}$

projections of R on Y . The range space
S projected to Y is $S_{1Y} = (Y, R_{1Y})$

If R_{1Y} contains all subsets of Y ($\text{if } Y \text{ finite, } |R_{1Y}| = 2^{|Y|}$)

then Y is shattered by R

VC dimension ($\dim_{VC}(S)$) maximum cardinality of a
shattered subset of X .

Interval $\rightarrow VC = 2$

Disks $\rightarrow VC = 3$

Convex sets $\rightarrow VC = \infty$

Complement : range space $S = (X, R)$ $\dim_{VC}(S) = \bar{S}$

$$\bar{S} = (X, \bar{R})$$

$$\bar{R} = \{X \setminus \sigma \mid \sigma \in R\}$$

If S shatters B , then for any $Z \subseteq B$, $(B \setminus Z) \in R_{|B}$

$$Z = B \setminus (B \setminus Z) \in \bar{R}_{|B}$$

$\Rightarrow \bar{R}_{|B}$ contains all the subsets of B .

$\Rightarrow \bar{S}$ shatters $B \Rightarrow \dim_{VC}(\bar{S}) = \dim_{VC}(S)$

* Let $P = \{p_1, \dots, p_{d+2}\}$ be a set of $d+2$ points in \mathbb{R}^d . There are real numbers $\beta_1, \dots, \beta_{d+2}$ not all of them zero such that $\sum_i \beta_i p_i = 0$ and $\sum_i \beta_i = 0$

Proof: $q_i = (p_i, 1)$ $q_1, \dots, q_{d+2} \in \mathbb{R}^{d+1}$ are linearly dependent.

There are coefficients $\beta_1, \dots, \beta_{d+2}$ not all of them zero such that $\sum_{i=1}^{d+2} \beta_i q_i = 0$ considering only the first d -coordinates $\sum_{i=1}^{d+2} \beta_i p_i = 0$ $(d+1)^{th}$ coordinate $\sum_{i=1}^{d+2} \beta_i = 0$

Rado's Thm: $P = \{p_1, \dots, p_{d+2}\} \subset \mathbb{R}^d$ Then, there exist two disjoint subsets C and D of P , such that $CH(C) \cap CH(D) = \emptyset$

$$C \cup D = P$$

Proof: By previous thm, $\sum_i \beta_i p_i = 0$ and $\sum_i \beta_i = 0$

$$\mu = \sum_{i=1}^k \beta_i = - \sum_{i=k+1}^{d+2} \beta_i$$

$$\sum_{i=1}^k \beta_i p_i = - \sum_{i=k+1}^n \beta_i p_i$$

$v = \sum_{i=1}^n (\beta_i / \mu) p_i$ is a point in Convex Hull($p_1 \dots p_n$)

$$v = \sum_{i=k+1}^{d+2} -(\beta_i / \mu) p_i \in \text{CH}(p_{k+1}, \dots, p_{d+2})$$

v = intersection of the two convex hulls

* $P \subseteq \mathbb{R}^d$ = finite set s = point in $\text{CH}(P)$ h^+ = halfspace containing s . Then there exists a point of P contained inside h^+ .

Proof: $h^+ = \{t \in \mathbb{R}^d \mid \langle t, v \rangle \leq c\}$

$$\sum_i \alpha_i = 1 \quad \text{and} \quad \sum_i \alpha_i p_i = s$$

$$\langle s, v \rangle \leq c \Rightarrow \left\langle \sum_{i=1}^m \alpha_i p_i, v \right\rangle \leq c \Rightarrow \beta = \sum_{i=1}^m \alpha_i \langle p_i, v \rangle \leq c$$

$\beta_i = \langle p_i, v \rangle$ β is a weighted average of $\beta_1 \dots \beta_m$

\Rightarrow there must be a β_i which is no larger than the average $\Rightarrow \beta_i \leq c \Rightarrow \langle p_i, v \rangle \leq c \Rightarrow p_i \in h^+$.

* Growth Function = $G_S(n) = \sum_{i=0}^n \binom{n}{i} \leq \sum_{i=0}^n \frac{n^i}{i!} \leq n^S$

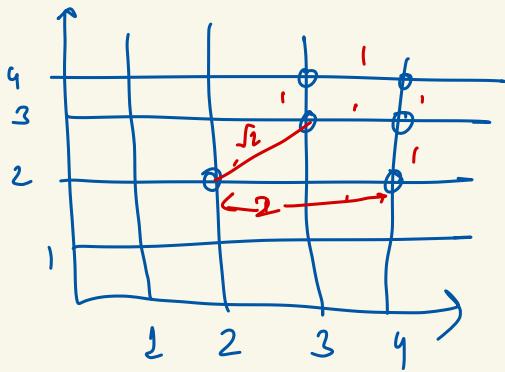
* Sauer's Lemma: If (X, R) is a range space of VC dimension S with $|X| = n$ then $|R| \leq G_S(n)$

Proof: holds for $n=0$ and $S=0$

$$R_x = \{\sigma \setminus \{x\} \mid \sigma \cup \{x\} \in R \text{ and } \sigma \setminus \{x\} \in R\}$$

$$R \setminus x = \{\sigma \setminus \{x\} \mid \sigma \in R\}$$

$$|R| = |R_x| + |R \setminus x| \leq G_{S-1}(n-1) + G_S(n-1) = G_S(n)$$



Euclidean TSP is NP-hard
but not known to be
NP

Sum of square roots (SRS)

Given a set of positive integers
 $\{a_1, \dots, a_k\}$ decide
 $\sum_{i=1}^k a_i \leq t$

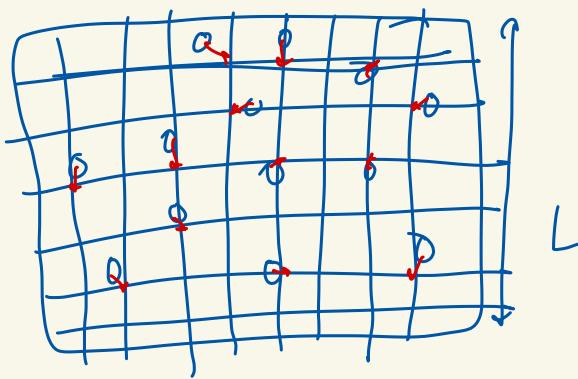
$$\{a_1, \dots, a_k\} \quad \{b_1, \dots, b_k\}$$

$$\sum_{i=1}^k a_i \leq \sum_{i=1}^k b_i$$

NP-hard
not in NP

PTAS - polynomial time approximation scheme.
for euclidean TSP

- Rounding the instance
- Partitioning (Exploit the structure of the instance by breaking it into more instances)
- Dynamic programming



Map each point
to the closest
grid point
(To make the
coordinates
rational)

Given - E

ϵ - "nice" instance

Defⁿ - An instance of Euclidean TSP is ϵ -nice if

1. Every point has integral co-ordinates in the interval $[0, O(\frac{n}{\epsilon})^2]$
2. Any two diff points have dist at least 4.

- Take a small bounding box (axis-parallel)

longer side - L .

s.t. rooted not origin

$$\text{Scale } L = \sqrt{\frac{8n}{\epsilon}}$$

Lemma - I is slp & OPT_I is optimal tour.
 I' is ϵ -nice instance $\text{OPT}_{I'}$ is optimal tour

$$OPT_I \leq (1 + \epsilon) OPT_{I'}$$

OPT is at least $2L$

- Draw a fine grid with spacing $\frac{\epsilon \times L}{2n}$
- Map every pt to its closest grid point (multiple pts could be mapped to the same grid pt).
- All pts have integer coordinates

$$L = \left\lceil \frac{8n}{\epsilon} \right\rceil \text{ or } O\left(\frac{n}{\epsilon}\right)$$

$$\text{Grid spacing } \frac{\epsilon L}{2n} \Rightarrow \frac{\epsilon L}{2n} \times \frac{8n}{\epsilon} = 4$$

→ Mapping each point in I has moved $\frac{\epsilon L}{2n}$

Every edge in the sol'^m changes by at most

$$\frac{\epsilon L}{2n} \text{ edges in OPT}$$

$$\text{Cost} = \epsilon \times L$$

$$\hookrightarrow \text{OPT}_I + \epsilon \times L$$

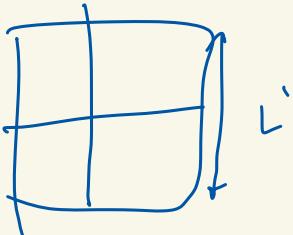
$$\text{OPT}_I + \epsilon \times \text{OPT}_I$$

$$\text{OPT}_{I'} \leq (1+\epsilon) \times \text{OPT}_I$$

$$L \leq \text{OPT}_I$$

Partition the space

- Extend the bonding box to square with new side length L'
 L' is the smallest power 2



* Recursively partition the box
 into four equal sized squares
 until the side length is 1
 ↴ (L' is power of 2)

- each pt is separated
- one pt in each "non-empty" square

Partitioning terminate after $O(\log L')$ steps

Height of the quadtree

- $O(\log L')$
- $O(\lg(\frac{n}{\epsilon}))$

Apply dynamic programming to the quad tree

Solve for each square that are leaves

↓
 Bottom-up combine

Portals

Limits the # of iterations

of portals
 Accuracy improve
 Running Time } Tradeoff

Select $m = \text{power of 2}$

$$m \in \left[\frac{k}{\epsilon}, \frac{2k}{\epsilon} \right]$$

for each square \rightarrow put portals in corners
 put $(m-1)$ portals equally spaced

Portal-respecting tour (p-tour)

Defn: p-tour enters/exits through portals

$$-\text{length of p-tour} \leq (1+\epsilon) \times \text{OPT}$$

Detours can add much more cost

Sol^m → Randomize

(i) Translate the grid by a random offset at most

$$\frac{1}{2} \text{ in each coordinate}$$

(ii) points remain grid points.

(iii) with high probability, the pts are nicely concentrated.

(iv) higher levels in the partition (quad tree)
have more portals → fine-grained tree

Defn
(a,b) dissection : origin of the grid is translated by (-a,-b)

Theorem: (a,b) picked up uniformly at random $\left[0, \frac{1}{2}\right]$ with prob at least $\left(\frac{1}{2}\right)$

p-tour such that cost (p-tour) $\leq (1+4\epsilon) \times \text{OPT}$.

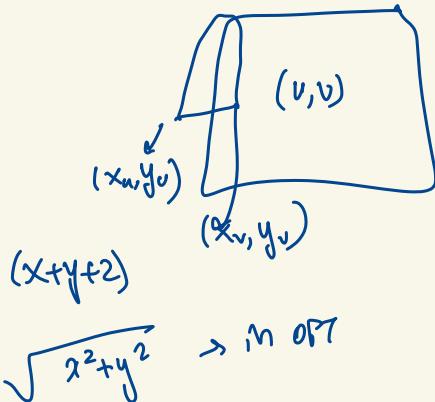
Extend non-ptour to a ptour

Proof: For each vertical/horizontal line l
 $+ (l) = \# \text{ of times intersects } l$

$$T_L = \sum_L t(L)$$

Claim : $T \leq 2 \times \text{opt}$

- e crosses $(x+1)$ vertical lines
- " " $(y+1)$ horizontal "
- total contribution



$$\begin{aligned} \sqrt{2(a^2+b^2)} &\geq a+b \\ \forall x,y \quad d(x,y) &\geq 0 \\ x+y+2 &\leq \sqrt{2(x^2+y^2)} + 2 \\ &\leq 2 \underbrace{\sqrt{x^2+y^2}}_{\text{OPT}} \end{aligned}$$

Bound - expected length of the detours

Detours might occur

$$|x_u - x_v| + |y_u - y_v| + 2$$

i of the quad-tree

$$\frac{L'}{2^{i-m}}$$

if L is in level i

$$\leq \frac{L'}{2^{i-m}}$$

Q: what is the prob that after random shift L crosses L at level-i

- l could be mapped to $\frac{l'}{2}$ many times [translated by $(0, \frac{l'}{2})$]

- 2^{i-1} many lines of level i :

$$\frac{2^{i-1}}{l'/2} = \frac{2^i}{l'}$$

Expected length $\sum_{i=1}^k \frac{2^i}{l'} \times \frac{l'}{2^i m} \leq \epsilon$

By linearity of expectation $2\epsilon \times \text{OPT}$

Markov Inequality

Pr (total length

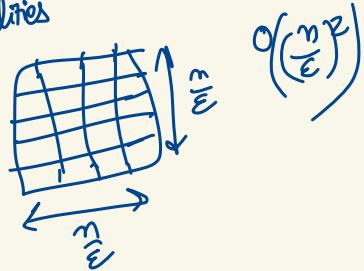
increase of detours $> 4\epsilon \text{ OPT}$)

$$\leq \frac{2\epsilon \text{ OPT}}{4\epsilon \text{ OPT}} = \frac{1}{2}$$

De-randomize

- fixed ϵ

- grid shifting by trying all possibilities



Final Step (DP)

Given (a, b) dissection, get p-tours

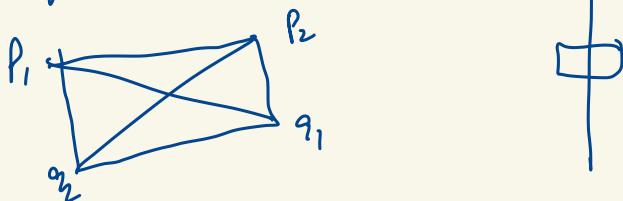
Introduce state

- Square

- any set of possible ways of entering/exiting the squares

$$\# \text{ of states} - (1 + 4 + 4^2 + \dots + L^2) = O\left(\frac{n^2}{\epsilon^2}\right)$$

Lemma: w.l.o.g. a portal is well-behaved 2-light



- 4 portals
- use one portal $m = O\left(\frac{n}{\epsilon}\right) = O\left(\log \frac{L}{\epsilon}\right)$

$$\text{Catalan number} = \frac{1}{2n+1} \binom{2n}{n} = O(2^{2n}) = O(2^{kn})$$

Algorithm = try all parenthesis

- translate them into paths

- Discard anything that intersects

$$m = O\left(\log \frac{n}{\epsilon}\right) \quad \# \text{ of entry exits} = O(n^{1/\epsilon})$$

Computation of values

A $\left[(s_1, t_1) \dots (s_\ell, t_\ell) \right]$ - Compute the whole table.

Clustering

- Learning, searching, data mining
- Given data, find an interesting structure
- Represented as points in \mathbb{R}^d

General metric space (X, d) where X is a set
 $d : X \times X \rightarrow [0, \infty)$

is a metric it satisfies -

- (i) $x=y \rightarrow d_\mu(x, y) = 0$
- (ii) $\forall x, y \quad d_\mu(x, y) = d_\mu(y, x)$
- (iii) $\forall x, y, z \quad d_\mu(x, y) + d_\mu(y, z) \geq d_\mu(x, z)$

Assumption

$(x, y) \quad d_\mu(x, y) \quad$ in $O(1)$ time

Norm

↳ norm defines distances between pts
 $p, q \in \mathbb{R}^d \quad \|p-q\|_p = \left(\sum_{i=1}^d |p_i - q_i|^p \right)^{\frac{1}{p}}$ for $p \geq 1$

$p=2$: Euclidean norm

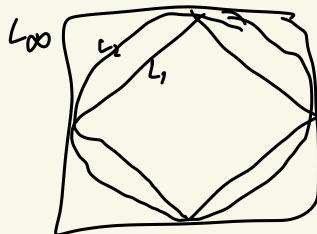
$p=1$: Manhattan distance (L_1 norm)

ℓ_∞ norm

$$\|p - q\|_\infty \leq \lim_{p \rightarrow \infty} \|p - q\|_p$$

max $|p_i - q_i|$

Triangle inequality holds for ℓ_∞ too, it's called Minkowski inequality



for any $p \in \mathbb{R}^d$

$$\|p\|_p \leq \|p\|_2 \quad \text{if } p \geq 0$$

Lemma — For any $p \in \mathbb{R}^d$

$$\|p\|_1 / \sqrt{d} \leq \|p\|_2$$

Proof: $p = (p_1, \dots, p_d)$ $p_i \geq 0 \ \forall i$

Const. α $f(x) = x^2 + (\alpha - x)^2$ minimized if $x = \frac{\alpha}{2}$

$$\text{Let } \alpha = \|p\|_1 = \sum_{i=1}^d |p_i|$$

By symmetry obs on $f(x) = \sum_{i=1}^d x_i^2$

$$\|p\|_2 \geq \sqrt{d(\frac{\alpha}{d})^2} = \|p\|_1 / \sqrt{d}$$

Metric space (X, d)

I/P : A set of points P , $|P|=n$

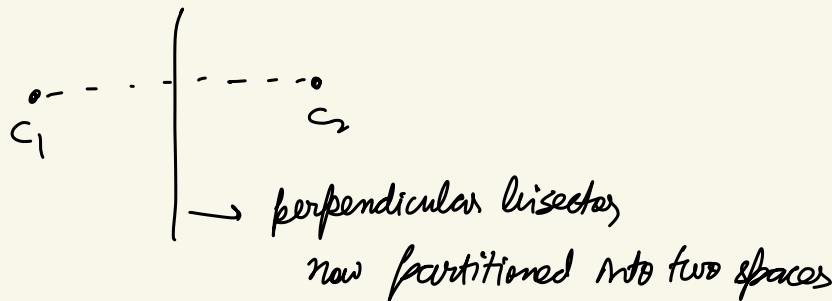
O/P : Find a clustering (set of centers) such that each pt is assigned to its nearest center

Set of clusters C

$$\text{Clusters}(C, \bar{c}) = \{p \in P \mid d_N(p, \bar{c}) = \underbrace{d_N(p, c)}_{\text{J}}$$

Voronoi Partition

minimum
across all the
pts



$$p_c = (d(p_1, c), d(p_2, c), \dots, d(p_n, c))$$

i^{th} coordinate $d(p_i, c)$ dist to p_i to its closest center

O/P = Find a set of k -centers $C \subseteq P$

such that the maximum distance of a point in P to its closest center is minimized

Defⁿ: Given a set of k -centers C ,
 $\|p_c\|_\infty = \max_{p \in P} d(p, C)$

Find C , s.t $\|p_c\|_\infty$ is minimized

$$\text{opt}_\infty(p, k) = \min_C \|p_c\|_\infty \quad C \subseteq P \quad k = |C|$$

- C_{opt}
- NP-hard
- Hard to approximate beyond 1.86
- 2-approx in the Euclidean space in \mathbb{R}^2

Greedy Algo

- Start by picking an arbitrary pt. \bar{c}_1

$$C_1 = \{\bar{c}_1\}$$

- Compute the distances for each $p \in P$ from \bar{c}_1

- Take the pt with worst distance

$$(x_1 = \max_{p \in P} d_1(p))$$

say \bar{c}_2

$$C_2 = C_1 \cup \{\bar{c}_2\}$$

$$C_i = C_{i-1} \cup \{\bar{c}_i\}$$

$O(n)$
space

$O(nk)$
T
data from
previous
iterations

For each pt $p \in P$, a single variable $d[p]$ with its current dist to the closest pt.

$$d[p] \leftarrow \min(d[p], d_N(p, \bar{c}_i))$$

Defⁿ: A ball of radius σ around a pt $p \in P$ is a set of pts in P with dist at most σ from p

$$b(p, \sigma) = \{q \in P \mid d_N(p, q) \leq \sigma\}$$

Remark: k -center is essentially covering P with k -balls of minimum radius.

Thm: Greedy Algo computes a set K of k -center such that k is 2 -apx $\|p_k\|_1 \leq 2 \|p_k\|_\infty$ takes $O(n \times k)$ time.

Proof: Running Time ✓

$$\text{Def}^n \quad r_k = \|p_k\|_\infty$$

Let \bar{c}_{k+1} is the point realising

$$r_k = \max_{p \in P} d(p, k)$$

$$C = K \cup \{\bar{c}_{k+1}\}$$

By the defⁿ of r :

$$r_1 \geq r_2 \geq \dots \geq r_k$$

$$i < j < k+1$$

$$d_N(\bar{c}_i, \bar{c}_j) \geq d_N(\bar{c}_i, \bar{c}_{i-1})$$

$$r_{i-1} \geq r_k$$

— the dist. between any pair of pts. in C is at least τ_k

opt — covers P by using k balls

by triangle inequality any two points within such a ball are with a dist at most $2 \times \text{opt}$.

↓
None of the balls contain two points from
contradiction!

$$C \\ \subseteq P$$

Greedy permutation

Let this run till we exhaust all pts

$$\langle P \rangle = \langle \bar{c}_1, \bar{c}_2, \dots, \bar{c}_n \rangle$$

$$\downarrow \\ \langle \tau_1, \tau_2, \dots, \tau_n \rangle$$

Defⁿ : τ -packing : A set $S \subseteq P$ for P

(i) covering property : all the pts in P are within dist of atmost τ from S .

(ii) separation property : $\forall p, q \quad d_M(p, q) \geq \tau$

τ -packing gives compact representation

* Greedy permutation gives such a rep.

Thm: $\langle \overline{c_1}, \overline{c_2}, \dots, \overline{c_n} \rangle < \infty$
 for any i , we have $c_i = \langle \overline{c_1} \dots \overline{c_i} \rangle$
 is an ∞_i -packing of P

Proof: By contradiction

$$\infty_{k-1} = d(\overline{c_k}, \overline{c_{k-1}}) + k = 2, \dots, n$$

$$\text{for } j < k \leq i \leq n$$

$$d_\mu(\overline{c_j}, \overline{c_n}) = \infty_{k-1} \geq \infty_i$$

K-medians clustering

A set $P \subseteq X$ ($|P|=n$), a parameter k . Find a set of k -points $C \subseteq P$ s.t. the sum of distances of the pts in P to its closest center is minimized.

Clustering price: $\|P_C\| = \sum_{p \in P} d(p, C)$

Objective f^* : $\text{opt}_p(p, k) = \min_{\substack{C \subseteq P \\ |C|=k}} \|P_C\|$

Optimal set of centres - C_{opt} .

Local search: move sol^n to sol^n in the space of candidate sol^n (the search space) by applying local changes

Continue until, end up on optimal or we exhaust the running time.

Notations:

$$\text{A set } U = \{P_c \mid C \in P^k\}$$

$$\text{opt}_{\infty}(P, k) = \min_{\substack{q \in U \\ \text{k-center}}} \|q\|_{\infty} \quad \left| \begin{array}{l} \text{opt}(P, k) = \min_{q \in U} \|q\|_1, \\ \text{k-median} \end{array} \right.$$

1.86 Apx X

2 Apx ✓ (Greedy)

Claim: For any set P , $|P| = n$, k

$$\text{opt}_{\infty}(P, k) \leq \text{opt}_1(P, k) \leq n \times \text{opt}_{\infty}(P, k)$$

$$\begin{aligned} \text{Proof: } P &\in \mathbb{R}^m & \|P\|_{\infty} &= \max_{i=1}^n |P_i| \\ && \leq \sum_{i=1}^n \|P_i\|_1 &= \|P\|_1 \end{aligned}$$

$$\|P\|_1 \leq \sum_{i=1}^n |P_i| \leq \sum_{i=1}^n \max |P_i| \leq n \times \|P\|_{\infty}$$

C -set of centers $|C| = k$ realising $\text{opt}_1(P, k)$ i.e.

$$\text{opt}_c(P, k) = \|P_c\|_1$$

$$\begin{aligned} \text{opt}_{\infty}(P, k) &\leq \|P_c\|_{\infty} \\ &\leq \|P_c\|_1 = \text{opt}_c(P, k) \end{aligned}$$

Similarly, k realizing $\text{opt}_{\infty}(P, n)$

$$\begin{aligned} \text{opt}_k(P, k) &= \|P_k\|_1 \leq \|P_k\|_1 \\ &\leq n \times \|P_k\|_{\infty} \\ &= n \times \text{opt}_{\infty}(P, k) \end{aligned}$$

($2n$ -factor for median)

$2n$ -apx

use this as a first step for local search

L - is $2n$ -apx

Improve : parameter $0 < \delta < 1$
 $\forall i \in [n] L_{\text{curr}}$

Local search

- Set $L_{\text{curr}} \leftarrow L$
- At each iteration



We will check if the current "sol" L_{curr} can be improved

by replacing one of the centers

by one center from outside (non-centers)



Swap

$$K \leftarrow (L_{\text{curr}} \setminus \{\bar{c}\}) \cup \{\bar{c}\}$$

if $\|P_k\|_1 \leq (1-s) \|P_{L_{\text{curr}}}\|_1$

- continue swap as long as it satisfies the constraint
- return L_{curr}

Running time : An iteration takes $O(m \times k)$ swaps

$(n-k)$ candidates to be swapped in k -candidates
to be out)

implementing swap (naively $O(n^k)$)
overall $O(n^{2k})$

Since

$$\frac{1}{1-s} \geq (1+s)$$

$$O((n^k)^2 \log \frac{1}{1-s} \frac{\|P_k\|_1}{\epsilon_{\text{pt}_1}})$$

$$= O(n^k)^2 \cdot \log(1+s)^{2n} = O((n^k)^2 \log \frac{n}{s})$$

K-means

Set $P \subseteq X$, K , find K pts $C \subseteq P$ $|C|=k$

$$\|P_C\|_2^2 = \sum_{p \in P} (d_{\mu_p}(p, C))^2$$

Obj : s.t. $\|P_C\|_2^2$ is minimized

$$\text{Opt}_2(P, k) = \min_{C, |C|=k} \|P_C\|_2^2$$

$O(n)$ -factor for k -means as well

Thm: $0 < \varepsilon < 1$

$(25 + \varepsilon)$ -approx

VC-dim

- A range space $(X, R) = S$

X = ground set (finite / infinite)

R = family of subsets of X .

Consider finite subset of X as the estimating ground set.

Dfⁿ (Measure): fixed subset of X . For a range $\tau \in R$

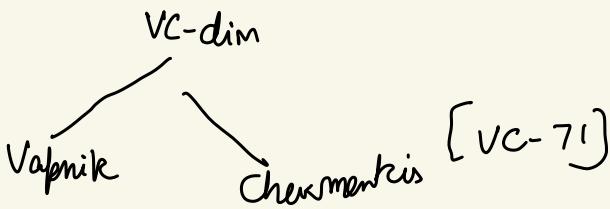
$$m(\tau) = \frac{|\tau \cap X|}{|X|}$$

For a subset N (multi-set) of X , the estimate of the measure of $m(\tau)$, for $\tau \in R$

$$\hat{s}(\tau) = \frac{|\tau \cap N|}{|N|}$$

Q2 Can we get methods to generate N s.t.

$$\overline{S}(x) = \overline{m}(x) \quad \forall x \in \mathbb{R}$$



Dfm: $S = (X, R)$ For $Y \subseteq X$

$$R_{S,Y} = \{\tau \cap Y \mid \tau \in R\}$$

be the projection of R on Y

$\binom{|Y|}{2}$

If this is the cardinality then it is called
shattered by R

The orange
space S_Y
is projected
to $S_{Y'} = \{Y, R_{Y'}\}$

Complement

$$S = (X, R) \quad S = \dim_{VC}(S)$$

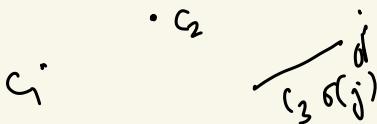
$$\overline{S} = (X, \overline{R}) \quad \text{where } \overline{R} = \{X \setminus \tau \mid \tau \in R\}$$

Q2 what is the VC-dim of \overline{S} ?

A subset $B \subseteq X$, is shattered in \overline{S} iff it is shattered in S
for any $Z \subseteq B$ $(B \setminus Z) \in \overline{R}_{IB} \Rightarrow Z = B \setminus (B \setminus Z) \in \overline{R}_{IB}$

Local search

- X be the set of arbitrary subset of k -centers
while true do:
(Swap) if i.e X and $i' \in F \setminus X$
 $\text{cost}(X - i + i') < \text{cost}(X)$
- (greedy solution of k -centers)



$\text{opt} = X^* - \text{optimal set of } X \leftarrow X - i + i'$
k-center otherwise break

Nearest

$$i^* \in X^*$$

$$i \notin X$$

for each center chosen in X ,
nearest center in X^* $\min_{i^*}(\|i\|_j)$

Inverse

Ties the centers in X^*

inverse is the
nearest map

Bijection

Claim: for any $j \in X$,
 $d_{\text{nearest}}(\sigma^*(i), j) \leq d_j + 2d_j^*$

Half-spaces

Let \mathcal{R} be the set of closed half spaces in \mathbb{R}^d

Claim: $P = \{P_1, \dots, P_{d+2}\}$ set of points in \mathbb{R}^d

Real numbers $\beta_1, \beta_2, \dots, \beta_{d+2}$ (not all are zero)

$$\text{s.t. } \sum_i \beta_i p_i = 0 \quad \& \quad \sum_i \beta_i = 0.$$

Proof: $a_i = (p_i, \beta_i)$ for $i=1 \dots d+2$

pts are linearly dependent . and these are
 $a_1, a_2, \dots, a_{d+2} \in \mathbb{R}^{d+2}$ coefficients $\beta_1, \dots, \beta_{d+2}$
 s.t. $\sum_{i=1}^{d+2} \beta_i p_i = 0$

- Considering first d -coordinates of these pts implies

$$\sum_{i=1}^{d+2} \beta_i p_i = 0$$

$$\text{Similarly } (d+1) \text{ coordinates } \sum_{i=1}^{d+2} \beta_i = 0$$

Radon's Thm: $P = \{P_1 \dots P_{d+2}\}$ \exists disjoint subsets
 $C \& D$ of P . $H(C) \cap H(D) = \emptyset$ then
 $C \cup D = P$

Shattering Dim:

Property : A range space (\mathcal{R}) with $VC\text{-dim}(S)$
 means # of ranges given polynomially on (n)
 (Generally this is \exp^n)

$$\underline{\text{Growth function}}: \quad G_{\delta}(n) = \sum_{i=0}^{\delta} \binom{n}{i} \leq \sum_{i=0}^{\delta} \frac{n^i}{i!} \leq n^{\delta} \quad \text{for } \delta > 1$$

$$\underline{\text{Sauer's Lemma}}: \quad S = (X, R) \\ \text{VC}(S) = \delta \quad |X| = n \quad |R| \leq G_{\delta}(n)$$

Proof: $n=0 \quad \delta=0 \quad \rightarrow \text{done!}$

$$x \in X$$

$$\text{contains}_x \{R_x\} = \left\{ \sigma \setminus \{x\} \mid \sigma \cup \{x\} \in R \text{ and } \sigma \supset \{x\} \in R \right\}$$

$$\text{does not contain}_x \{R \setminus x\} = \left\{ \sigma \setminus \{x\} \mid \sigma \in R \right\}$$

$$\text{Observation: } |R_x| + |R \setminus x| = |R|$$

Shatter function: $S = (X, R)$ shatter f^n
 $\pi_S(m)$ is the maximum # of sets that might be created by S , when restricted to the subsets of size m .

$$\pi_S(m) = \max_{\substack{B \subseteq X \\ |B|=m}} |R_{|B|}|$$

Shattering dim: The smallest d such that $\pi_S(m) = O(m^d)$ $\forall m$

Then $S = (X, R)$ has shattering dim d , then the VC-dim is bounded by $O(d \log d)$

Proof: Let $N \subseteq X$ be the largest subset of X shattered by S and s is the cardinality

$$2^s = |R_{|N|}| \leq \pi_S(m)$$

$$s \leq \log c + d \log s \quad (s \geq \max(2, \frac{2}{c}))$$

$$\Rightarrow \frac{s \cdot \log c}{\log s} \leq d$$

$$\frac{s}{2 \log s} \leq d \Rightarrow \frac{s}{\log s} \leq O(1) \times d$$

$$f(x) = \frac{x}{\log x} \rightarrow \text{non-increasing } x > c$$

$c > \sqrt{e}$ if $f(x) \geq e$ $\Leftrightarrow x > 1$
 $f(x) \leq x$ then $x \leq \log x$

ε -net and ε -sampling

$S = (X, R)$ x is a finite subset of X
 $0 \leq \varepsilon \leq 1$

Informally, ε -sampling captures R , upto some ε -error

a subset $c \subseteq x$ is an ε -sample for x if for any range $r \in R$

$$|\bar{m}(r) - \xi(r)| \leq \varepsilon$$

\downarrow measure \curvearrowright estimate

Thm: (ε -sample, VCT₁) — There is a free constant C s.t. if (X, R) is any range space with VC dim S .

$x \subseteq X$ finite subset of X and $\forall \varepsilon, \phi > 0$
 \exists a random subset $C \subseteq X$ of

(with probability $= \phi$) cardinality $S = \frac{C}{\varepsilon^2} \left(\delta \log \frac{\delta}{\varepsilon} + \log \frac{1}{\phi} \right)$

ε -net : A set $N \subseteq X$ is an ε -net for x if for any range space $r \in R$ if $\bar{m}(r) \geq \varepsilon$
 then r contains at least one pt. of N (i.e. $r \cap N \neq \emptyset$)

(Intuitively: hit all heavy subsets)

ϵ -net Thm (HWF7)

$S = (x, R)$ has $\text{VC dim}(S) \leq S$

x is a finite subset of X .

Suppose $0 \leq \epsilon \leq 1$ & $\phi < 1$

- N a set obtained by random independent draws.
- $m \geq \max\left(\frac{4}{\epsilon} \log \frac{4}{\phi}, \frac{8S}{\epsilon} \log \frac{16}{\epsilon}\right)$
- Then, N is a ϵ -net with prob $(1-\phi)$.

* Remark: Both of the thms hold for spaces with shattering dim S . ($O\left(\frac{1}{\epsilon} \log \frac{1}{\phi} + \frac{S}{\epsilon} \log \frac{S}{\epsilon}\right)$)

Range Searching $p \in \mathbb{R}^d$ we have a database
Given a hyper rectangle, we want to report the points that lie inside

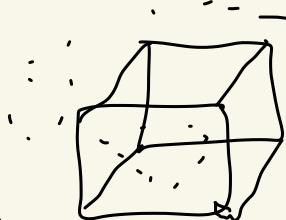
Allow 1% error,

ϵ -sample (Thm) says there is
a subset of const. size (which depends on ϵ)

Use this to perform an estimation.

Rectangle has bounded VC-dim

Random sample with probability $(1-\phi)$



Learning Concepts

Assume we know a f^* that returns 1 if inside,
0 otherwise

Query
Oracle

There is a distribution D defined over the space. We pick points from D .

Growth function $G_d(n) = \sum_{i=0}^d \binom{n}{i} \leq \sum_{i=0}^d \frac{n^i}{i!} \leq n^d$

Sauer's Lemma $S = (X, R)$

Suppose $\text{VC dim}(S) \leq d$

$$|R| \leq G_d(n) \leq \sum_{i=0}^d \binom{n}{i}$$

$T_F(n)$

Proof: By induction on n, d
 $n=d=1$ holds

Assume that it holds for $n-1 \& d$
and as well as for $n-1 \& d-1$

We prove for $n \& d$
define $f^* : \sum_{i=0}^d \binom{n}{i} = h(n, d)$

Our induction hypothesis is for F with $\text{VC-dim} \leq d$

$$T_F(n) \leq d$$

$$\binom{n}{d} = \binom{n-1}{d} + \binom{n-1}{d-1}$$

$h(n, d) = h(n-1, d) + h(n-1, d-1)$

recurrence

Now let's fix a class F

$$VC\text{-dim}(F) = d \quad \text{and a set}$$

$$X_1 = \{x_1, \dots, x_m\} \subseteq X$$

$$f_1 = f_{1X}$$

$$f_2 = f_{2X}$$

$$F_3 = \{f_{1X} | f \in F \text{ & } f' \in F \text{ s.t. } \forall x \in X_2, f'(x) = f(x) \text{ & } f'(x_1) = -f(x_1)\}$$

$$VC\text{-dim}(F')$$

$$\leq VC\text{-dim}(F) \leq d$$

$$|F_1| = |F_2| + |F_3| \leq d \leq d-1$$

Induction hypothesis

$$\begin{cases} |F_2| \leq h(n-1, d) \\ |F_3| \leq h(n-1, d-1) \end{cases} \rightarrow |F_1| \leq h(n-1, d) + h(n-1, d-1) \leq h(n, d)$$

Ex. Let F be s.t. $VC\text{-dim}(F) \leq d$ for $n \geq d$

$$\pi_F(n) \leq \left(\frac{mc}{d}\right)^d$$

Set-cover / Hitting set (piercing)

U = universe of elements

X = set of subsets

$S = (U, X)$ - set system

choose a subset $X' \subseteq X$
which is a cover

- NP hard

Greedy approximation

- (1) sort all the sets based on cardinality
- { (2) choose the set with max cardinality.

↳ \exists a lower bound shows that we can't get better than log factor.

wish: for "nice" set families, can we beat greedy (log factor)

$$S = (X, R)$$

↓

set of elements set of ranges

Goal: choose a subset $R' \subseteq R$ that covers X .

class of objects \rightarrow has bounded VC-dim

ε -net: Sample that "wits" all the heavy sets ($> \varepsilon_n$)

A set $N \subseteq X$ is an ε -net for a finite subset x if
for any range $r \in R$, $m(r) = \frac{m(x)}{|x|} > \varepsilon$

then r contains at least one pt.

Construction of ε -net

- choose a random sample if it is large enough its ε -net
- small hitting set " (which is also a net)

ε -net thm

We can get a subset N by m ind. draws for a finite subset x .
(uniformly chosen)

$$N \geq \max \left\{ \frac{9}{\varepsilon} \log \frac{a}{\phi}, \frac{8\varepsilon}{\varepsilon} \log \frac{16}{\varepsilon} \right\}$$

with prob $> 1 - \phi$.

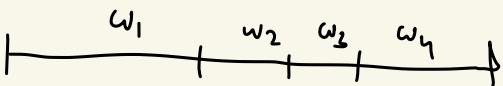
Suppose the shattering dim is d .

$$\text{sample size} \geq O\left(\frac{d}{\varepsilon} \log \frac{d}{\varepsilon}\right)$$

Weighted net: suppose the elements are wt,
($W: x \rightarrow \mathbb{R}^+$)

r -subset
 $w(r) = \sum_{j \in r} w(j)$

Goal : all the r 's with wt. $> \epsilon x_w$



Algorithm for set cover

(1) Repeatedly select an ϵ -net (for some ϵ)

$$S = (X, R) \text{ dual} - S^* = (X^*, R^*)$$

\downarrow
shattering dim S^*

$$\text{size of the net} = O\left(\frac{\xi^*}{\epsilon} \log \frac{\xi^*}{\epsilon}\right)$$

Verify if it is a net. If not - discard.

\downarrow
if it is a net check if it is a setcover
if yes - done

Let $R_p = \{\infty \in R \mid p \in \infty\}$ all ranges that contain p .

Double the weight of the elements in R_p

Observation: every time we double we are increasing not more than $(1+\epsilon)$ multiplicative function

$$w_i \leq (1+\epsilon)^i w_0$$

$i \rightarrow$ iteration

Q1 What is the min wt. of elements ($K = \text{optt}$) in opt?

$$K \times 2^{\frac{i}{k}} \leq w_i = (1+\varepsilon)^i w_0 = (1+\varepsilon)^i x_m$$

$$\leq \varepsilon^{\frac{i}{k}} x_m$$

$$K \times 2^{\frac{i}{k}} \leq w_i$$

$$i = k \times g$$

$$k \times 2^g \leq e^{\varepsilon i} x_m$$

$$\log(k + g) \leq \log m + \sum i$$

$$g(1-\varepsilon k) \leq \log m - \log k = \log\left(\frac{m}{k}\right)$$

Suppose, we take $\varepsilon = \frac{1}{2k}$

$$\Rightarrow g \leq O\left(\log \frac{m}{k}\right)$$

$$\# \text{ of iteration } O\left(k \log \frac{m}{k}\right)$$

Size of our cover $O(S^*k \log S^*k)$ $k = \text{opt. cover}$

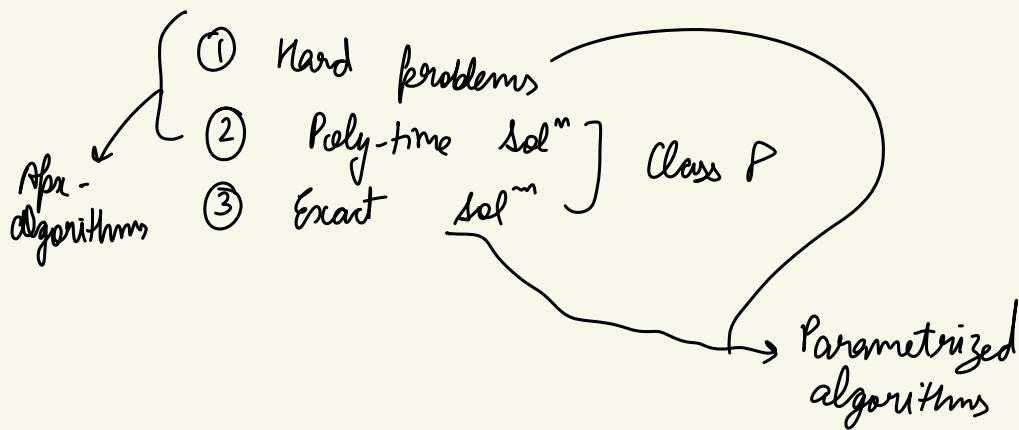
Q2 How to choose ε ?

ε is independent on k

Suppose $\varepsilon = \frac{1}{uk}$ instead of $\frac{1}{2k}$

Guess the value of $\frac{\varepsilon}{k_i}$

$O(k_i \log \frac{m}{k_i})$ iterations



Idea: Aim is to get exact algo

But we want to isolate \exp^n terms (parameters)

\Rightarrow obtain very fast solⁿ when the parameter small.

(Note: parameters are small in practice).

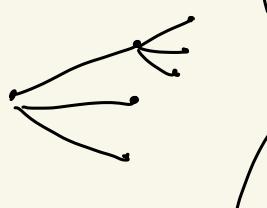
parameters - non-negative integer $k(x)$ (comes with prob i/p)
 - denote by k
 - Not necessarily efficiently computable.

Parametrized Problem

problem + parameter (k)
 (w.r.t. k)

Goal: poly. complexity on n
 Expⁿ complexity on k

Example:



I/p : $G = (V, E)$ $k \in \mathbb{N}$

o/p : Does there exist a
 k -size vertex cover

↓
output a set ($\subseteq V$) s.t. $\forall e \in E$
 $\exists v \in S$

Brute force solution

(1) Try all $\binom{n}{k} + \binom{n}{k-1} + \dots + \binom{n}{0}$
 ↓
 All sets of k vertices

— Test valid VC takes $O(E)$ time

— Total = $O(V^k E)$ kely for fixed k .

slow for large n and reasonable k .

Branching (Bounded search tree technique)

→ Pick an arbitrary edge $e \in E$
 ↓
 (v, v')

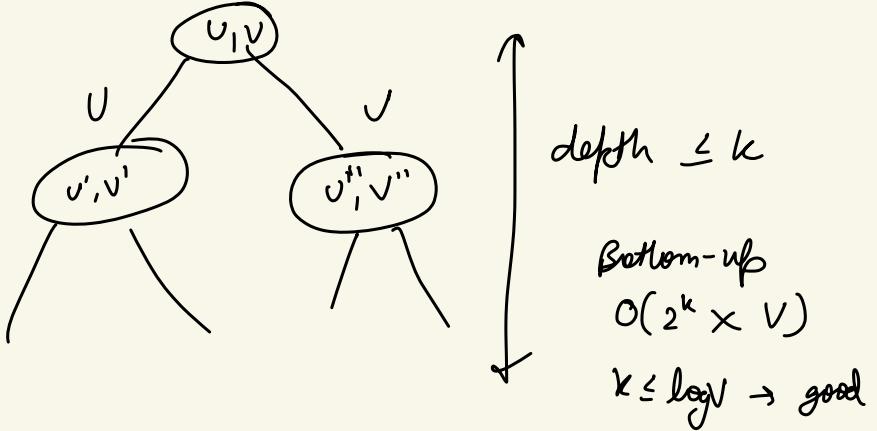
→ Know either $v \in S$ or $v' \in S$ or $\{v, v'\} \in S$

Guess — Try both

① Add v to S
 (delete v & $N(v)$ from S)
 recursive $k' = k - 1$

② same for v' .

— Return or of the outcomes



Fixed parameter-tractable (FPT)

If \exists an algo with running time $\leq f(k) \times n^{O(1)}$

$f: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ polynomial
 parameters

Q: why $f(k) \times n^{O(1)}$ and not $f(k) + n^{O(1)}$?

Thm: $f(k) \times n^c \Leftrightarrow f(k) + n^{c'}$

Proof: \Rightarrow if $n \leq f(k)$

$$f(k) \times n^c \leq f(k)^{c+1}$$

if $f(k) \leq n$

$$f(k) \times n^c \leq n^{c+1}$$

$$\text{So. } f(k) \times n^c \leq \max \left\{ f(k)^{c+1}, n^{c+1} \right\} \leq f(k)^{c+1} + n^{c+1}$$

(\Leftarrow Trivial, assuming $f(k) & n^{c'} \geq 1$)

Kernelization

simplifying
self-reduction

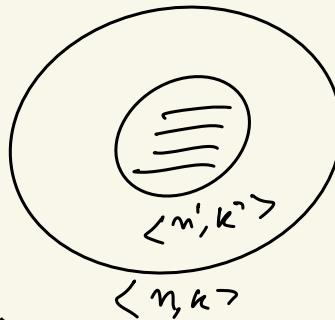
(poly-time reduction)

i/p - $\langle n, k \rangle$

converts it into $\langle n', k' \rangle$

How small? $|n'| \leq f(k)$

Equivalent — Ans($\langle n, k \rangle$) = Ans($\langle n', k' \rangle$)



Thm:

FPT \Leftrightarrow kernelization

kernelization $\Rightarrow n' \leq f(k)$

run any finite $g(n')$

$\Rightarrow n^{O(1)} + g(f(k))$ time \rightarrow FPT

\Leftrightarrow

A runs in $f(k) > n^c$

if $n \leq f(k)$ in kernelized

if $f(k) \leq n$

run A $\rightarrow f(k) \times n^c \leq n^{c+1}$

O/p, O(1) size

k is known in advance

Sunflower Lemma (Erdős Rado Cons)

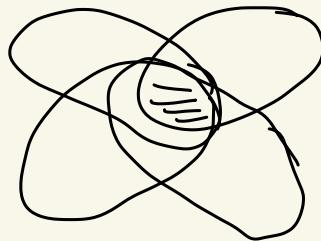
- Classical result from 1960

- Apply in kernelization

- k petals

- A core γ

(\Rightarrow None of them can be empty)



Collection of sets S_1, \dots, S_k

s.t. $S_i \cap S_j = \gamma$

$\forall i \neq j$

Petals $\forall i \in S \setminus \gamma$

Lemma F - family of sets (no duplication) over a universe U

s.t. each set has cardinality exactly d .

- If $|F| > d! (k-1)^d$, then F contains k petals

- Poly-time algorithm to compute this

w.r.t. $|F|, |U|, k$

Proof: for $d=1$, singletons

Suppose $d \geq 2$

Let $G = \{S_1, S_2, \dots, S_l\} \subseteq F$

- If $l \geq k$ then G is a sunflower

already with at least k petals

Assume $l < k$

be inclusion-wise
maximal family of
pairwise disjoint sets
in F

$$S = \bigcup_{i=1}^k S_i; \quad \text{Then } |S| \geq d \times (k-1)$$

Since G is maximal, every set $A \subseteq F$ intersects at least one set from G . $A \cap S \neq \emptyset$.

There is an element $v \in V$ which is contained in at least

$$\frac{|F|}{|S|} \text{ sets.} \quad \frac{|F|}{|S|} > \frac{d! (k-1)^d}{d(k-1)} = \underbrace{(d-1)!}_{\text{sets for } F} (k-1)^{d-1}$$

- Take all sets of F containing this element v .

Construct F' of sets union cardinality $(d-1)$ by removing v .

$$|F'| > d!. (k-1)^d$$

By induction hypothesis

F' contains a sunflower $\{S'_1 \dots S'_n\}$ with k -petals

$$\{S'_1 \cup v\} \dots \{S'_n \cup v\}$$

Poly-Time Algorithm

(1) greedily select maximal sets. If size is at least k done. Else find v and return.

Erdos-Rado - FIGO

Each set in F has cardinality d

If $|F| > d!(k-1)^d$ then there is a sunflower.

$(\log k)^d \rightarrow$ recent bound.

d -Hitting set

(Application of Sunflower Lemma)

Input: Family of sets A over V . each set has cardinality at most d .

a non-negative integer k

Output: whether there is a subset $H \subseteq V$ of size at most k , such that H contains 1 element of each of sets of A .

Proof: If A contains a sunflower, say $S = \{S_1, \dots, S_{k+1}\}$ of $\#(k+1)$ then every hitting set H of A of cardinality at most k intersects its core Y .

Reduction rule : (V, A, k)

Return (V', A', k') $A' = (A \setminus S) \cup \{x\}$
and $V' = \bigcup_{x \in A'} X$

If # of sets are larger than $d! \times k^d$ find a sunflower
— Apply green consider kernel size $O(d! k^d)$

Kernelization

A data reduction rule for a parametrized problem \mathcal{Q} is a function $\phi : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N}$ that maps an instance (I, k) of \mathcal{Q} to an equivalent instance (I', k') of \mathcal{Q} such that ϕ is computable in time polynomial in $|I|$ and k .

$$\text{size}_A(k) = \sup \left\{ |I'| + k' : (I', k') = A(I, k), I \in \Sigma^* \right\}$$

* A kernelization algorithm for a parametrized problem \mathcal{Q} is an algorithm A that, given an instance (I, k) of \mathcal{Q} works in polynomial time and returns an equivalent instance (I', k') of \mathcal{Q} . Moreover, we require that $\text{size}_A(k) \leq g(k)$ for some computable function $g : \mathbb{N} \rightarrow \mathbb{N}$.

* If a parametrized problem \mathcal{Q} is FPT then it admits a kernelization algorithm

Proof: $\mathcal{Q} = \text{FPT} \Rightarrow \exists A (I, k) \in \mathcal{Q}$ in time $f(k)|I|^c$
 (I, k) algo runs A on (I, k) for at most $|I|^{c+1}$ steps
 If it terminates with an answer, use that for yes/no.
 If A does not terminate within $|I|^{c+1}$ steps, then return (I, k) itself

$$f(k) \cdot |I|^c > |I|^{c+1} \Rightarrow |I| < f(k)$$

$$|I| + k \leq \underbrace{f(k) + k}_{\text{computable}} \quad (\text{kernel size})$$

Sunflower Lemma

A sunflower with k petals and a core γ is a collection of sets S_1, \dots, S_k such that $S_i \cap S_j = \gamma$ for all $i \neq j$; the sets $S_i \setminus \gamma$ are petals and we require none of them to be empty (γ can be empty).

* Let A be a family of sets (without duplicates) over a universe U , such that each set in A has cardinality exactly d . If $|A| > d!(k-1)^d$, then A contains a sunflower with k petals and such a sunflower can be computed in time polynomial in $|A|, |U|$ and k .

For $d=1$, family of singletons, statement holds
 $d \geq 2$ $A = \text{family of sets of cardinality at most } d \text{ over a universe } U \text{ such that } |A| > d!(k-1)^d$.
 $G = \{S_1, \dots, S_l\} \subseteq A$ be an inclusion-wise maximal family of pairwise disjoint sets in A .
If $l \geq k$ then G is a sunflower with at least k petals.

G is maximal, every set $A \in A$ intersects at least one set from G i.e. $A \cap S \neq \emptyset$.

$$S = \bigcup_{i=1}^l S_i \quad |S| \leq d(k-1)$$

There is an element $v \in V$ contained in at least

$$\frac{|A|}{|S|} \geq \frac{d! (k-1)^d}{d(k-1)} = (d-1)(k-1)^{d-1}$$

sets from A . We take all sets of A containing such an element v , and construct a family A' of sets of cardinality $d-1$ by removing from each set the element v . Because $|A'| \geq (d-1)! (k-1)^{d-1}$, by the induction hypothesis A' contains a sunflower $\{S'_1, \dots, S'_r\}$ with k -petals. Then $\{S'_1 \cup \{v\}, \dots, S'_k \cup \{v\}\}$ is a sunflower with k -petals.

d -hitting set

Given a family A of sets over a universe U , where each set in the family has cardinality at most d , and a positive integer k . The objective is to decide whether there is a subset $H \subseteq U$ of size at most k .

such that H contains at least one element from each set in A .

* d -Hitting sets admits a kernel with at most $d!k^d$ sets and at most $d!k^d \cdot d^2$ elements.

Let (U, A, k) be an instance of d -hitting set and assume that A contains a sunflower $S = \{S_1, \dots, S_{k+1}\}$ of cardinality $k+1$ with core y . Then return (U', A', k') where $A' = (A \setminus S) \cup \{y\}$ is obtained from A by deleting all sets $\{S_1, \dots, S_{k+1}\}$ and by adding a new set y and $U' = \bigcup_{X \in A'} X$.

Additional Notes

* Voronoi Partitions: set of centers C , every point of P assigned to nearest neighbour in C

$$\Pi(C, \bar{C}) = \{p \in P \mid d(p, \bar{C}) \leq d(p, c)\}$$

* Greedy clustering algorithm: arbitrary point \bar{c}_1 into C , for every point $p \in P$ compute $d_{\bar{c}_1}(p)$ from \bar{c}_1 . Pick point \bar{c}_2 with highest distance from \bar{c}_1 . Add this to the set of centers and denote this expanded set of centers as C_2 .

overall algorithm = $O(nk)$

→ This algorithm is 2-approx.

Proof: Case-1 Every cluster of C_{opt} contains exactly one point of k .

$$p \in P$$

$\bar{c} =$ center p belongs in C_{opt}

$\bar{k} =$ center of k that is in $\Pi(C_{opt}, \bar{c})$

$$d(p, \bar{c}) = d(p, C_{opt}) \leq r_{\infty}^{opt}(p, k)$$

$$d(\bar{k}, \bar{c}) = d(\bar{k}, C_{opt}) \leq r_{\infty}^{opt}$$

$$d(p, \bar{k}) \leq d(p, \bar{c}) + d(\bar{c}, \bar{k}) \leq 2r_{\infty}^{opt}$$

Case-2: Two centers \bar{k} and \bar{v} of k both in $\Pi(C_{opt}, \bar{c})$

σ was added later

$$r_{\infty}^k(p) \leq r_{\infty}^{C_{i-1}}(p) = d(\bar{v}, C_{i-1})$$

$$\leq d(\bar{v}, \bar{k})$$

$$\leq d(\bar{v}, \bar{c}) + d(\bar{c}, \bar{k}) \leq 2r_{\infty}^{opt}$$

* A set $S \subseteq P$ is a σ -net for P if the following two properties hold :
 (i) Covering property = All the points of P are in distance at most σ from the points of S .
 (ii) Separation property = for any pair of points $p, q \in S$
 $d(p, q) \geq \sigma$.

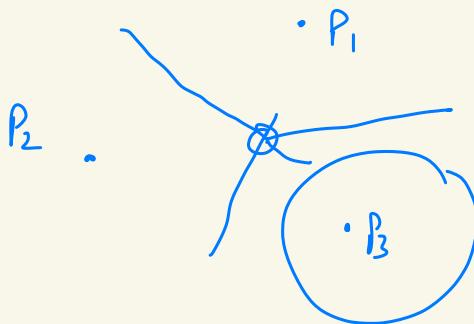
* Let P be a set of n -points in a finite metric space, and let its greedy permutation be $\langle \bar{c}_1, \dots, \bar{c}_n \rangle$ with the associated sequence of radii $\langle \bar{\sigma}_1, \dots, \bar{\sigma}_n \rangle$ for any i , $C_i = \langle \bar{c}_1, \dots, \bar{c}_i \rangle$ is a σ_i -net of P .

* $0 < p < 2 \quad \|x\|_p \geq \|x\|_2$

$$\|x\|_p \leq \sqrt{n} \|x\|_2 \quad \text{and} \quad \|x\|_2$$

Nearest Neighbour Search

- Let P be a set of n points in \mathbb{R}^d . Pre-process the pts in P .
- s.t. given a query pt. q , we can determine efficiently the closest pt to q in P .



worst possible running time $O(n^2)$
 Voronoi diagram
 $O(n \log n)$
 (through a complex algo)

- Voronoi Diagram $O(n \log n)$
- Apply point location query roughly takes $n^{[d/2]}$ time.

Approximate NN

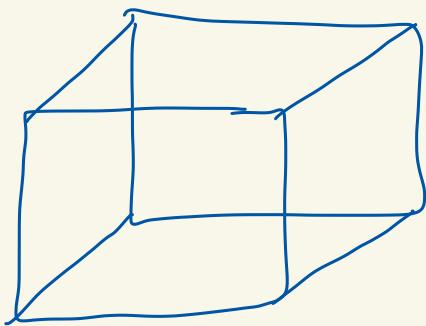
- Specify a parameter $\epsilon > 0$

Build a data structure that ans. $(1+\epsilon)$ -Apx NN (ANN)

Input: P in \mathbb{R}^d ; for a query pt. q $nn(q) = \underbrace{nn(q, P)}$

$d(q, P)$ = distance of q to the closest point in P

$||q - s|| \leq (1 + \epsilon) d(q, P) \rightarrow$ Find s .



$\text{Spread} = L$ is bounded
Unit hypercube in \mathbb{R}^d

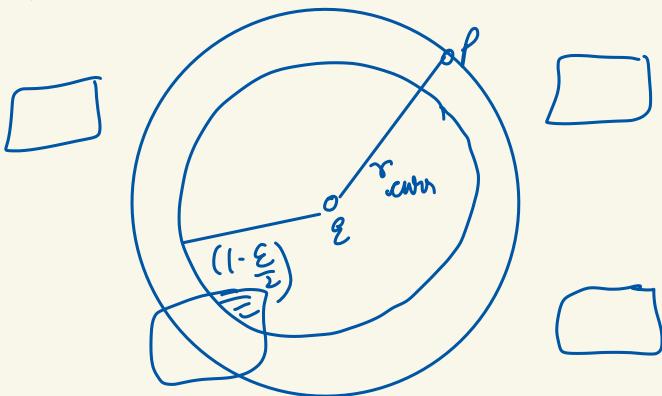
T is a quadtree of the space

$$\text{diam}(P) = \sqrt{1}$$

T

Assume for each (internal) node m in T a representative (rep w) is one of its of P stored that rooted at v .

- Idea:
1. Maintain a set of nodes of T that might contain ANN of q .
 2. Each node has a representative (compute the distance to the query q)
 3. At each (i), - Θ search gets refined by replacing a node by its children.



$$\|q - \text{rep}_w\| - \text{diam}(\square_w) > (1 - \frac{\varepsilon}{2}) * r_{\text{center}} \Rightarrow \text{Abort}$$

If not then keep doing

Alg:

Let $A_0 = \{\text{root}(z)\}$ $r_{\text{curr}} = \|q - \text{rep}(\text{root}(z))\|$

- In the i^{th} iteration

for $i > 0$ Alg expands the nodes of A_{i-1}

to get to A_i

(if cond. not satisfied)

add w to A_i

Continue until all elements of A_{i-1} are considered

stop when A_i is empty.

Correctness:

Alg adds a node w to A_i

only if P_w misnot contain pts which are close to Σ
then current best.

- say w is the last node
inspected by P Alg s.t. $\text{nn}(q) \in P_w$

if it throws away :

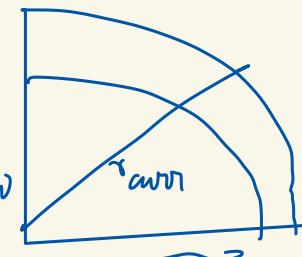
$$\|q - \text{nn}(q)\| \geq \|q - \text{rep}_w\| - \underbrace{\|\text{rep}_w - \text{nn}(q)\|}_{\geq \dfrac{1-\epsilon}{2} r_{\text{curr}}}$$

$$\geq \|q - \text{rep}_w\| - \underbrace{\text{diam}_w(\Delta_w)}_{\geq \left(1 - \frac{\epsilon}{2}\right) r_{\text{curr}}}$$

$$\text{so, } \|q - \text{nn}(q)\| / \left(1 - \frac{\epsilon}{2}\right) \geq r_{\text{curr}}$$

$$\left(1 - \frac{\epsilon}{2}\right) \leq (1 + \epsilon) \quad r_{\text{curr}} \leq (1 + \epsilon) \times d(q, P)$$

$$\geq \left(1 - \frac{\epsilon}{2}\right) r_{\text{curr}}$$



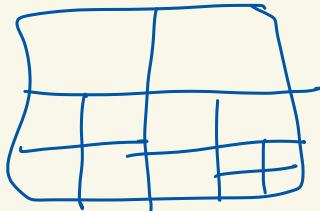
Running Time : Algorithm visits \mathcal{T} level by level

As long as level's grid cells are bigger than ANN
(# nodes visited so far)

$$\text{running time} = O\left(\varepsilon^{-1} + \log\left(\frac{1}{w}\right)\right)$$

↓
prob. to spread.

Compressed Quadtree



Alg handled nodes level by level
(keep a heap of integers in the range of
 $O(\log \phi(P))$, ..., $\log(\phi(P))$)

$$O(\varepsilon^{-d} + \log\left(\frac{\text{diam}(\mathcal{T})}{w}\right))$$

General unbounded case:

- Get rough abx
- Apply previous arguments

Embedding Finite Metric Spaces into Normed Spaces (Metric Embedding)

Recap: Metric space is a pair (X, d) , X is a set
 $d: X \times X \rightarrow [0, \infty)$

is a metric if the following happens:

$$(I) \quad d(x, y) = 0 \quad \text{if } x=y$$

$$(II) \quad d(x, y) = d(y, x)$$

$$(III) \quad \forall x, y \quad d(x, y) + d(y, z) \geq d(x, z) \quad (\text{Triangle inequality})$$

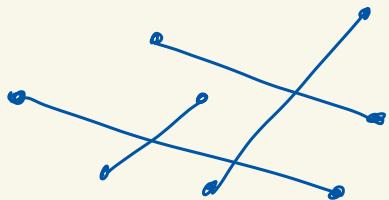
A metric on m points ($m \times n$) matrix $\binom{m}{2}$

Applications

Microbiology — X is a collection of bacterial strains.

"Nice" Representation

P in \mathbb{R}^2



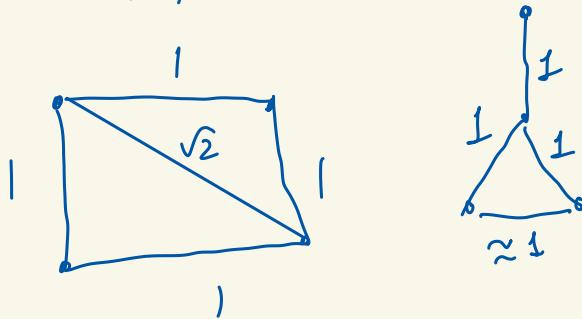
assign $x \in X$ to a point $f(x)$ in \mathbb{R}^2 s.t.,
 $f(x, y) \approx \|f(x) - f(y)\|_2$

Identify clusters, isolated entries and so on.

Geometry helps to get faster algo.

Too good to be true.

$X = \{P_1, P_2, P_3, P_4\} \rightarrow$ isometric space



Isometric Embedding

A mapping $f: X \rightarrow Y$

$X \rightarrow$ metric space with a metric P
 $Y \rightarrow$ metric space with a metric σ .

* is isometric if it preserves distances

$$\sigma(f(x), f(y)) = p(x, y) \quad \forall x, y \in X$$

Typically small "errors" (distribution) in embedding is ok

D embedding -

A mapping $f: X \rightarrow Y$ is called a D-embedding for

$$D > 1$$

real numbers

$$\text{if } \exists \tau > 0$$

$$\text{s.t. } \forall x, y \in X$$

$$\tau \times p(x, y) \leq \sigma(f(x), f(y))$$

$$\leq D \times \tau \times \sigma(x, y)$$

The infimum of number D s.t. f is a D-embedding - distortion

If X is a Euclidean space (in a normed space generally).
We can scale up/down.

Mapping with bounded D, were called bi-Lipschitz

Distortion can be defined in terms of Lipschitz constants of f and
inverse of f (f^{-1})

$$\|f\|_{\text{lip}} = \sup \left\{ \frac{\sigma(f(x), f(y))}{p(x, y)} \mid \forall x, y \in X, x \neq y \right\}$$

distortion on f

$$\|f\|_{\text{lip}} \times \|f^{-1}\|_{\text{lip}}$$

$$\ell_p\text{-norm} \quad pt \quad x \in \mathbb{R}^d \quad p \in [1, \infty)$$

$$\|x\|_p = \left(\sum_{i=1}^d |x_i|^p \right)^{1/p}$$

Qb Does \exists some euclidean space into which a given metric space embeds isometrically

$$x = (x_1, \dots) \quad \text{of reals} \quad \|x\|_p < \infty$$

$$\|x\|_p = \left(\sum_{i=1}^d |x_i|^p \right)^{1/p}$$

Remark - ℓ_2 is a separable hilbert space
 \hookrightarrow has countable basis.

Known:

n dim. regular simplex has no isometric embedding in \mathbb{R}^k , $k < n$

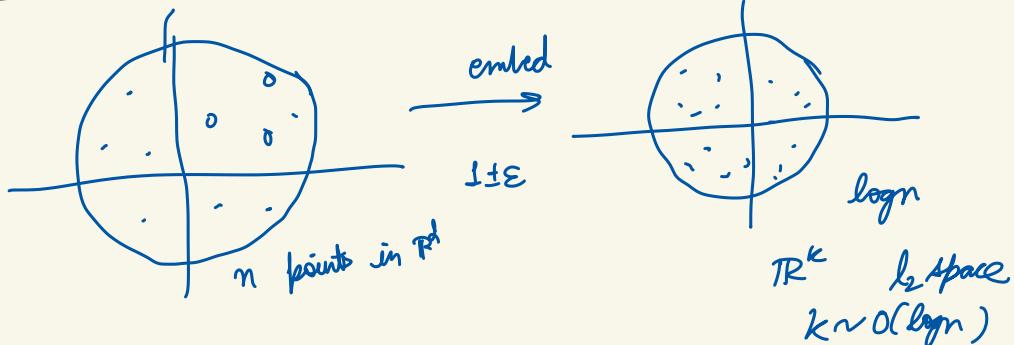
Johnson - Lindenstrauss Flattening Lemma

X be a n pt. set in \mathbb{R}^d $\epsilon \in (0, 1)$ \exists a $(1+\epsilon)$ embedding of X into ℓ_2^k $k \approx \Theta(\log n)$

Recap:

- Metric Embedding
- Isometric embedding
- Ex: isometric can't be preserved even for simple cases
- D-embedding

Johnson-Lindenstrauss Lemma



Lemma: given any $\epsilon \in (0, 1]$ and a set $X \subset \mathbb{R}^d$ with $|X| = n$
 \exists a linear map $f: \mathbb{R}^d \rightarrow \mathbb{R}^m$ with $m = O\left(\frac{\log n}{\epsilon^2}\right)$ that
extends (X, d_2) to (\mathbb{R}^m, d_2) with distortion ($\leq \epsilon$)

Idea: Use randomness to define the map f
 $A \in \mathbb{R}^{m \times d}$ be a random matrix with entries
chosen i.i.d from Gaussian.

$$f(x) = Ax$$

Hope - this exists with high probability

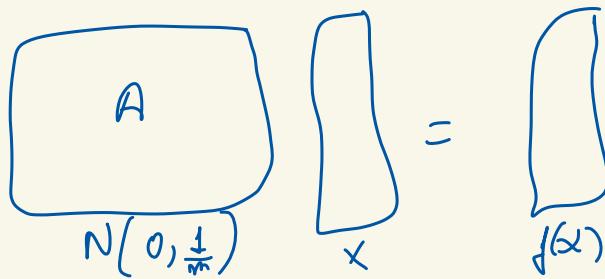
Goal - Identify such a f

$$A \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} f(x) \end{bmatrix}$$

$N(0, \frac{1}{m})$

Geometry: If the rows of A were orthogonal then \exists orthogonal projection onto the row span of A .

Pick a random subspace given by the row span of A .
Project x onto that subspace.



Proof: Multiplying A preserves the ℓ_2 distances

Fix $x, y \in \mathbb{R}^d$ show with high probability P

$$\|A(x-y)\|_2 = (\pm \varepsilon) \|x-y\|_2$$

$$\|f(x)-f(y)\|_2$$

Then we will take the union bound of all $x, y \in X$ pairs.

Fact: Gaussian distribution = spherically symmetric

$A(x-y)$ is sort of scaled version of the first column
equivalent to show that $\|y\|_2 = 1 \pm \varepsilon$

Chernoff Style Bound

We have a set of ind. gaussian random variables

$$z_1, z_2, \dots, z_m \sim N(0, 1)$$

$$\Pr \left[\sum_i z_i^2 > (1+\varepsilon)m \right] \leq \underbrace{e^{-m\varepsilon^2/8}}_{\text{Small}}$$

Let's assume this happens.

$$\|y\|_2 = 1 \pm \varepsilon \text{ with high } p$$

y is a random vector with entries $\sim N(0, \frac{1}{m})$

$$\begin{aligned} \Pr \left[\|y\|_2 \geq 1 + \varepsilon \right] &= \Pr \left[\|y\|_2^2 \geq (1 + \varepsilon)^2 \right] \\ &\leq \Pr \left[\|y\|_2^2 \geq 1 + \varepsilon \right] \end{aligned}$$

def. of ℓ_2 -norm + eqn. 1

$$\Pr \left[\sum_i z_i^2 \geq m(1 + \varepsilon) \right] \leq e^{-m\varepsilon^2/8}$$

$$\Pr \left[\|Ax(x-y)\|_2 \geq (1 + \varepsilon) \|x-y\|_2 \right] \leq e^{-m\varepsilon^2/8}$$

Union bound $\forall x, y \in X \approx O(n^2)$

$$\Pr \left[\exists x, y \in X \text{ s.t. } \|Ax(x-y)\|_2 \geq ((1 + \varepsilon) \|x-y\|_2) \right] \leq n^2 \times e^{-m\varepsilon^2/8}$$

$$\text{Q: what is } m? \quad O\left(\frac{\log n}{\varepsilon^2}\right) \geq \frac{1}{\text{poly}(n)}$$

Fast JL transformation

- matrix multiplication is slow
- Via FFT (fast fourier transformation) we can speed up.

Motivation

Bring any metric to L_2

Bourgain's Embedding

Transformation from other metric to Euclidean.

(Bourgain 1985) [Linial, London, Rabani and BG'S]

Given any metric space (X, d) with $|X| = n$

\exists an embedding of (X, d) into (\mathbb{R}^k, d_1) where

$$k \approx O(\log^2 n)$$

$$\text{Distortion} \sim O(\log n)$$

$d_1 - L_1$ metric dist.
works for any L_p for
 $p \geq 1$

Remark:

distortion is tight because some metrics require $\Omega(\log n)$. Sometimes, it's possible to improve the dim.

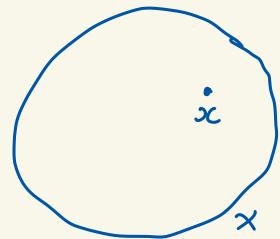
Algorithm:

- For $i = 1, 2, \dots, \log n \xrightarrow{\text{const. (will decide later)}}$
- For $j = 1, 2, \dots, c \log n$
- choose a set $S_{i,j} \subseteq X$ at random such that every $x \in X$ is contained in $S_{i,j}$ with probability 2^{-i} (independently)

Embedding (defⁿ):

$\forall x \in X, f(x) = (\underbrace{d(x, S_{1,1})}_{\text{minimum distance}}, d(x, S_{1,2}), \dots, d(x, S_{\log n, c \log n}))$

between x and any point of the set.



Q: Why this is good?

- (i) Non-expansive
- (ii) Non-shrinking

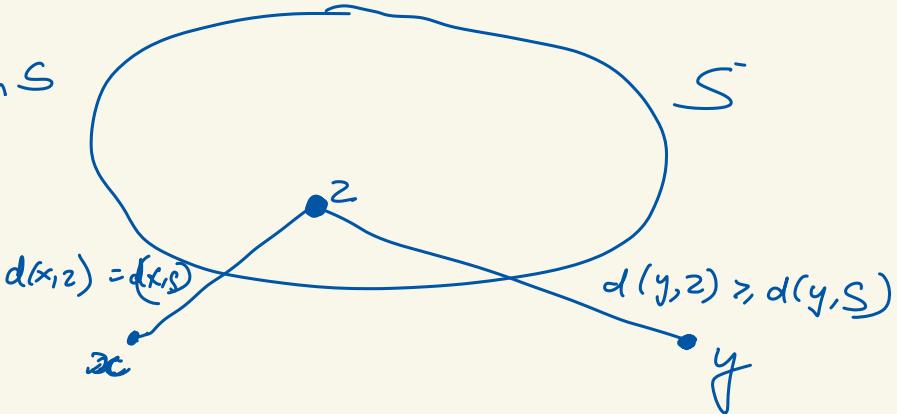
Non-expansive

This map is non-expansive
 $\forall x, y \in X$

non-expansive

$$\|f(x) - f(y)\| \leq d(x, y)$$

z is the closest point to x in S

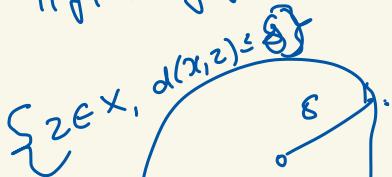


$$\begin{aligned}
 \|f(x) - f(y)\| &= d(y, S) - d(z, S) \\
 &\leq d(y, z) - d(x, z) \\
 &\leq d(x, y)
 \end{aligned}$$

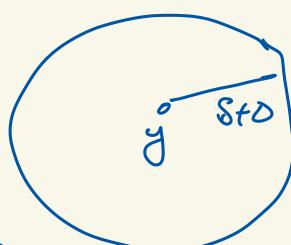
Triangle Inequality

Non-shrinking

$$\|f(x) - f(y)\| \geq \delta$$



bounding distortion above and below



$$\{z' \in X; d(y, z') \leq \delta + \epsilon\}$$

"nice instant"

Non-expansiveness

$$d(y, S) - d(x, S) > (\delta + \Delta) - \delta = \Delta$$

Idea: Pick a lot of random S and our hope is that

"nice instance" happens enough number of times.
(inner loop) — this happens if S has enough intensity.

- Good choice of S depends on $d(x, y)$ (outer loop)
it varies the size of S

$$\underbrace{k d(x, y)}_{\text{log } n} \stackrel{\text{ineq-2}}{\leq} \|f(x) - f(y)\| \stackrel{\text{ineq-1}}{\leq} k \times d(x, y)$$

$$\underbrace{\|f(x) - f(y)\|}_{\text{ineq-1}} \leq k \times d(x, y)$$

enough to show $|d(x, S) - d(y, S)| \leq d(x, y)$ for every S .
 $\Rightarrow d(x, S)$ is non-expanding

$$\|f(x) - f(y)\|_1 = \sum_{i,j} d(x, s_{ij}) - d(y, s_{ij}) \leq k \times d(x, y)$$

$$\underbrace{k \times d(x, y)}_{\text{log } n} \leq \|f(x) - f(y)\|_1$$

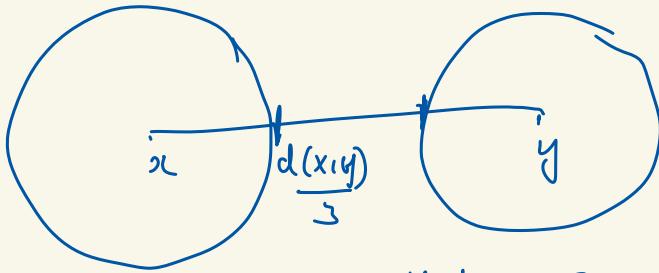
choose bunch of different deltas (S 's) A "nice" thing holds with good prob.

$$\text{fix } x, y = 0$$

$$\text{choose } 0 = \delta_0 < \delta_1 < \delta_2 < \dots < \delta_t$$

s.t. δ_i : δ_i is the smallest S

$B(x, \delta_i) \& B(y, \delta_i)$ contain $> 2^i$ pts.



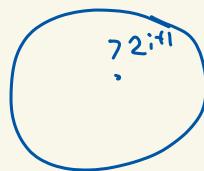
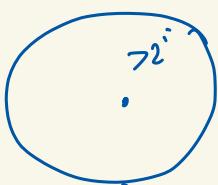
Stop at the tangent k so that $s_t < \frac{d(x,y)}{3}$

$$s_{t+1} = \frac{d(x,y)}{3}$$

nice instance holds $|d(x,s) - d(y,s)| > s_{t+1} - s_t$

Claim: It is very likely that this situation happens.

By def^m of $s_i > 2^i$



$$P[x \in S_{i,j}] = 2^{-i}$$

$$\begin{aligned} P[\text{nice event happens}] &= P[\text{non-empty intersection happens}] \times P[\text{disjointness happens}] \\ &\geq \left(1 - \left(1 - \frac{1}{2^i}\right)^{2^i}\right) \end{aligned}$$

$$\begin{aligned} &\geq \left(1 - \frac{1}{e}\right) \times \left(1 - \frac{1}{2}\right)^{2^i} \\ &\geq \frac{1}{2^{\frac{1}{2}}}, \end{aligned}$$

Claim: Fix i
 $\Pr \left[\sum_{j=1}^{c \log n} \text{function of } S_{i,j} \text{ have nice event} \right]$

1 - $\Pr \left[\sum_{j=1}^{c \log n} \text{if } \left\{ \text{nice event} \right\} \leq \frac{1}{2} \times \frac{c \log n}{2^6} \right]$

Invoke Chernoff bound

$\geq 1 - \exp \left(-\frac{c \log n}{8 \times 2^6} \right) \geq 1 - \frac{1}{n^3}$ if we choose $c \geq 3 \times 2^6$

Putting these together
By a union bound $\approx n^2$ fair and $c \log n$ choices of j
with high prob. $\forall i \forall x, y \geq \frac{c \times \log n}{2^6}$

$$\begin{aligned} & |d(x, \underset{S_{i,j}}{\cancel{s_{ij}}}) - d(y, \underset{S_{i,j}}{s_{ij}})| \geq s_{i+1} - s_i \\ & ||f(x) - f(y)|| \leq \sum_{i,j} |d(x, \underset{S_{i,j}}{s_{ij}}) - d(y, \underset{S_{i,j}}{s_{ij}})| \\ & \geq \sum_{i,j} \left(\frac{c \log n}{2^6} \right) (s_{i+1} - s_i) \\ & = \frac{c \log n}{2^6} \times s_{i+1} = \frac{c \log n}{3 \times 2^6} \times d(x, y) \end{aligned}$$

$$k = c \log n \geq \frac{k \times d(x, y)}{3 \times 2^6 \times \log n}$$

non-shrinking

Approximate Nearest Neighbour Search in Low Dimensions

- * Voronoi Diagram = $O(n^{d(d)/2})$
- * $(1+\epsilon)$ -approximate neighbours if $\|q-s\| \leq (1+\epsilon)d(q, P)$
- Alternatively $\forall t \in P \quad \|q-s\| \leq (1+\epsilon) \|q-t\|$
- * Spread of a point set $P = \phi(P) = \text{ratio of diameter and the distance of the closest pair of } P.$

* Bounded spread case

Algorithm: $A_0 = \{\text{root}(\tau)\}$ $\tau_{\text{curr}} = \|q - \text{rep}_{\text{root}(\tau)}\|$

i-th iteration $\rightarrow \tau > 0 \quad A_{i-1} \rightarrow A_i$
 $v \in A_{i-1} \Rightarrow C_v = \text{set of children of } v \in \bigcup_j \square_j$

$w \in C_v \quad \tau_{\text{curr}} \leftarrow \min(\tau_{\text{curr}}, \|q - \text{rep}_w\|)$ quadrants of P

Algorithm checks if $\|q - \text{rep}_w\| - \text{diam}(\square_w) < (1 - \frac{\epsilon}{2}) \tau_{\text{curr}}$
 if so add w to A_i :

$$\begin{aligned} \text{Correctness: } \|q - \text{nn}(q)\| &\geq \|q - \text{rep}_w\| - \|\text{rep}_w - \text{nn}(q)\| \\ &\geq \|q - \text{rep}_w\| - \text{diam}(\square_w) \\ &\geq \left(1 - \frac{c}{2}\right) \tau_{\text{curr}} \end{aligned}$$

$$\frac{\|q - \text{nn}(q)\|}{\left(1 - \frac{\epsilon}{2}\right)} \geq \tau_{\text{curr}} \quad \frac{1}{1 - \frac{\epsilon}{2}} \leq 1 + \epsilon$$

* Let $P = \text{set of } n \text{ points contained inside the unit hypercube in } \mathbb{R}^d$ and let $\mathcal{T} = \text{quadtree of } P$, where $\text{diam}(P) = 2(1)$.
 $q = \text{query point}$, $\varepsilon > 0 = \text{parameters } (1+\varepsilon)-\text{ANN to } \varepsilon$
 can be computed in $O(\varepsilon^{-d} + \log(\frac{1}{\varepsilon}))$ time $\bar{w} = d(q, P)$

Proof: If $w \in \mathcal{T}$ = considered by the algorithm and
 $\text{diam}(\Delta w) < (\frac{\varepsilon}{n}) \bar{w}$ then

$$\begin{aligned} \|q - \text{rep}_w\| - \text{diam}(\Delta w) &\geq \|q - \text{rep}_w\| - (\frac{\varepsilon}{n}) \bar{w} \\ &\geq \varepsilon_{\text{min}} - (\frac{\varepsilon}{n}) r_{\text{max}} \\ &\geq (1 - \frac{\varepsilon}{n}) r_{\text{max}} \end{aligned}$$

\Rightarrow no nodes of depth $\geq h = \lceil -\lg(\frac{\bar{w}\varepsilon}{n}) \rceil$ are considered

The number of relevant cells (i.e. cells that the algorithm explores and do not get immediately thrown out) = number of grid cells of $G_{2^{-h-1}}$ that intersect a box of sidelength $2h$, centered at q :

$$n_i = \left(2 \left(\frac{l_i}{2^{i-1}}\right)\right)^d = \left(1 + (2^i \bar{w})^d\right)$$

$$(a+b)^d \leq (2 \max(a, b))^d \leq 2^d (a^d + b^d)$$

$$\sum_{i=0}^h n_i = O\left(\log \frac{1}{\bar{w}} + \frac{1}{\varepsilon^d}\right)$$

Dynamic Inputs

- * Online Algorithms
- * Dynamic Algorithms
- * Streaming Algorithms

Online Algorithms

I - instance
 σ - ordering

when σ_i arrives, we have to make the decision instantly

$$\langle \sigma_1, \sigma_2, \dots, \sigma_n \rangle$$

Decisions are irrevocable

measure the optimality

$$\text{Competitive ratio} = \sup \frac{\text{Alg}(\sigma)}{\text{OPT}(\sigma)}$$

Dynamic \longrightarrow e.g.: B.S.T (Binary Search Tree)

I - instance

$$\sigma = \langle \sigma_1, \dots, \sigma_m \rangle$$

insertion and deletion both are allowed.

Goal: Build data structures which can handle insertion + deletion

optimize: Update complexity

Query complexity
 optimality

Streaming

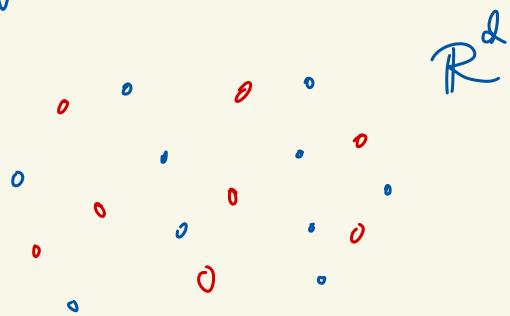
I - instance

σ - sequence

optimize: space complexity
 & Query & Time

Data Reduction (Core set)

A small subset of the input which captures all interesting features



Coreset for directional width

$|P| - |P'| = n$ pts in \mathbb{R}^d ; for any $l \in \mathbb{N}$
Projecting P onto the direction $U \in \mathbb{R}^d$ in the set

→ P onto a line l through the origin

Formally, for a vector $U \in \mathbb{R}^d$ $U \neq 0$

$$\bar{\omega}(U, P) = \max_{p \in P} \langle U, p \rangle - \min_{p \in P} \langle U, p \rangle$$

Properties: (a) Its translation invariant for any vector $s \in \mathbb{R}^d$

$$\bar{\omega}(U, s + P) = \bar{\omega}(U, P)$$

(b) Direction width scales linearly

(c) for any set $P \subseteq \mathbb{R}^d$

$$\bar{\omega}(U, P) = \bar{\omega}\left(U, \bigcup_{i=1}^k C_i(P)\right)$$

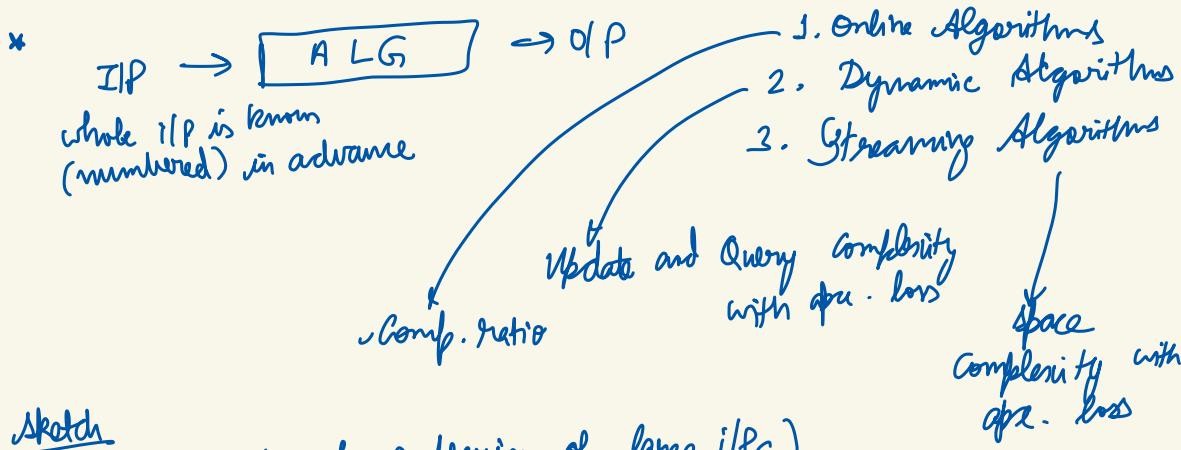
Convex Hull

(d) for any $Q \subseteq P$ and any vector U

$$\bar{\omega}(U, Q) \leq \bar{\omega}(U, P)$$

Coreset - $S \subseteq P$ is ε -coreset
 for directional width if
 sphere of directions in \mathbb{R}^d $\forall v \in S^{(d-1)}, \bar{w}(u, S) > (1-\varepsilon) \bar{w}(v, P)$
sketch: $X \subseteq Y \subseteq P \subseteq \mathbb{R}^d$
 Y is an ε -coreset (for directional width) of P
 and X is an ε' -coreset of Y . Then X is a $(\varepsilon + \varepsilon')$ coresset of P .

Merge $X \subseteq P \subseteq \mathbb{R}^d$ and $X' \subseteq P \subseteq \mathbb{R}^d$
 then $(X \cup X')$ is also a coresset of $(P \cup P')$

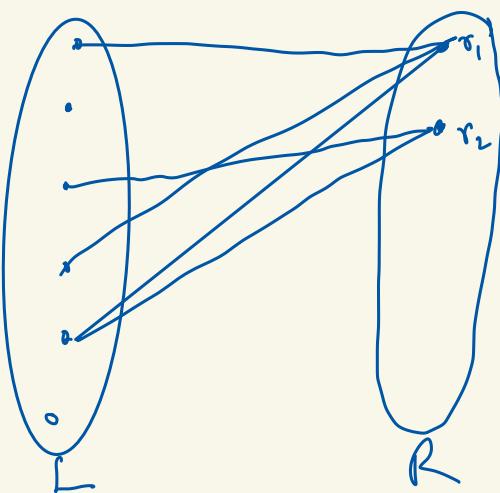


Sketch
Core set (sketch for compression of large i/Ps)

Online Algo.

I - instance
G - ordering

$\langle \sigma_1, \dots, \sigma_n \rangle$ At step i , $\sigma_i \in \{r\}$



I/P: L is known in advance

R comes one by one

a vector $v \in R$ arrives with all incident edges

- * A vertex $w \in R$ can be matched only at the time of its arrival.

Goal = maintain the maximum matching

Greedy Algo (Deterministic)

when a vertex $v \in R$ arrives look at the neighbourhood (S_v) look at the unmatched neighbours & match with an arbitrary vertex.

Thm (LB): It is not possible to get better than $\frac{1}{2}$ for any deterministic alg.

Thm: We can get $\frac{1}{2}$ by using Greedy.

Proof: via weak LP-duality.

$$L \cup R = V$$

Primal: $\max_{\mathbf{x} \in E} \sum_{e \in E} x_e$ (edge prices)

E

sub to $\sum_{e \in S(v)} x_e \leq 1$ for all $v \in L \cup R$

$x_e \geq 0$ for all $e \in E$

Dual: $\min_{v \in L \cup R} \sum_{v \in U \cup R} p_v$ (vertex prices)

sub to $p_u + p_v \geq 1 + (u, v) \in E$
 $p_v \geq 0$ & $u \in L \cup R$

for every vertex $v \in L \cup R$

$$q_v = \begin{cases} \frac{1}{2} & \text{if greedy matches } v \\ 0 & \text{otherwise} \end{cases}$$

* Comparing with the feasible solution in the dual setting -

when the algorithm adds (v, v) to its matching $q_u, q_v, -\frac{1}{2}$

$$|M| = \sum_{v \in L \cup R} q_v$$

m -matching that greedy algorthm computes

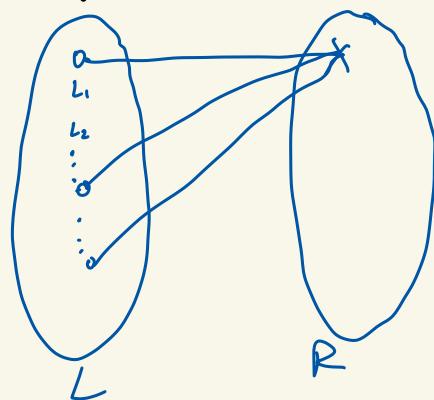
obs: Finally $(L \cup R, E)$ at least one of q_u, q_v is $\frac{1}{2}$

if not both

implies if we scale up q by a factor 2 $P = 2q$

$$|M| = \frac{1}{2} \leq P_v \geq \frac{1}{2} \times \text{opt}$$

online Bipartite Matching



I/P : left side of the bipartite matching

O/P : maximum matching maintained

$$\xrightarrow{\text{Comp. ratio}} \frac{\text{ALG}(r)}{\text{OPT}(r)}$$

Thm: There exist no deterministic online alg. with Comp. ratio better than $\frac{1}{2}$.

Thm: This is tight (e.g.: greedy algos)

Random matching

- whenever a vertex (v_i) appears in R (for any $i \in [n]$)
- Pick a random unmatched vertex (say l_j) match.

Expt. size $\frac{n}{2} + O(\log n)$

$$\text{Matrix } B_{ij} = \begin{cases} 1 & \text{if } i=j \text{ or } \begin{array}{l} \text{if } \frac{n}{2} \leq j \leq n \\ 1 \leq i \leq \frac{n}{2} \end{array} \\ 0 & \text{otherwise} \end{cases}$$

Ranking

Initialization: Pick a random permutation (say, π) of the left side.

Matching phase: When a vertex arrives (v_k) match v_k to vertex ($l_j \in L$) with highest rank.

Lower Bound claim: Thm: It is not possible to get better than

$$n\left(1 - \frac{1}{e}\right) + O(n)$$

T - $n \times n$ - complete upper-triangular matrix
Assume that columns of T arrive in the order $\langle n, n-1, \dots \rangle = \sigma$
 k^{th} column arrival - $(n-k+1)$

Def^m: $T - n \times n$
 with every permutation π on $\{1, \dots, n\}$ associated with
 $\langle T, \pi \rangle$. Let P be the uniform prob. distribution
 over $n!$ instances.

Lemma:- A deterministic algorithm "greedy" Then exp. size of the matching which A computes $\langle T, \pi \rangle$ is same as the exp. size of the matching that "random" computes.

Lemma:- The performance of some online algorithm (Alg) is upper bounded by exp. size what random achieves on $\langle T, \pi \rangle$

Brief :- For Alg A for $\langle T, \pi \rangle$ as we "random" if we have k available rows at some time t . Then there are equally likely to be $\binom{n}{k}$ rows for the first $n-t+1$ rows of T .

② for each k , the prob. that there are k eligible options at time t is same for random as it is for A .

Proof: Let $\mathbb{E}[R, \langle T, \pi \rangle]$ - exp size of matching by some randomized Alg.

$$\mathbb{E}[A] = \min_{\pi} \mathbb{E}(R, \langle T, \pi \rangle) \leq \underbrace{\max_A \{ \mathbb{E}[A(P)] \}}_{\text{Yao's min-max principle}}$$

Yao's lemma

for any randomized Alg \exists a prob. distribution over the I/P for Alg. s.t. the exp. cost of Alg on its worst case is as large as the best deterministic Alg on a random i/p from the same distribution.

Lemma :- Expected size of matching by a random alg. over T is $n\left(1 - \frac{1}{e}\right) + O(n)$

Proof : $x(t), y(t) \rightarrow$ random variables - # of columns remaining and # of rows still available at t

$$\Delta(x) = x(t+1) - x(t)$$

$$\Delta(y) = y(t+1) - y(t)$$

$$\begin{aligned} \mathbb{E}[\Delta y] &= -1 - \frac{y(t)}{x(t)} \cdot \frac{y(t)-1}{y(t)} \\ &= -1 - \frac{y(t)-1}{x(t)} \end{aligned}$$

$$\frac{\mathbb{E}[\Delta y]}{\mathbb{E}[\Delta x]} = 1 + \frac{y(t)-1}{x(t)}$$

Kartz's thm : As the prob. tends to 1 as n tends to ∞

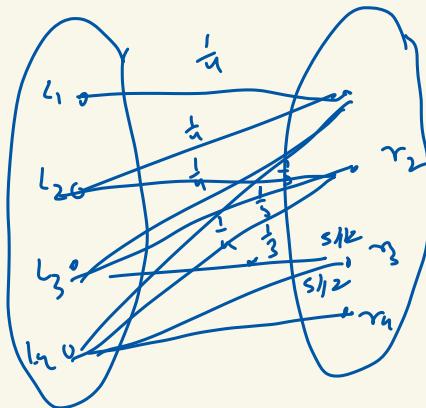
$$\frac{dy}{dx} = 1 + \frac{y-1}{x}$$

$$y = 1 + x \left(\frac{n-1}{n} - \log \frac{x}{n} \right)$$

Waterfilling Algorithm

Each vertex $v \in L$ as v contains of capacity 1.
 → when a vertex $w \in R$ arrives, it pours water of unit 1.

- Pour in the current lowest container
- Keep doing until equal



Dynamic Optimality

Binary Search Trees (BST)

- AVL
- Red - Black

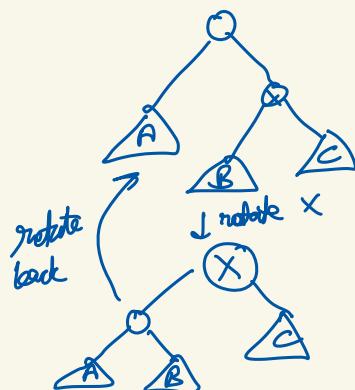
operations $\rightarrow O(\log n)$

- obj. to this the best complexity that we can hope for?

BST-model (pointer-machine model)

Data is stored as keys in BST.

- operations : left, right, parent
 - rotation



search(x)

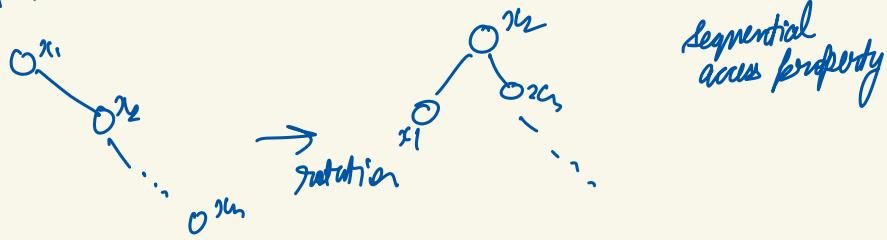
- start at root
 - must visit node with key x
- ↳ LB can't beat logn (Adversarial)

Q: Is this logn always required
 - Depends on the sequence (σ) that we have.

Keys - $\{1, 2, \dots, n\}$

Sequence σ - $\langle x_1 < x_2 < x_3 \dots < x_n \rangle$

Maintain a data structure s.t. amortized is $O(1)$

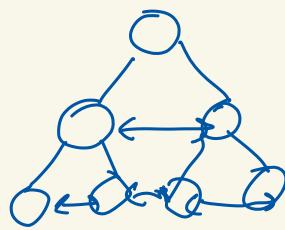


Stronger Version:

Dynamic finger property

$$|x_i - x_{i-1}| = k$$

$\Rightarrow O(\log k)$ operations



Static OPT / Entropy Bound

k appears Pn function of time

$$O\left(\sum_{i=1}^n p_i \log \frac{1}{p_i}\right)$$

amortized per operation

Working set property (roughly output sensitive)

If t_i distinct keys have been accessed so far

$$O(\log t_i) \text{ / operation}$$

Unified property

t_{ij} distinct keys accessed in x_i, \dots, x_j then x_i costs

$$O(\log \min_{i \leq j} [(x_i - x_j) + t_{ij} + 2])$$

Generalizes dynamic finger property

Conjecture - $O(1)$ complexity

Total cost of access keys = $O(OPT)$

What is OPT ? (offline) minimum over all BST Algo
over the access seq. X .

Q Does \exists online BST which $O(1)$ to the offline OPT .

$O(\log n) \rightarrow$ trivial

$O(\log \log n) \rightarrow$ Best known

Splay tree, Greedy Conj.

Primal Dual Method

Linear programming

$A \rightarrow m \times n$ matrix

$x \rightarrow n$ vector

$b \rightarrow m$ vector

$w \rightarrow m$ -vector

$$\begin{aligned} Ax &\geq b \\ x &\geq 0 \\ \min(w^T x) \end{aligned}$$

$$y^T A x \geq y^T b$$

$y \rightarrow m$ -vector

multiply i^{th} inequality

by $y_i \geq 0$

add up all the inequalities

Suppose this holds

$$y^T A \leq w^T$$

$$w^T x \geq y^T A x$$

$$\geq y^T A b$$

for any solution
 x to primal and
 y for the dual

$$\max(y^T b)$$

$$y^T A \leq w^T \Leftrightarrow A^T y \leq w$$

$$y \geq 0$$

→ dual of the linear program

Suppose (\bar{x}, \bar{y}) are solutions to the primal and dual that satisfy following conditions -

(i) if $\bar{x}_i > 0$ the dual constraint corresponding to x_i is satisfied exactly (inequality is an equality).

→ i^{th} row of A^T

(ii) if $\bar{y}_j > 0$ the primal constraint corresponding to j^{th} row in A is tight (an equality).

→ Complementary Slackness

If these conditions are satisfied then the cost of x^* and y^* are the same and both are optimal solutions for respective LPs.

$$w^T x^* = \sum_{i=1}^n w_i x_i^* = \sum_{i=1}^n w_i x_i^* = \sum_{i, x_i^* > 0} \left(\sum_{j=1}^m a_{ij} y_j^* \right) x_i^*$$

$x_i^* > 0$ because the corresponding inequality is tight

if $x_i^* > 0$
 $A^T y = w^T$

then the i^{th} row
is an equality

$$= \sum_{i, x_i^* > 0} \left(\sum_j a_{ij} y_j^* \right) x_i^*$$

$$= \sum_{y_j^* > 0} \left(\sum_{x_i^* > 0} a_{ij} x_i^* \right) y_j^*$$

$$= \sum_{y_j^* > 0} b_j y_j^*$$

second condition
of Complementary slackness

Integer Linear Programming

Linear programs with the additional restriction that the variables are required to take integer values

NP Hard

* Vertex Cover in a graph

Given an undirected graph with positive weights assigned to vertices, find a minimum subset of vertices such that every edge has at least one endpoint in the subset.

$x_i \rightarrow$ for each vertex i
 $x_i = 0$ means vertex i is not in the subset.
 $x_i = 1$ means it is in the subset.

wt. of subset = $\sum_{i=1}^n w_i x_i$

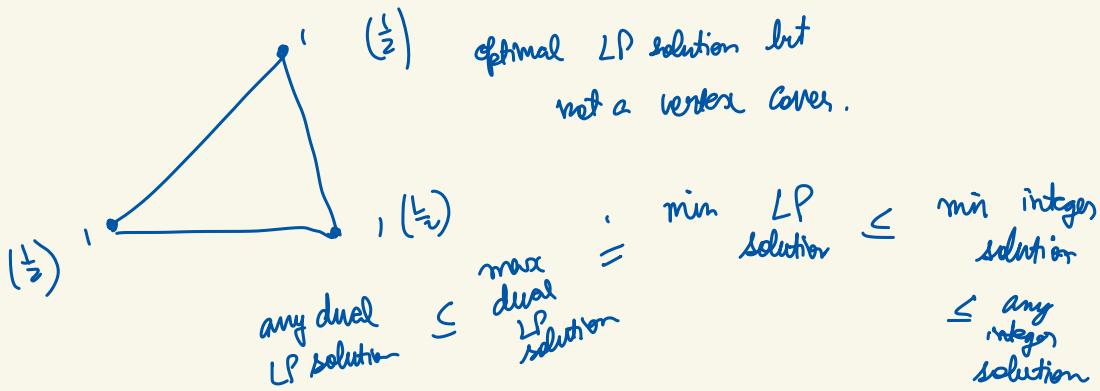
ILP formulation

$$\min \sum_{i=1}^m w_i x_i$$

for every edge $ij \quad x_i + x_j \geq 1$

for all vertices $x_i \geq 0$

LP-relaxation allows fractional values for x_i



dual vertex cover

$$\begin{aligned} & \min \sum w_i x_i \\ & x_i + x_j \geq 1 \text{ each edge} \\ & x_i \geq 0 \end{aligned}$$

y_{ij} for each edge ij

$$\begin{aligned} & y_{ij} \geq 0 \\ & \text{for every vertex } i \\ & \sum_{\substack{\text{all edges} \\ ij \text{ incident} \\ \text{with } i}} y_{ij} \leq w_i \end{aligned}$$

even this could be approximate

Approximate complementary slackness

{ if $x_i^* > 0$ then corresponding inequality is tight
if $y_i^* > 0$ then corresponding formal inequality is approximately tight.
instead of requiring equality, we allow LHS to be some constant C times RHS.

Initially start with all dual variables = 0 and $x_i = 0$

→ Pick any edge and start increasing its dual value.

↳ do this till the dual inequality corresponding to at least one of the endpoints becomes tight.

↳ Select the endpoint for which the inequality becomes tight. and
(include in vertex cover)
delete all edges incident with it.

→ Repeat until all edges are deleted.

* a vertex is selected only when the corresponding dual inequality is tight.

primal inequality is approximately satisfied with a constant factor of 2.

because for any edge $ij \quad 1 \leq x_i + x_j \leq 2$

Cost of integer solution $\leq 2 \times$ cost of dual LP solution

Set Cover

Let V and a collection of subsets of V

$$\{S_1, \dots, S_m\}$$

each set S_i has wt w_i

Find min-wt collection $C' \subseteq C$ such that every element in V is contained in some subset in C' .

$$\{e_1, \dots, e_m\}$$

$$\{S_i \mid \text{edges incident with vertex } v_i\}$$

Vertex cover = special case of set cover

Assume that every element belongs to some set in C

$x_i \rightarrow$ let $s_i = 0$ if s_i is not selected
 $= 1$ if s_i is selected.

$$\min (\sum w_i x_i)$$

for every element a in U , $\sum_{i, \text{ such that } a \in s_i} x_i \geq 1$
 (one set has to be selected).

If every element in U belongs to at most k sets
 k -approximation algorithm.

Shortest Path with positive costs on edges

think of this as a set cover problem

directed graphs \rightarrow two vertices s and t

+ve integers wts to the edges

min-cost path from s to t .

set cover

$\{x_1, \dots, x_n\}$ and a collection of subsets of $U \rightarrow$ with their union is U
 cost assigned to each subset

min wt collection of subsets whose union is U .

For edge e_i , we have a variable e_i ($\min w_i e_i$)

for every subset S of vertices that includes s but not t .

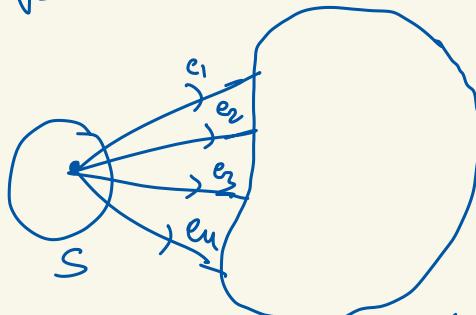
$$\sum e_i \geq 1$$

all edges e_i that join a vertex in S to a vertex not in S

\downarrow
 2^{n-1} possible constraints

$y_S \rightarrow$ increasing its value

→ till inequality
for some edge
becomes tight



↓
the min wt
edge leaving S

↓
we can include
it in the set
cover.

for an edge e the dual constraint

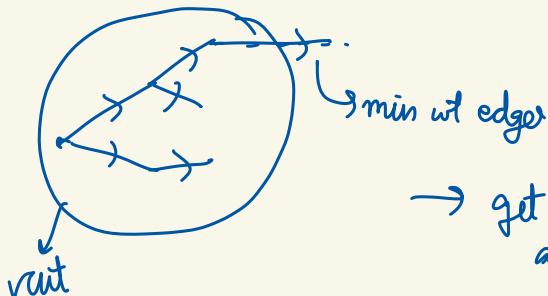
$$\sum_S y_S \leq w_e$$

such that
 e goes from
 S to S'

the weights of other edges are
reduced by w_{\min}

Consider this
as the cut

Now we will check these
edges in the next iteration



→ get a set cover as soon
as t gets added to the
tree

delete all edges not on the path from s to t .

Finally when only edge in a path are left, for every cut with a non-zero dual value, exactly one edge in the cut will be included in the solution.

If a dual variable is not zero then the corresponding primal inequality is tight.

Undirected graph with positive weights assigned to vertices. Find a min wt. subset of vertices whose removal leaves only isolated vertices or edges.

Vertex cover \rightarrow find a subset of vertices whose removal only leaves isolated vertices.

set of all edges $\vee^{\{ \text{edges incident with } v \}}$ \rightarrow Vertex cover

set of all paths of length 2 (3 vertices)

{ set of all paths of length 2 that contain v }

V

x_v

3- approximation

$\min w_v x_v$

For every path $v_1 v_2 v_3$
of length 2

$$x_{v_1} + x_{v_2} + x_{v_3} \geq 1$$



km (all weights 1)

optimal integer solution
is n-2

$$x_v = \frac{1}{3}$$

satisfies all inequalities, it is a fractional solution whose cost is $\frac{n}{3}$.
 $\frac{n}{3} \approx \frac{1}{3} (\text{Cost of OPT-integer solution})$

Integrality gap of the LP ratio between $\frac{\text{optimal integer solution}}{\text{optimal fractional solution}}$

↓
 cannot hope to prove a better bound than this for a primal dual problem.

for any subset S of 3 or more vertices that induces a connected graph

$$\sum_{v \in S} d_S(v) x_v \geq |E_S| - \frac{|S| - \tau(S)}{2}$$

↓
 no. of neighbours
 of v in S

where $\tau(S)$ denotes
 the min number of
 vertices from S to be removed
 to leave only isolated vertices.

number of
 subgraphs may
 be exponential

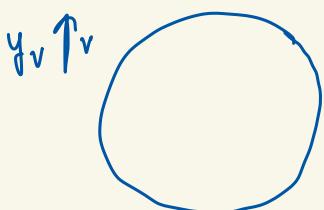
initially we have a connected graph with at least 3 vertices.

dual inequality of a vertex

$$\sum_{\substack{\text{all such} \\ \text{sets containing} \\ v}} d_S(v) y_S \leq w_v$$

Initially increase y_S for the set S of all vertices in G .

tilt it becomes tight for some vertex



↓ This one for which $\frac{w_v}{d(v)}$ is minimum.

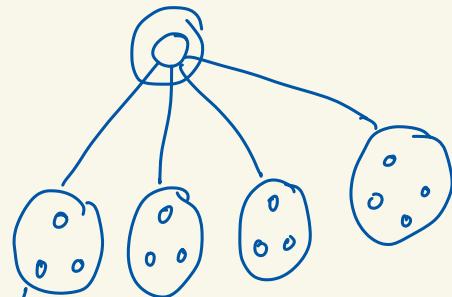
Select the vertex with minimum value $\frac{w}{\text{degree}} = S$

$$y_v = S$$

reduce the weights of all other vertices u by $S \cdot d(u)$

$$\text{new-wt of } v = \text{original-wt of } u - \frac{w(v)}{\deg(v)} \times \deg(u)$$

delete the selected vertex, all edges incident with it
and recursively work with all connected components with at least 3
vertices.

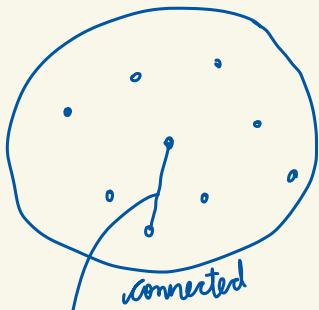


After the recursive call, check if initially selected vertex v required i.e. whether deleting it still gives a valid solⁿ and if no, delete it.

whenever a vertex is selected \Rightarrow dual inequality is tight

For any minimal subset with the required property in the graph $G(S)$ where y_S is not zero, satisfies the primal inequality within a factor of 2.

$$\sum_{x \in X} d(x) \leq 2|E| - |S| + \tau(S)$$

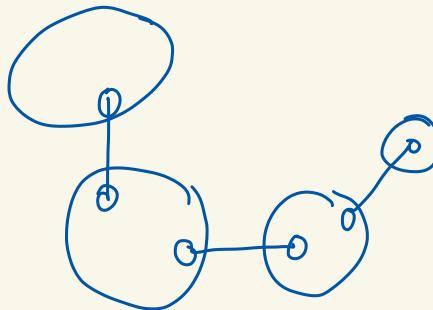


$$S = V$$

connected

Adds 2 to both LHS and RHS, hence we can reduce the graph.

$|E| \geq |S|-1$ graph is connected



Connected

k edges between vertices in X such that removing any one disconnects the graph then

$$\tau(S) \geq k+1$$

This is satisfied for any minimal subset X and connected graph G with at least 3-vertices.

* If we are allowed to leave complete subgraphs, then 3-apx can be found.

Approximate Complementary slackness

Let $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ be feasible solutions to the primal and dual linear programs, respectively, satisfying the following conditions:

- Primal complementary slackness: for $\alpha \geq 1$, for each i , $1 \leq i \leq n$, if $x_i > 0$ then $\frac{c_i}{\alpha} \leq \sum_{j=1}^m a_{ij} y_j \leq c_i$
 - Dual complementary slackness: for $\beta \geq 1$, for each j , $1 \leq j \leq m$, if $y_j > 0$ then $b_j \leq \sum_{i=1}^n a_{ij} x_i \leq b_j \cdot \beta$.
- Then $\sum_{i=1}^n c_i x_i \leq \alpha \cdot \beta \cdot \sum_{j=1}^m b_j \cdot y_j$

Greedy Algorithm for the set cover problem

Initially, $C = \emptyset$. Let U be the set of yet uncovered elements. As long as $U \neq \emptyset$, let $S \in S$ be the set that minimizes the ratio $\frac{c_S}{|U \cap S|}$:

- (i) Add S to C and set $x_S \leftarrow 1$
- (ii) for each $e_i \in (U \cap S)$, $y_{e_i} \leftarrow \frac{c_S}{|U \cap S|}$

The greedy algorithm is an $O(\log n)$ -approximation algorithm for the set-cover problem:

Proof: P, D = values of the objective functions of the primal and dual solutions produced.

- (1) Clearly if there exists a feasible primal solution, then the algorithm will also produce a feasible solution.
- (2) In a iteration $\rightarrow \Delta P = c_S$ (addition of set S)
 $\Delta D = c_S$ (we set the variables corresponding to $|U \cap S|$)
 $\frac{c_S}{|U \cap S|}$.
- (3) e_1, e_2, \dots, e_n be the elements belonging to the set S , sorted with respect to the order they were covered by the greedy algo.
 Element e_i : $y_{e_i} \leq \frac{c_S}{(k-i+1)}$. Algo chooses the set that

minimizes the ratio of cost to number of uncovered elements.
At the time, e_i is covered, sets contained at least $k-i+1$
elements still uncovered.

$$\sum_{e_i \in S} y_{e_i} \leq \sum_{i=1}^k \frac{c_s}{k-i+1} = k_k \cdot c_s = c_s \cdot O(\log n)$$

y_{e_i} = i^{th} harmonic numbers

$P \subseteq D$. By claim (3) we get that $D' = \frac{D}{O(\log n)}$
Primal is at most $O(\log n)$ times a feasible dual solution.
By weak duality, feasible dual \leq feasible Primal.
 \Rightarrow Primal solution $\geq (O(\log n))$ optimal - primal solution.

Randomized Routing Algorithm:

- (1) For each set $s \in S$, choose $2lmn$ independently random variables $x(s,i)$ uniformly at random in the interval $[0,1]$.
- (2) For each set s , let $\Theta(s) = \min_{i \in s} x(s,i)$.
- (3) Solve linear program (P) .
- (4) Take set s to the cover if $\Theta(s) \leq x_s$.

The algorithm produces a solution with the following properties:

- (1) The expected cost of the solution is $O(\log n)$ times the cost of the fractional solution.
- (2) The solution is feasible with prob. $1 - \frac{1}{n} > \frac{1}{2}$.

Proof: For each i , $1 \leq i \leq 2lmn$, the prob. that $x(s,i) \leq x_s$ is exactly x_s .

s is chosen $\rightarrow \exists i$ such that $x(s,i) \leq x_s$

A_i = event $\rightarrow x(s,i) \leq x_s$

Prob that s is chosen = $\bigcup_{i=1}^{2lmn} A_i \rightarrow$ atmost the sum of individual prob which is $2x_s lmn$.

Using linearity of expectation, expected cost of the solution is at most $2lmn$ times the cost of the fractional solution.

To prove (2), pick an element e .

$$\prod_{\substack{S \in S \\ e \in S}} (1 - x_S) \leq \exp(-\sum x_S) \leq \exp(-1)$$
$$(1 - x \leq \exp(-x))$$

Probability that e is not covered due to is the probability that none of the variables $x(S_i)$ result in choosing a set S covering e . Since we choose $2mn$ random variables, the prob. that e is not covered is at most $\exp(-2mn) = \frac{1}{n^2}$. Using union bounds, prob. that there exist e which is not covered is at most $\frac{1}{n^2} \times n = \frac{1}{n}$.

Primal-Dual Schema

while there exists an uncovered element e_i :

- (1) Increase the dual variable y_{e_i} continuously.
- (2) If there exists a set S such that $\sum_{e \in S} y_e = c_S$: take S to the cover and set $x_S \leftarrow 1$.

f = maximum number of sets an element belongs to.

The algorithm = f -approx.

Both the primal and dual constraints are satisfied.

Proof: Both the primal and dual constraints are satisfied. Primal complementary slackness condition holds with $\alpha=1$ since

if $x_S > 0$ then $\sum_{e \in S} y_{e_i} = c_S$.

Dual complementary slackness holds with $\beta=f$ since if $y_{e_i} > 0$

$1 \leq \sum_{S \ni e_i} x_S \leq f \Rightarrow$ Use prev. thm.

For vertex cover, $f=2n$.

Ski-Rental Problem

Rent $\rightarrow \$1$ per day

Unknown = number of skiing days remaining.

Buying $\rightarrow \$B$

$x \rightarrow 1$ if the skier buys the skis
For each day j , $1 \leq j \leq k \rightarrow z_j \rightarrow 1$ if the skier rents on day j

$$x + z_j \geq 1$$

$$x \in \{0, 1\}$$

$$z_j \in \{0, 1\}$$

Dual: $y_j \leq 1 \quad \sum_{j=1}^k y_j \leq B \quad \text{max } \sum_{j=1}^k y_j$

optimal Deterministic Algorithm:

rent the skis for the first $B-1$ days (2-competitive,
Buy them on the B^{th} day.)

Dual-dual: On the j^{th} day, a new primal constraint $x + z_j \geq 1$ dual variable y_j

Primal is already satisfied then do nothing.
otherwise increase y_j continuously, until some dual constraint becomes tight, set the corresponding primal variable to 1.

If $y_j > 0$ then $1 \leq x + z_j \leq 2$
 if $x > 0$ then $\sum_{j=1}^k y_j = B$
 if $z_j > 0$ then $y_j = 1$

} Apply Thm.

Algorithm.

(1) Initially $x \leftarrow 0$ (j^{th} new constraint), if $x < 1$:

(2) Each new day (j^{th} new constraint), if $x < 1$:

$$(a) z_j \leftarrow 1-x$$

$$(b) x \leftarrow x(1+\frac{1}{B}) + \frac{1}{c \cdot B}$$

$$(c) y_j \leftarrow 1$$

Number of ski-days = k
 Since $z_j = 1 - x \rightarrow$ primal feasible
 Increments in $x = z_1, \dots, z_k$ $x = \sum_{j=1}^k z_j$

$$x_i = \frac{1}{CB} \quad q = 1 + \underbrace{\left(1 + \frac{1}{C}\right)^B}_{C} \quad \text{geometric}$$

$$x = \left(1 + \frac{1}{C}\right)^B - 1 \quad \Rightarrow \text{set} = \left(1 + \frac{1}{C}\right)^B - 1$$

If $x < 1$, dual objective function increases by k
 primal $\Delta P = Bx + z_j = z_j + \frac{1}{C} + 1 - x = \underbrace{1 + \frac{1}{C}}_{\text{I}}$ } ratio
 for $B \gg 1$ $1 + \frac{1}{C} \approx \frac{C}{e-1}$ use weak-duality
 Then, algo is $\underbrace{1 + \frac{1}{C}}_{\text{competitive}}$.

* We need to convert the fractional solⁿ to randomized competitive algorithm. We arrange the increments of x on the $(0,1]$ interval and choose $\alpha \in [0,1]$ (before) uniformly in random. We are going to buy skis on the day corresponding to the increment of x to which α belongs.

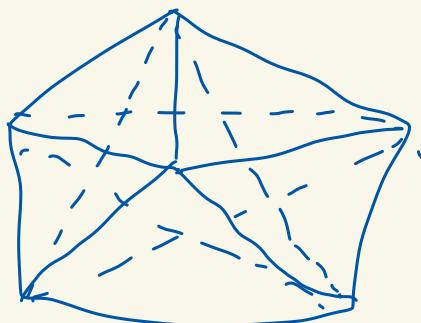
Prob. of buying skis on the j^{th} day = x_j
 Expected cost = $B \cdot \sum_{j=1}^k x_j = Bx$

$$z_j = 1 - \sum_{i=1}^{j-1} x_i > 1 - \underbrace{\sum_{i=1}^k x_i}_{\text{Prob. of not buying skis}}$$

Prob of getting is at most z_j .
 By linearity of expectation, for any numbers of ski days, the expected cost of the randomized algo is at most the cost of the fractional solution.

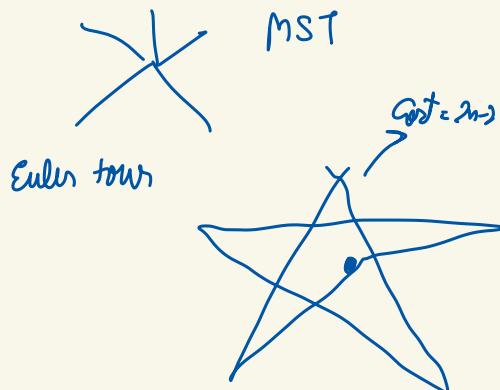
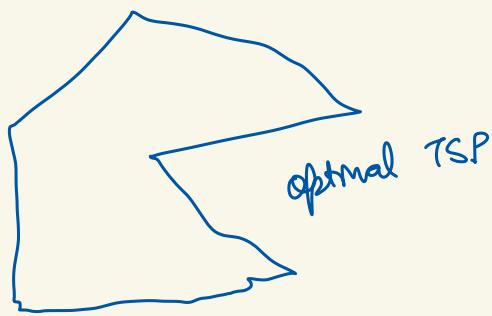
Tight Example of Eulerian Metric TSP Algo (2x MST)

Complete graph on n vertices with edges of cost 1 and 2



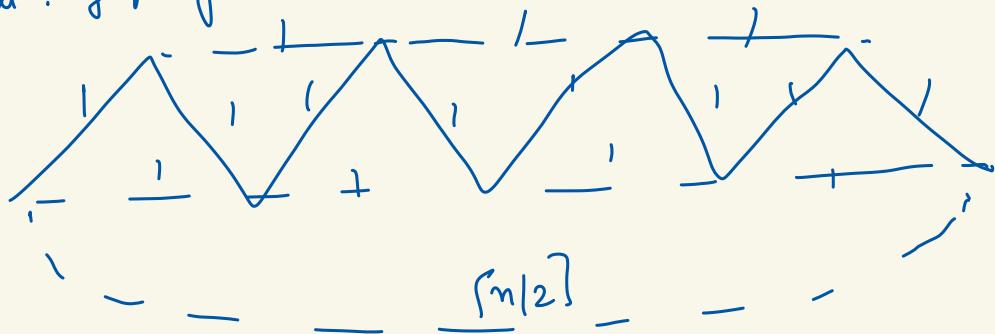
Thick edges \rightarrow cost = 1
Remaining \rightarrow cost = 2

For arbitrary n ,
graph has $2n-2$ edges of
cost 1, with these edges
forming union of a star,
and an $n-1$ cycle, all others
have cost 2.



Tight Example of Christofides Algorithm

n odd : graph of n vertices



MST = thick edge

$$\text{Christofides} = (n-1) + \lceil \frac{n}{2} \rceil$$

optimal = n

K-center clustering

- * find a set of k points, $C \subseteq P$, such that the maximum distance of a point in P to its closest point in C is minimized.

Greedy k centers

arbitrary point \bar{c}_1 served by $C_1 \rightarrow r_1 = \max_{p \in P} d_1(p)$ Add the point which is worst

$O(nk)$ time
 $\Theta(n)$ space

This algo is 2-approx to optimal k-center clustering.

$$r_k = \|p_k\|_\infty$$

$$\bar{c}_{k+1} = \text{point in } P \text{ realizing } r_k = \max_{p \in P} d(p, C) \quad C = k \cup \{\bar{c}_{k+1}\}$$

$$r_1 > \dots > r_k$$

$$d_m(\bar{c}_i, \bar{c}_j) > d_m(\bar{c}_j, c_{j-1}) = r_{j-1} > r_k$$

$$; c_j \leq k+1$$

distance between any pair of points in C is at least r_k .

assume $r_k > 2 \text{opt}_{\infty}(P, k)$

optimal solution that covers P with k balls of radius opt_{∞} .

By Δ -inq, any two points are within a distance of at most 2opt_{∞} .
thus none of these balls can cover two such points of $C \subseteq P$.
The optimal covering by k balls of radius opt_{∞} cannot cover C (and thus P) as $|C| = k+1 \rightarrow$ contradiction.

* Spread = Ratio between diameter and the distance of the closest pair of P .

ANN for bounded spread

* Let $A_0 = \{\text{root}(\mathcal{T})\}$ $r_{\text{curr}} = ||q - \text{rep}_{\text{root}}(T)||$
 In the i^{th} iteration, for $i > 0$, expand the nodes of A_{i-1} to get A_i
 for $v \in A_{i-1}$, $C_v = \text{set of children of } v \in \mathcal{T}$
 $\square_v = \text{denote the cell}$

for every node $w \in C_v$:
 $r_{\text{curr}} + \min(r_{\text{curr}}, ||q - \text{rep}_w||)$

$$\text{if } ||q - \text{rep}_w|| - \text{diam}(\square_w) < (1 - \frac{\epsilon}{2}) r_{\text{curr}}$$

stop when $A_i = \text{empty}$

* The algorithm adds a node w to A_i only if the set P_w might contain points which are closer to q than the (best) current nearest neighbour of the algorithm. Start node w inspected by the algo such that $\text{nm}(q) \in P_w$. Algorithm decided to throw it away:
 $||q - \text{nm}(q)|| \geq ||q - \text{rep}_w|| - ||\text{rep}_w - \text{nm}(q)||$
 $\geq ||q - \text{rep}_w|| - \text{diam}(\square_w) \geq (1 - \frac{\epsilon}{2}) r_{\text{curr}}$

$$\text{Thus } \frac{||q - \text{nm}(q)||}{1 - \frac{\epsilon}{2}} \geq r_{\text{curr}} \rightarrow \frac{1}{(1 - \frac{\epsilon}{2})} \leq 1 + \epsilon$$

$$\rightarrow r_{\text{curr}} \leq (1 + \epsilon) d(q, P)$$

Lemma: Let P = set of n points inside the unit hypercube in \mathbb{R}^d \mathcal{T} = quadtree of P , where $\text{diam}(P) = \Omega(1)$. q = query point, $\epsilon > 0$
 A $(1 + \epsilon)$ ANN to q can be computed in $O(\epsilon^{-d} + \log(\frac{1}{\epsilon}))$
 time, where $\bar{w} = d(q, P)$.

If a node $w \in \mathcal{T}$ is considered by the algo and $\text{diam}(\square_w) < (\frac{\epsilon}{n}) \bar{w}$ then

$$\|q - \text{ref}_w\| - \text{diam}(\square_w) \geq \|q - \text{ref}_w\| - \left(\frac{\varepsilon}{\bar{w}}\right)\bar{w} \geq \text{curr} - \left(\frac{\varepsilon}{\bar{w}}\right)\text{curr}$$

No nodes of depth $\geq h = \lceil -\log \left(\frac{\bar{w}\varepsilon}{\bar{w}} \right) \rceil$ are being considered. $\geq (1 - \frac{\varepsilon}{\bar{w}})\text{curr}$

node w of T of depth i containing $m(q, p)$ distance between q and ref_w is at most $l_i = \bar{w} + \text{diam}_w = \bar{w} + \sqrt{d} 2^i$
at the end of i^{th} iteration, $\text{curr} \leq l_i$

The number of relevant cells (i.e. cells that the algorithm deems and do not get immediately thrown out) is the number of grid cells of $G_{2^{-i-1}}$ that intersect a box of sidelength $2l_i$ centered at q i.e.

$$n_i = \left(2 \left\lceil \frac{l_i}{2^{-i-1}} \right\rceil \right)^d = O \left((1 + (2^i \bar{w}))^d \right)$$

$$(a+b)^d \leq (2 \max(a, b))^d \leq 2^d (a^d + b^d)$$

$$\sum_{i=0}^n n_i = O \left(\log \frac{1}{\bar{w}} + \frac{1}{\varepsilon^d} \right)$$

METRIC EMBEDDING

Let (X, d_X) and (Y, d_Y) be metric spaces. A mapping $f: X \rightarrow Y$ is an embedding if it is C -lipschitz if $d_Y(f(x), f(y)) \leq C \cdot d_X(x, y)$ for all $x, y \in X$. The mapping f is called κ -bi-lipschitz if there exists a $C > 0$ such that :

$$C^{-1} d_X(x, y) \leq d_Y(f(x), f(y)) \leq C \cdot d_X(x, y)$$

$\forall x, y \in X.$

least $\kappa = \text{distortion of } f$