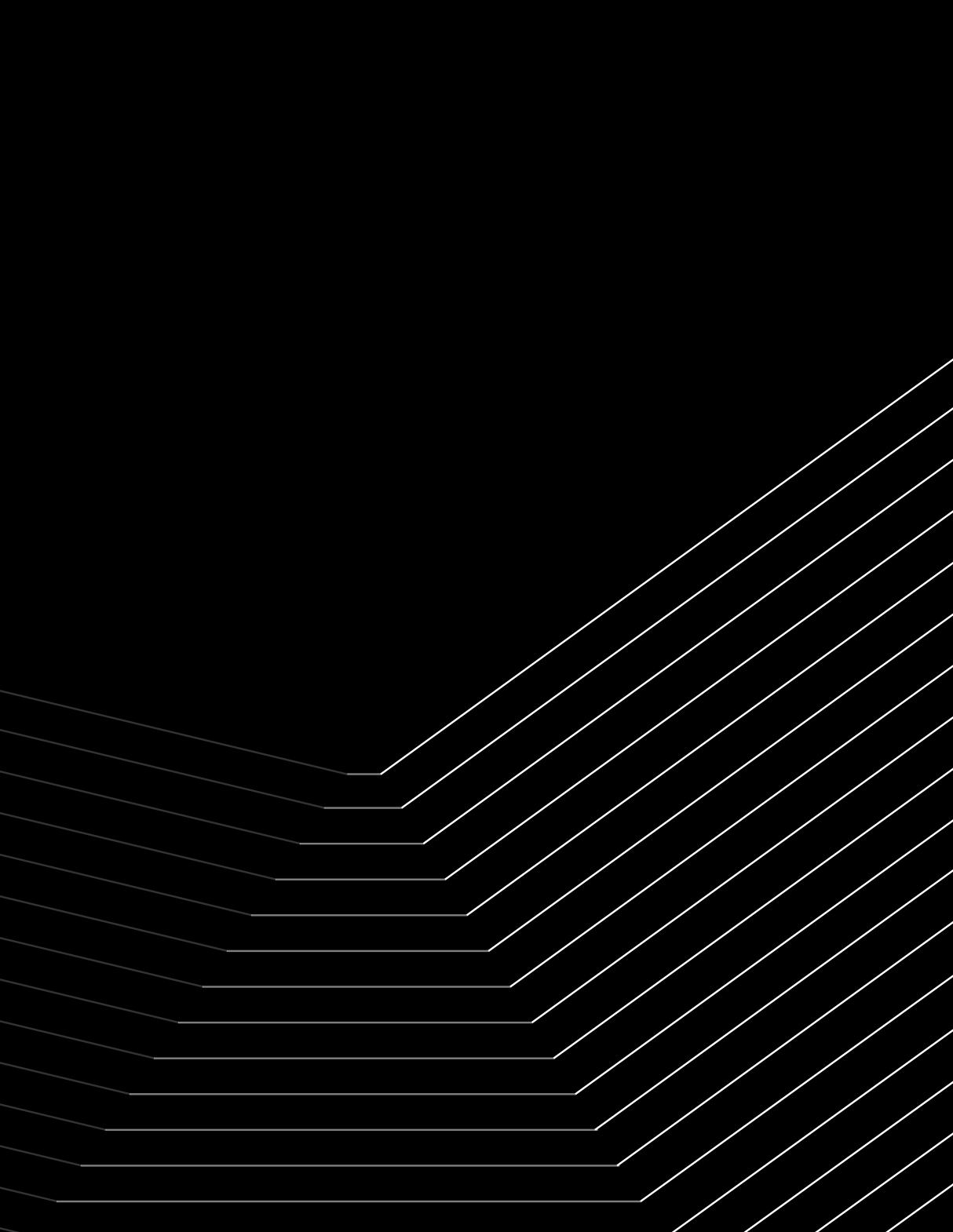
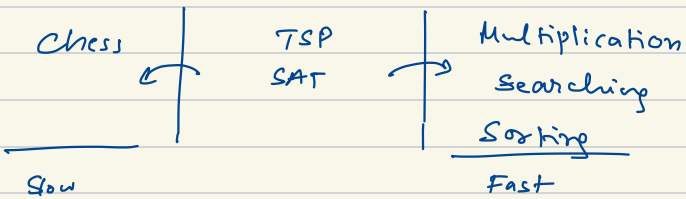


Computational Hardness

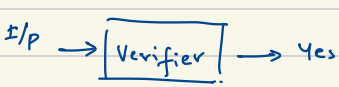
Q Given a set of locations, find the shortest possible route to visit all locations by visiting each exactly once. (TSP)

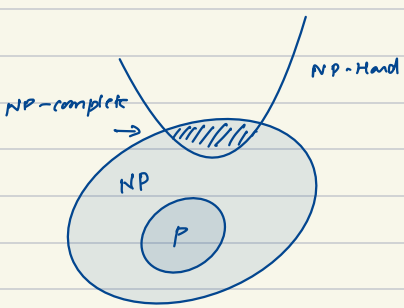


P = { Problems solvable in polynomial time by deterministic TM }

$n^{O(1)}$ time → polynomial

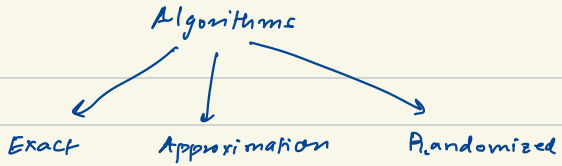
NP = { Decision problems solvable in polynomial time by non-deterministic TM }





A problem X is

- NP Complete if $X \in NP$ & X is NP-Hard
- NP Hard — A class of problems already established. If we can show a reduction to these then we call such problems NP Hard



Course Content

First Half

Approximation Algorithms: Ind. Set, Set Cover, Hitting Set, TSP, Clustering (k-means, k-median), Steiner Tree

Parametrized Algorithms: Kernelization, Color Coding, Parametrized approximation, FPT Approximation, Lossy Kernelization, Bi-dimensionality

Second Half

Dynamic Aspects: Data is not static

Paradigms:
- Online } → Reveal the input piece by piece
- Dynamic → stochastic Model where input is unknown but is drawn from some known distribution
- Streaming

Query complexity, update time of the data structure

Graphs: connectivity, reachability, apx. distance oracles

Geometry: LSH, point location, range searching.

Other: Succint DS, External memory, sketching

2/1/24

Approximation Algorithm

Design an algorithm that strictly runs in polynomial time ($n^{O(1)}$)

Output is allowed to be a "provable" factor away from the optimal soln.

Maximization Problems

Eg: Ind. Set

Variable $\alpha \geq 1$. α-approximation if we output a soln that is $(\frac{1}{\alpha})$-factor to the output

Minimization Problems

Eg: Hamiltonian Cycle.

α-appx if we output a soln that is at most $\alpha \times$ opt.

Polynomial-Time Approximation Scheme

An algorithm (given with some parameter $\varepsilon > 0$)

for any input, output a soln within a factor $(1+\varepsilon)$ of the optimal soln that runs in $n^{f(1/\varepsilon)}$.

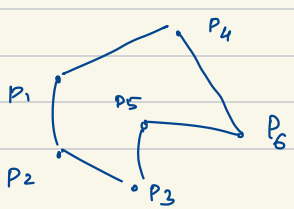
For some computable function f.

Approx. Algos – Vazirani, Shmoys & Williamson

(Running time is polynomial on n, some ε)

Travelling Salesman Problem (TSP) ~1930s

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 each city exactly once.



Decision version: Given length L, Is it possible to find a solution of length atmost L.

Graph I/p: G(V, E, W)

o/p - Visit all vertices without repetition

Minimizing the sum of edge weights

* Hamiltonian Cycle Problem

→ Hamiltonian Cycle is NP-Complete [Richard karp 70's]

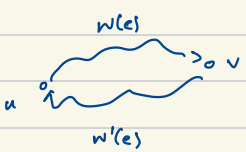
→ Infact, No constant factor approx is possible

Symmetric

e = (u, v)

w(e).

Asymmetric



Metric TSP

- $d(x,y) \geq 0$
- $d(x,y) = d(y,x)$
- Triangle Inequality: $d(x,y) + d(y,z) \geq d(x,z)$

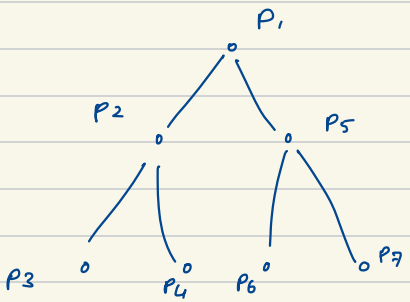
S – Set of Edges

Cost(S) – Sum of the weights of edges (min)

(Same as finding minimum weight Hamiltonian Path/ Cycle.)

Base structure :

- Min. Spanning Tree (kruskal)



Repeat not allowed

$P_1\ P_2\ P_3\ P_2\ P_4\ P_1\ P_5\ P_6\ P_5\ P_7\ P_5\ P_1$

↓

This is not a valid tour

→ We know that there is no constant factor approximation. And we haven't really done anything special

→ Adding neco paths to remove re-visitation will decrease cost Because of the triangle inequality in a metric space.

- DFS Traversal
- Delete the duplicates
  - ↳ Due to triangle inequality, duplication is always possible

Analysis : Obviously, algorithm is polynomial time.

→ Every edge of the MST is travelled twice (pre duplication)

$$cost(C) \leq 2 \times cost(MST)$$

$Cost(MST) \leq cost(opt) \rightarrow$

Cycle -> can delete the longest edge to get SpanningTree $\geq$ MST.

$\therefore$ $Cost(C) \leq 2 \times Cost(MST) \leq 2 \times Cost(opt)$

2-factor approximation!

Q Can we do better?

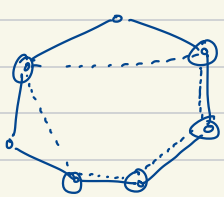
PROPERTY 1

I/p : $G(V,E)$ —— $OPT(G)$

Take a subset $S \subseteq V$

Induced subgraph $G[S]$ —— $opt(S)$

$OPT_S \leq OPT_G$



PROPERTY 2 :

Perfect Matching (Can be computed in Polynomial Time)

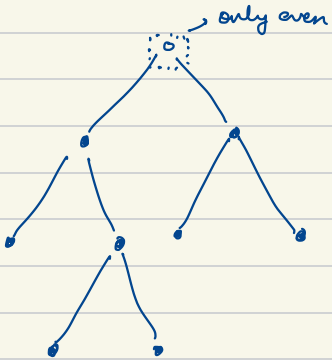
Minimum cost perfect matching is a perfect matching with smallest cost.

(Poly.)

PROPERTY 3 :

Eulerian tour (Circuit) : Start from one vertex and return back to it after visiting all edges only once. (Allowed to repeat an edge but not repeat any vertices)

If graph has even degree vertices then you always have an Eulerian cycle.



Compute an Eulerian circuit

→ if we match odd degree vertices, their degree increases by 1

→ However, we can have an odd number of odd degree vertices, can't we?

Q How many odd degree vertices can we have?

$$\sum d(v_i) = 2 \times |E|$$

Since this is even, ignoring even degree vertices, we must have an even number of odd degree vertices.

Every edge is counted twice

Now we have a graph with all even edges.

→ Compute Eulerian circuit

→ Delete duplication

Analysis :

$$Cost(C) = cost(MST) + cost(Matching)$$

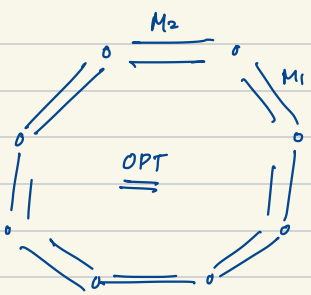
↓ removing duplication

$$Cost(C')$$

$$(cost(C') \leq cost(C))$$

$$cost(C') \leq cost(MST) + cost(Matching)$$

$$\leq cost(opt)$$



$$cost(M_1) \leq cost(opt)$$

$$cost(M_2) \leq cost(opt)$$

$$cost(opt) \geq \frac{cost(M_1) + cost(M_2)}{2}$$

$$cost(M) \leq \frac{cost(M_1) + cost(M_2)}{2} \leq \frac{1}{2} cost(opt)$$

$$cost(C') \leq cost(opt) + \frac{cost(opt)}{2} = 1.5\, cost(opt)$$

Metric TSP

- 2 - apx. (MST doubling)
- 1.5 - apx. (Chivstfeders Algo.) [1976]

↓

1.5 - ε appx. ε ~ 10-20

→ No apx. is possible if the distance is arbitrary

Q Does there exist a PTAS for Metric TSP

$(1+\varepsilon)$-approx.

$n^{O(f(1/\varepsilon))}$ time

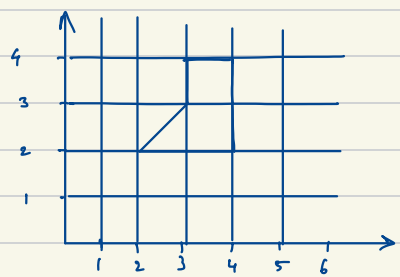
Thm. There can't be a PTAS $220/219$-apx unless $P = NP$

Restrict the metric: Euclidean Metric

I/p: A set of points in $\mathbb{R}^2$, with Euclidean distance

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

O/p: Find the shortest route that visits all pts.



It is not known whether the problem is in NP.
It is however known that the problem is NP Hard.

Sum of Square Roots

Given a set of positive integers $a_1, a_2, \ldots, a_k$, t

Decide

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

$\{a_1, a_2, \ldots, a_k\}$ $\{b_1, b_2, \ldots b_k\}$

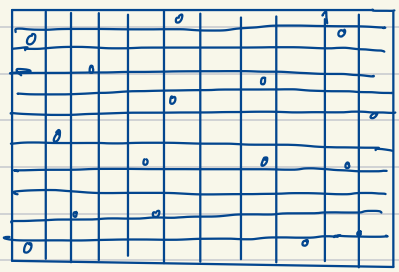
$$\sum_{i=1}^{k} a_i \leq \sum_{i=1}^{k} b_i$$

Affects all Euclidean Problems

- Euclidean TSP
- Euclidean MST
- Steiner Tree
- Matching

PTAS for Euclidean TSP [Arora (1998) & Joe Mitchell (1999)]

- Rounding the instance
- Partitioning: Exploit the structure of the instance by breaking it into "nice" instances.
- Apply dp to the instance.



R_2

① Wlog you snap each point to the closest grid point

We must first prove that doing so doesn't lead to much "loss"

ε - "nice" instance

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

1. Every point has integral coordinates in the interval $[0, O(\frac{n}{\varepsilon})^2]$
2. Any 2 different points have distance atleast 4.

→ Take a small bounding box (axis-parallel)
→ Longer side - L
→ Translate the instance st rooted at origin, scale $L = \lceil \frac{8n}{\varepsilon} \rceil$ } I'

Lemma - I is i/p & OPT_I is optimal tour
I' is ε-nice instance & $OPT_{I'}$ optimal tour

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

We know that OPT is atleast $2L$.

Draw a fine grid with spacing $\frac{\varepsilon \times L}{2n}$

- Map every point to its closest grid pt.
  (multiple pts. could be mapped to same)

- All points have int coords

$$x = \left\lceil \frac{4n}{\varepsilon} \right\rceil \in O(n/\varepsilon)$$

$$\frac{\varepsilon L}{2n} \leq \frac{\varepsilon L}{2n} \times \frac{8n}{3} = 4$$

→ Mapping each pt. in I has moved $\frac{\varepsilon L}{2n}$
(Every edge in the sol^n changed atmost $\frac{\varepsilon L}{2n}$)

Cost : $opt_I + \varepsilon \times L$

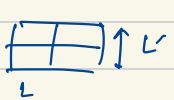
$\leq$ (since $L \leq opt_I$)

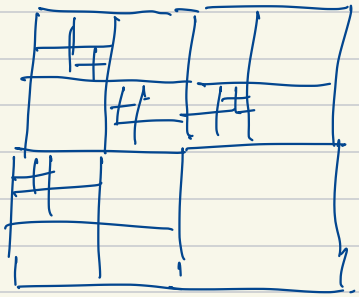
$opt_I + \varepsilon \times opt_I$

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

② Partition the Space

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





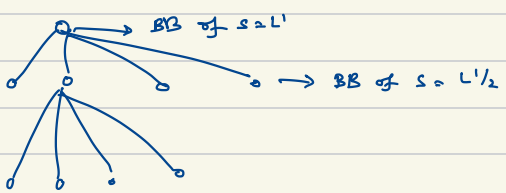
Quadtree

* Recursively partition the box (square) into four equal sized squares until the side length becomes "1 ir

↓ (L' is a power of 2)

At the end of this partition, we can guarantee that there will be atmost 1 pt- in each square
(since distance b/w 2 pts is atleast 4)

- Each pt. is separated
- One pt. in each "non-empty" square



height: $\log(L')$

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

Height of quadtree — $O(\log L')$ — $O(\log(n/\varepsilon))$

Idea: Apply dynamic programming to the Quadtree

↓

Solve for each square that are leaves

↓

Bottom-up combine.

9/8

Euclidean TSP

Given a set of n points in $\mathbb{R}^d$ with distances, $\forall x,y \; d(x,y) = \|x-y\|_2$

Output: shortest tour that visit all points

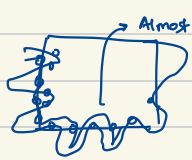
PTAS $(1+\varepsilon)$ in $n^{O(f(1/\varepsilon))}$

- ε-niceness
- quadtree partitioning
- DP

ε-niceness – 1. Integral

2. $d(x,y) \geq 4$

Quadtree - Every cell must have exactly 1 point



It is possible that the optimal tour enters and exits multiple times

$(1+\varepsilon)$- opt

Too many calls to the box will lead to exponential blowup.

→ Add "portals" on the box

→ Show that optimal tour must only enter and exist through these portals

→ Show that entry & exits through these portals is bounded.

- Limits the # of intersections.
  # of portals
  - Accuracy
  - Running time
  } Trade off

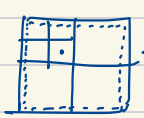
Select m to be a power of 2 $\quad m \in \left[\frac{k}{\varepsilon}, \frac{2k}{\varepsilon}\right]$

For each square,
- Put portals in corners
- Put $(m-1)$ portals equally spaced
- Ann many portals

Portal Respecting Tours

Def^n : p-tour enters & exits through portals

- Length of p-tour $\leq (1+\varepsilon)$ - optimal
- Detours can add much more cost.



→ Not a uniform distribution

Sol^n : Randomize

1. Translate the grid by a "random offset" almost $L/2$ in each coordinate
2. Points remain grid points
3. With high probability, the points are "nicely" concentrated.
4. Higher-levels in the partition (quadtree) have more portals → have fine grained tour

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

Thm: (a,b) picked up uniformly @ random $\left[0, \frac{L'}{2}\right]$
with prob atleast $1/2$

p-tour such that

$$\text{Cost}(\text{p-tour}) \leq (1+\varepsilon) \times \text{opt}$$

Proof :- Extend non p-tour to a p-tour.

* For each vertical / horizontal line l

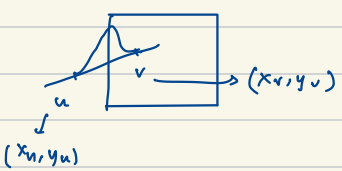
$t(l)$: # times p tour intersects l

$$T = \sum_l t(l)$$

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

- e crosses $x+1$ vertical lines
- e crosses $y+1$ horizontal lines

e crosses $(x+y+2)$ lines



$$\sqrt{2(a^2+b^2)} \geq (a+b) \quad -①$$

$$\forall x,y \quad d(x,y) \geq q \quad -②$$

$$(x+y+2) \leq \sqrt{2(x^2+y^2)} + 2$$

$$\leq 2 \underbrace{\sqrt{x^2+y^2}}_{\text{OPT}}$$

Bound - expected length of the detour

Detour might cross

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

i of the quad-tree

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

→ if ℓ is in level i $\leq \frac{L'}{2^i m}$

$\underline{?}$:- What is the probability that after "random shift" ℓ crosses e at level i:

→ ℓ could be mapped to $L'/2$ many lines [translate by $(0, L'/2)$]

→ 2^{i-1} many lines of level i

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

ub on expected length

$$\sum_i^K \frac{2^i}{L'} \times \frac{L'}{2^i m} \leq \xi$$

By linearity of expectation

$2\varepsilon \times OPT$

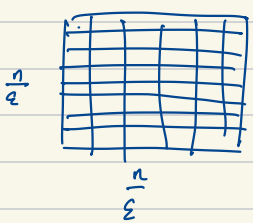
Markov Inequality

$$Pr\left[\begin{array}{l}\text{total length increase}\\ \text{of detours}\end{array} \geq 4\varepsilon\ OPT\right] \leq \frac{2\varepsilon\, OPT}{4\varepsilon\, OPT} = \frac{1}{2}$$

De randomization

- fixed ε

grid shifting by trying all possibilities



$O\left(\left(\frac{n}{\varepsilon}\right)^2\right)$

Final Step (DP)

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

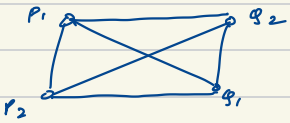
Introduces state -

- a square
- any set of possible ways of entering/exiting the square.

states must be polynomial

$1 + 4 + 4^2 + \cdots + L^2 = O\left(n^2/\varepsilon^2\right)$

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

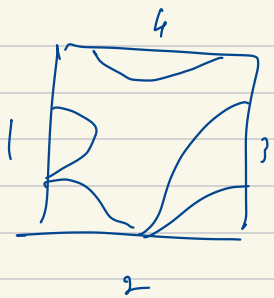


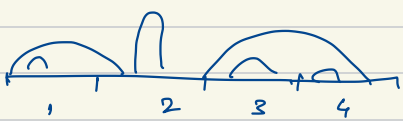
- $4m$ portals
- Use one portal $\{0,1,2\}$ times

$3^{4m} \rightarrow \quad m = o(k/\varepsilon)$

$= o(\log k/\varepsilon)$

$= O((\log n/\varepsilon))$





$(( ))( )( )( )$

r^{th} Catalan Number $\rightarrow C_r = \frac{\binom{2r}{r}}{r+1} = o(2^{2r})$

$= o(2^{8m})$ $\quad m$ is $o(k/\varepsilon)$

→ Translate them into paths

→ Discard anything that intersects

$m : O\left(\log\left(n/\varepsilon\right)/\varepsilon\right)$ // Entry/Exit = $O(n/\varepsilon)$

Computation of values:

$A[(s_1, t_1), (s_2, t_2), \ldots (s_Q, t_Q)]$

↳ Compute the whole table

CLUSTERING

- learning, searching, data-mining, ...

→ Data represented as points in $\mathbb{R}^d$

→ General metric space (X, d)

- X: space ($\mathbb{R}^d$) (set)
- d: distance metric, $d : X \times X \to [0, \infty)$

(X, d) is a metric if it satisfies —

(i) if $x = y$ $d_M(x, y) = 0$

(ii) $\forall x, y$ $d_M(x, y) = d_M(y, x)$

(iii) $\forall x, y, z$ $d_M(x, y) + d_M(y, z) \geq d_M(x, z)$

Assumption: Given x, y $d_M(x, y)$ can be computed in $O(1)$ time.

Norm

↳ norm defines distances between points

$p, q \in \mathbb{R}^d$

$$\|p-q\|_t = \left(\sum_{i=1}^{d} |p_i - q_i|^t \right)^{1/t} \quad \text{for } t \geq 1$$

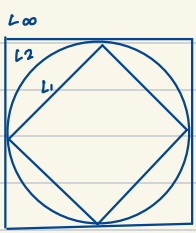
$t=2$: Euclidean Norm

$t=1$: L_1 - norm. Manhattan norm

L_∞ - norm : $\|p-q\|_\infty = \lim_{t \to \infty} \left(\sum |p_i - q_i|^t \right)^{\frac{1}{t}}$

$$= \max_i \{ |p_i - q_i| \}$$

Triangle inequality holds for L_∞ too. it's called Minkowski Inequality



For any pt. $p \in \mathbb{R}^d$

$$\|p\|_t \leq \|p\|_s \quad \text{if} \quad t > s$$

Lemma - For any $p \in \mathbb{R}^d$

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

Proof :- $p = (p_1\ p_2\ \ldots\ p_d)$ & $p_i \geq 0\ \forall i$

Const. α

Obs: $f(x) = x^2 + (\alpha - x)^2$ minimized if $x = \alpha/2$

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

By symmetry & obs. on $f(n)$

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

K-centre clustering

Metric Space (X, D)

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

O/P : Find n-clustering (set of n clusters)
such that each pt. is assigned to its nearest centre

Set of clusters C

$$\text{cluster}(C, \tilde{c}) = \left\{ p \in P \;\middle|\; d_M(p, \tilde{c}) = d_M(p, C) \right\}$$

($\tilde{c}$: prime cluster; $d_M(p, C)$: max dist in the set)

n dim pt. $p_c = (d(p_1, c), d(p_2, c), \ldots, d(p_n, c))$
ith coordinate $d(p_i, c)$ dist. to p_i
to its closest centre.

O/P :- Find a set of k-centres $C \subseteq P$ such that the maximum distance of a pt. in P to its closest centre is minimized.
(Facility Location)

Formal Statement

Given a set of k-centres C,

$$\|p_c\|_\infty = \max_{p \in P} d(p, C)$$

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

$$Opt_\infty(P,k) = \min_{C \subseteq P, |C|=k} \|P_C\|_\infty$$

→ C_{opt}

→ NP-Hard

→ Hard to approximate within 1.86

→ 2-approximation in the Euclidean case in $\mathbb{R}_2$

Greedy-Algo

- Start by picking an arbitrary point, $\bar{c}_1$: $C_1 = \{\bar{c}_1\}$
- Compute the distances for each $p \in P$, from $\bar{c}_1$
- Take pt. with worst dist.

$$\left(r_1 = \max_{p \in P} d_1(p) \right)$$

Say $\bar{c}_2$

add c_2 : $C_2 = C_1 \cup \{\bar{c}_2\}$

i^{th} Step: $C_i = C_{i-1} \cup \{\bar{c}_i\}$

We have atmost k-centres $\Rightarrow$ atmost k iterations to the algorithm. $O(n)$ time to compute distances & update.

$O(nk)$ time

$O(n)$ space

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

then

$$d[p] \leftarrow \min\left(d[p],\ d_M(p, \bar{c_i}) \right)$$

Defn

A ball of radius r around a pt. $p \in P$ is a set of points in P, with dist. atmost r from P

$$b(p, r) = \{ q \in P \mid d_M(p, q) \leq r \}$$

Remark - K-centre is essentially covering k with k balls of min Radii

Thm. Greedy Algo computes a set k of k-centres s.t. k is 2-approx.

$\|p_k\|_\infty \leq 2 \times opt_\infty$

and takes $O(nk)$ time.

Proof :- Running Time ✓

→ By Defn $r_k = \|p_k\|_\infty$

Let $\tilde{c}_{k+1}$ is the pt. realizing $r_k = \max_{p \in P} d(p, k)$

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

By the definition of r_i,

$$r_1 \geq r_2 \geq \cdots \geq r_k.$$

$i < j < k+1$

$$d_M(\bar{c_i}, \bar{c_j}) \geq d_M(\bar{c_j}, c_{j-1}) \qquad r_{j-1} \geq r_k$$

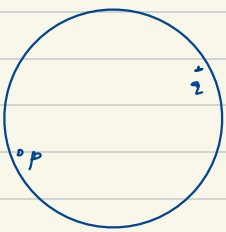
- the distance between any pair of points in C_i is atleast r_k

Opt :- Clusters P by using k balls

By Triangle inequality - any 2 points within such a ball are with a dist. atmost 2x opt

↓

None of the balls contain 2 points for $C \subseteq P$



Greedy Permutation

Let this run till we exhaust all points

$\langle p \rangle = \langle \bar{c}_1, \bar{c}_2, \ldots \bar{c}_n \rangle$

↓

$\langle r_1, r_2, \ldots r_n \rangle$

Def: r-packing. A set $S \subseteq P$:

(i) Covering Property: All points in P are within distance of atmost r from S.

(ii) Separation Property: $\forall p, q \quad d_\mu(p,q) \geq r$

* r-packing gives compact representation
* Greedy permutation gives such a representation

Then

$$\langle \bar{c_1}, \bar{c_2}, \ldots, \bar{c_n} \rangle \;,\; \langle r_1, r_2, \ldots, r_n \rangle$$

for any i, we have $c_i = \langle \bar{c_1}, \ldots \bar{c_i} \rangle$ is an r_i-packing of P

Proof :- By contradiction,

$$r_{k-1} = d(\bar{c_k}, \bar{c_{k-1}}) \qquad \forall k = 2, \ldots, n$$

For $j < k \leq i \leq n$

$$d_\mu(\bar{c_i}, \bar{c_k}) = r_{k-1} \geq r_i$$

---

k-median cluster

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 ptr. in P to its closest centre is minimized.

Clustering price

$$\|P_C\|_1 = \sum_{p \in P} d(p, c)$$

Objective fn

$$OPT_P(P, k) = \min_{\substack{C \subseteq P, \\ |C| = k}} \|P_C\|_1$$

Optimal Set of Centre – C_{OPT}

Local search: Move sol^n to sol^n in the space of the candidates Sol^n (the search space) by applying local change

Continue until - end up on optimal

or

we exhaust the running time

For this problem ^nC_k possible solutions - so an exhaustive local search should lead to the optimal sol^n.

Notations

A set $U = \{ p_c \mid c \in P^k \}$

$OPT_\infty(P,k) = \min_{q \in U} \|q\|_\infty$ — k-center

$OPT_1(P,k) = \min_{q \in U} \|q\|_1$ — k-median

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

$$OPT_\infty(P,k) \le OPT_1(P,k) \le \sqrt[d]{n} \times OPT_\infty(P,k)$$

Proof:.. $p \in \mathbb{R}^n$

$$\|p\|_\infty = \max_{i=1}^{n} |p_i| \le \sum_{i=1}^{n} \|p_i\| = \|p\|_1$$

$$\|p\|_1 = \sum_{i=1}^{n} \|p_i\| \le \sum_{i=1}^{n} \left(\max_{i=1}^{n} \|p_i\| \right) \le n \max_{i=1} |p_i| = \sqrt[d]{n} \|p\|_\infty$$

C - set of centres ; $|C| = k$

realizing $opt_1(P,k)$ i.e. $opt_1(P,k) = \|P_c\|_1$

$$opt_\infty(P,k) \le \|P_c\|_\infty \le \|P_c\|_1 = opt_1(P,k)$$

Similarly,

k relating $OPT_\infty(P,k)$

$$opt_1(P,k) = \|P_c\|_1 \le \|P_k\|_1 \le n \times \|P_k\|_\infty = n \times opt_\infty(P,k)$$

→ $2d$ - factor for k-median

This is not good. We can't effectively modify the Greedy Algorithm either since it starts getting very inefficient.

Here we need Local Search. Given a initial algorithm, we try to improve it using Local Search.
We use this greedy algorithm as the first step of Local Search.

L is $2d$ - approximation

Improve - parameter $0 < \delta < 1$

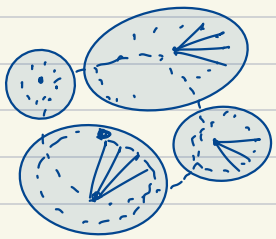
$\forall i \in [n]$

$L_i \subseteq C_{i-1}$

Local Search –

→ Set : $L_{curr} \leftarrow L$

→ At each iteration



This is very bad — non centers

We try and kick out a point & add a new pt. to improve cost by some factor δ

Swap → $K \leftarrow (L_{curr} \setminus \{\bar{c}\}) \cup \{\sigma\}$ ↳ new

if $\|P_K\|_1 \leq (1-\delta)\|P_{L_{curr}}\|_1$

→ Continue "swap" as long as it satisfies the constraint

→ Return L_{curr}

Running Time

An iteration takes $O(n \times k)$ (naively) swaps

$n-k$ can be swapped in

k to be swapped in

Since $\frac{1}{1-\delta} \geq 1+\delta$

$$O\left((nk)^2 \log_{\frac{1}{(1-\delta)}} \frac{\|P_L\|_1}{opt_1}\right) = O\left((nk)^2 \log_{1+\delta} 2n\right)$$

$$= O\left((nk)^2 \frac{\log n}{\delta}\right)$$

K-means

Set $P \subseteq X$, K,

Find k points $C \subseteq P$, $|C| = k$,

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

Obj:-

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

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

$O(1)$ - ~~factor~~ for k-means too.

Same strategy, apply standard greedy & then Local search

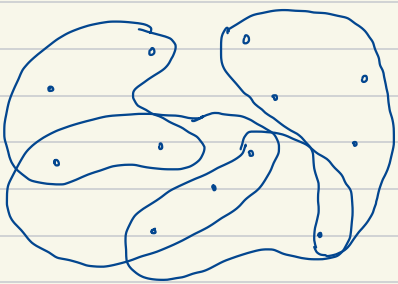
Then $0 < \varepsilon < 1$

$(25 + \varepsilon)$ factor appx $O\left(n^2 k^3 \frac{\log n}{\varepsilon}\right)$

VC-Dimension

→ A range space $(X, R) = S$

X : ground set (finite / infinite)
R : family of subsets of X

→  $\mathbb{R}^d$

→ Sampling for exit polls for example
- blue blobs are groups.
- possible for e to be in multiple grps

→ Consider finite subset of X as the estimating ground set.

$R \setminus N$

Defⁿ (Measure)

X fixed subset of X. For a range $r \in R$, measure for r

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

Estimate

For a subset N (multi-set) of X, its estimate of the measure of $\bar{m}(r)$, for $r \in R$

$$\bar{s}(r) = \frac{|r \cap N|}{|N|}$$

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

$$\bar{s}(x) \simeq \bar{m}(x), \quad \forall x \in \mathbb{R}$$

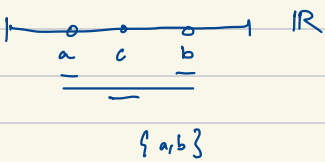
VC-Dimension
↳ Chervonenkis
Vapnik (1971)

Def^n $S = (X, R)$ for $Y \subseteq X$

$R_{|Y} = \{ r \cap Y \mid r \in R \}$ ↳ if $2^{|Y|}$ then we call it shattered by R

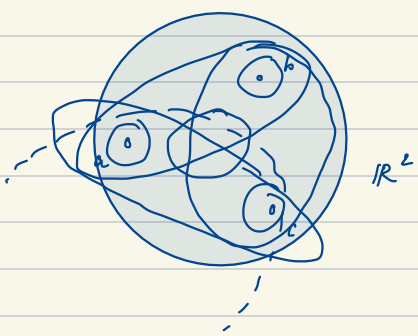
be the projection of R on Y

The range space S is projected to $S_{|Y} = (Y, R_{|Y})$

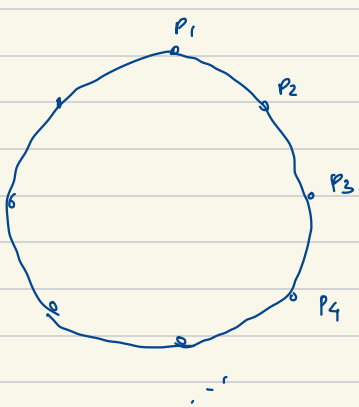


$\mathbb{R}$

Can't shatter



$\mathbb{R}^2$



$\dim_{VC}(S) = \infty$

Complement

$S = (X, R)$, $s = dimvc(S)$

$\bar{S} = (x, \bar{R})$, where $\bar{R} = \{ X \setminus R \mid R \in R \}$

Q What is the VC-dimension of $\bar{S}$?

A subset $B \subseteq X$ is shattered in $\bar{S}$.
iff it is shattered in S.
For any $Z \subseteq B$, $(B \setminus Z) \in R_{|B} \Rightarrow Z = B \setminus (B \setminus Z) \in \bar{R}$

Thm :- $dim_{VC}(S) = dimvc(\bar{S})$

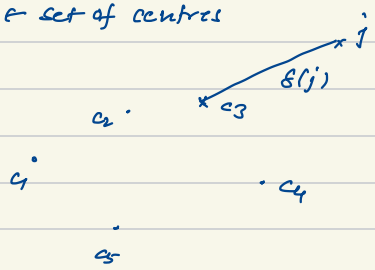
Local Search

- Let X be the set of arbitrary subset of k-centres (greedy sol^n k-centres)

```
while True :
    swap : if i ∈ X and i' ∈ P \ X
           Cost ( X - i + i' ) < cost(X) :
               X ← X - i + i'
           otherwise :
               break;
```

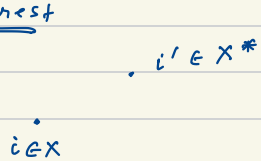
Claim: 5-approx

$X \leftarrow$ set of centres



opt $= X^*$ — optimal set of k-centre

Nearest



→ For each centre chosen in X, the nearest centre in X^* is what we want

minimum(i, i^*)

Inverse

For the centers in X^*, inverse is the nearest map.

Bijection

Claim:- For any $j \in X$,

$$d(\text{nearest}(o^*(j)), j) \leq d_j + 2d_j^*$$

Half Space

If d dimensional then $d-1$ dimensional half spaces

Let 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 nos. $\beta_1, \beta_2, \dots \beta_{d+2}$ (not all are zero)

s.t. $\sum_i \beta_i p_i = 0$ & $\sum_i \beta_i = 0$

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

pts are linearly dependent

$(q_1, q_2, \dots, q_{d+2} \in \mathbb{R}^{d+1})$

$\exists$ these are coefficients $\beta_1, \dots, \beta_{d+2}$

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

$\leftarrow$ Considering first d- coordinates of these pts.

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

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

Radon's Theorem :- $P = \{p_1, \dots p_{d+2}\}$

$\exists$ disjoint subsets C & D of P

s.t. $CH(C) \cap CH(D) = \phi$

$C \cup D = P$

Shattering dim

Property :- If range space (R) with VC-Dim(δ) means # of ranges grow polynomially on (n), (Generally this exp)

Growth fn :- $G_\delta(n) = \sum_{i=0}^{\delta} \binom{n}{i} \leq \sum_{i=0}^{\delta} \frac{n^i}{i!} \leq n^\delta$ for $\delta \geq 1$

Sauer Lemma :- $S = (X, R)$

$VC(S) = \delta$, $|X| = n$, $|R| \leq G_\delta(n)$

Proof :- $n = 0$, $\delta = 0$, done

For some element $x \in X$

$$R_x = \{ r \setminus \{x\} \mid r \cup \{x\} \in R \ \& \ r \setminus \{x\} \in R \}$$

contains

and

$R \setminus x = \{ r \setminus \{x\} \mid r \in R \}$ doesn't contain x

Obs :- $= |R_x| + |R \setminus x|$

Shatter Function

$S = (X, R)$, shatter fn

$\Pi_S(m)$ is the maximum # of sets that might be created by S, when restricted by the subsets of size m

$$\Pi_S(m) = \max_{\substack{B \subset X \\ |B| = m}} |R_{|B|}|$$

Shattering dim

↳ The smallest d s.t. $\Pi_S(m) = O(m^d)$, $\forall m$

Thm. $S = (X, R)$ has shattering dimension d, then the VC-Dimension is bounded by $O(d \log d)$

Proof:- $N \subseteq X$ be the largest subset of X shattered by S & S is its cardinality

$$2^\delta = |R_{|N|}| \leq \Pi_S(|N|) \quad (\delta \geq \max(2, 2\log c))$$

$$\delta \leq \log c + d \log \delta$$

$$\Rightarrow \frac{\delta - \log c}{\log \delta} \leq d$$

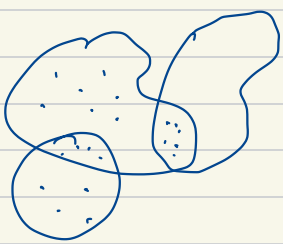
$$\frac{\delta}{2\log \delta} \leq d \Rightarrow \frac{\delta}{\log \delta} \leq O(1) \times d$$

$f(x) = x/\log x$

- mon. inc. $x \geq e$
- $f(x) \geq e$, for $x > 1$
- $u \geq \sqrt{e}$. $f(x) \leq u$ then $x \leq u \log u$

ε-net and ε-sampling

ε-sample



$S = (X, R)$

X is a finite subset of $\mathcal{X}$

$0 \leq \varepsilon \leq 1$, a subset $C \subseteq X$ is an ε-sample for X if for any range $r \in R$

$$|\underbrace{\bar{m}(r)}_{\text{measure}} - \underbrace{\bar{s}(r)}_{\text{estimate}}| \leq \varepsilon$$

(informally, ε-sample captures R, upto some "ε" error)

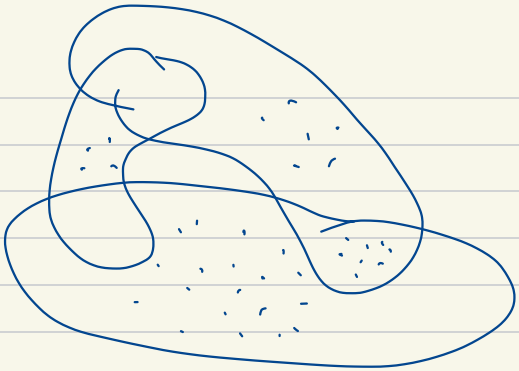
Thm. (ε-sample, VC-dim) :- There is a +ve constant c, s.t. if (X, R) is any range space with VC-Dimension δ.

$X \subseteq \mathcal{X}$: finite subset of $\mathcal{X}$ and $\varepsilon, \phi > 0$,

a random subset $C \subseteq X$ of cardinality

$$s = \frac{c}{\varepsilon^2}\left(\delta \log \delta/\varepsilon + \log 1/\phi\right)$$

with probability $1 - \phi$



ε-net :- A set $N \subseteq X$ is an ε-net for X if for any range space $r \in R$ if $m(r) \geq \varepsilon$

$$|r \cap X| \geq \varepsilon |X|$$

then r contains atleast one pt. of N i.e. $r \cap N \neq \phi$

ε-net theorem (HW&7)

$S : (X, R)$ has VC-Dimension $(S) \leq \delta$

X is a finite subset of X

Suppose $0 \leq \varepsilon \leq 1$ & $\varphi < 1$.

– N a set obtained by random independent m draws

$$m \geq \max\left(\frac{4}{\varepsilon} \log \frac{4}{\varphi}, \frac{8\delta}{\varepsilon} \log \frac{16}{\varepsilon} \right)$$

Then N is a ε-net with prob. $1 - \varphi$

Remark:- Both of the thms. hold for spaces with shattering dim δ

$$O\left(\frac{1}{\varepsilon} \log \frac{1}{\varphi} + \frac{\delta}{\varepsilon} \log \frac{\delta}{\varepsilon} \right)$$

Range Searching

Points in a d dimensional space $\mathbb{R}^d$.
We have a database.

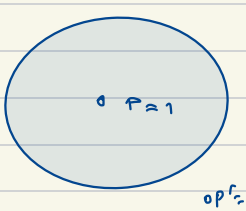
Given a hyper-rectangle we want to report the points that lie inside

Allow, 1% error

\# ε-sample thm. says that there is a subset of constant size (which depends on ε)
Use this to perform an estimation

Rects. have $O(1)$- VC dimension
Sample with prob. $1-\varphi$.

Learning Concepts



Query oracle: Assume, we know a function that returns "1" if inside, & "0" otherwise

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

Growth-function

VC-dim = d

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

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

Suppose VC-Dimension of $S \leq d$

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

$|R|$ = $\Pi_f(n)$

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

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

We prove for n & d

define f :- $\sum_{i=0}^{d} \binom{n}{i} = h(n, d)$

Our induction hypothesis is for f with VC-Dimension $\leq d$

$\Pi_f(n) \leq d$

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

Easy to verify $h(n,d) \le h(n-1,d) + h(n-1,d-1)$

recurrence.

Now let's fix a class F

VC-Dim$(F) = d$ and a set $X_1 = \{x_1, \dots, x_m\} \subseteq X$

$F_1 = f_{|X_1}$

$F_2 = F_{|X_2}$

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

VC-Dim$(F') \le$ VC-dim$(F) \le d$

$|F_1| = \underbrace{|F_2|}_{\le d} + \underbrace{|F_3|}_{\le d-1}$

$\frac{\text{I.H.}}{|F_1|} \le h(n-1, d)$

$|F_3| \le h(n-1, d-1)$

$\Rightarrow |F| \le h(n-1,d) + h(n-1,d-1)$

$\le h(n,d)$

Ex :- Let F be s.t.

VC Dim$(F) \le d$ for $n \ge d$

$\pi_F(n) \le \left(\frac{ne}{d}\right)^d$

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

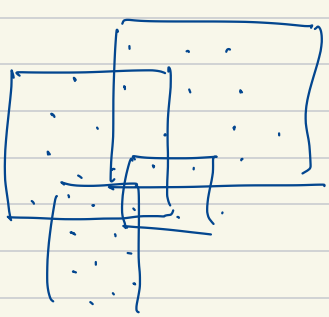
Greedy approx: 1. Sort the set based on cardinalities
2. Choose the set with max cardinality.
} → log-factor

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

Wish for "nice" set families. can we bound Greedy (log-factor)

$S = (X, R)$
- X: Set of Elements
- R: Set of Ranges

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



Class of objects – has bounded VC-Dimension

↳ This implies that there is an epsilon-net

↳ Epsilon-net is a sample that "hits" all the heavy sets ($\geq \varepsilon n$)

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

Construction of ε-net

1 – Choose a "random" sample. If it's large enough it's 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

$$m \geq \max\left\{ \frac{4}{\varepsilon} \log \frac{4}{\varphi}, \frac{8d}{\varepsilon} \log \frac{16}{\varepsilon} \right\}$$

with prob. $> 1-\varphi$

Suppose the shattering dimension is $\dim(d)$

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

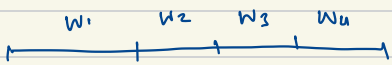
Weighted Net

Suppose the elements are weighted ($w: X \to R^+$)

R - subset

$$w(R) = \sum_{j \in R} w(j)$$

Goal: Hit all the rs with wt. $\geq \varepsilon \times w$.



Algorithm $S = (X, R)$ dual - $S^* = (X^*, R^*)$

1. Repeatedly select an ε-net (for some ε) of S^* → shattering dim S^*
2. Size of the net = $O\left(\frac{S^*}{\varepsilon} \log \frac{S^*}{\varepsilon}\right)$
3. Verify if it is a net. If not, discard.
   ↳ if it is a net, check if it is a set cover
   if yes → Done
   if no →
   - Let $R_p = \{r \in R \mid p \in r\}$ all ranges that contain p
   - Double the weight (Multiplicative weight update) of the elements in R_p.

Obs:- Every time we double, we are increasing not more than $(1+\varepsilon)$ multiplicative factor

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

i many iterations.

Minimum weight of elements $k = |opt|$ in opt?

$$k \times 2^{i/k} \leq w_i = (1+\varepsilon)^i w_0 = (1+\varepsilon)^i \times m$$
$$\leq e^{i\varepsilon} \times m$$

$$i = k \times q$$
$$k \times 2^q \leq e^{iq} \times m$$

$$\log k + q \leq \log m + \varepsilon i$$

$$q(1-\varepsilon k) \leq \log \frac{m}{k}$$

# of iterations $O(k \log \frac{m}{k})$

Suppose we take $\varepsilon = \frac{1}{2k} \Rightarrow q \leq O\left(\log\left(\frac{m}{k}\right)\right)$

The problem is that we don't know what ε is?

↳ Binary search → this is almost a log overhead to the whole process.

→ Size of our core

$$O(S \times k \log(S \times k))$$

↳ k opt cover

Q How to choose ℓ?

ℓ is independent of k
Suppose we choose $\ell = \frac{1}{4k}$.

Guess the value of ℓ

$O\left(k_i \log \frac{m}{k_i}\right)$

Parametrized Algorithms

↳ Hard Problems & Exact Solutions

Idea: We aim for exact algorithm
But we want to isolate exp^n term (parameters)

⇒ obtain very fast soln, when parameter small.
(Hope: parameters are small in practice)

Parameters — non-negative integer $k(x)$ (comes with probability i/p)

↳ denote by k
↳ Not necessarily efficiently computable.

Parametrized Problem

- Problem + Parameter (k)
  (wrt k)

Goal:- polynomial complexity on n.
exponential complexity on k.

Example:-

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

O/p: Does there exist

a k-size VC (output a set $S (\leq k)$ s.t. $\forall e \in E, \exists v \in S \rightarrow v \in e$)

Brute force soln

- Try all $\binom{n}{k} + \binom{n}{k-1} + \binom{n}{k-2} + \cdots + \binom{n}{0}$,
  All sets of k-vertices
- Testing invalid VC takes $O(E)$
- Total - $O(V^k E)$ - poly. for fixed k - this is slow

Branching (Bounded Search tree technique)

- pick an arbitrary edge $e \in E$
- We know either $u \in S$ or $v \in S$
  or $\{u, v\} \in S$

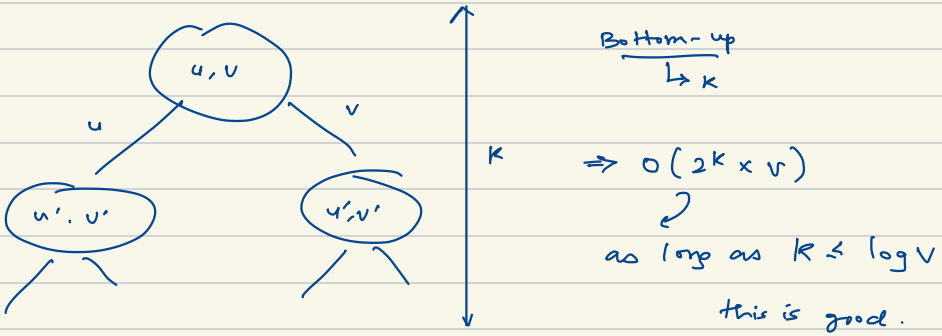
Guess - Try both

① Add u to S

(delete u & $N(u)$ from S). recurse $k' = k-1$

② Same for v

- Return OR of the both



Bottom-up

↳ k

$\Rightarrow O(2^k \times V)$

as long as $k \leq \log V$

this is good.

Fixed-parameter Tractable (FPT)

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

↳ polynomial

$f : \mathbb{N} \to \mathbb{N}$

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

Thm :- $f(k) \times n^{O(1)} \iff f(k) + n^{O(1)}$

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

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

if $f(k) \leq n$ then

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

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

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

$\Leftarrow$ Trivial assumption

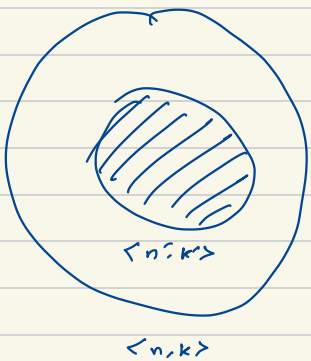
$f(k) \& n^{c'} \geq 1$

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 :- $\text{Ans.}(\langle n, k \rangle) = \text{Ans.}(\langle n', k' \rangle)$

Thm.

FPT $\Leftrightarrow$ Kernelization

($\Leftarrow$) Kernelize $\Rightarrow n' \leq f(k)$ run any finite $g(n')$ $\Rightarrow n^{O(1)} + g(f(k))$ time

($\Rightarrow$) A runs in $f(k) \times n^c$ if $n \leq f(k)$ - kernelized

If $f(k) \leq n$

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

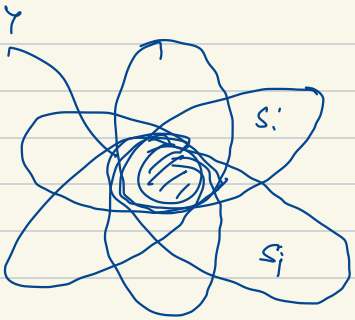
o/p O(1) size Yes/No

Sunflower Lemma (Erdos-Rado Conj)

- Classical results from 1960
- Apply in kernelization

Sunflower is a collection of sets $S_1, S_2, \ldots, S_k$
s.t. $S_i \cap S_j = Y \quad \forall\, i \neq j$ → Petals $\forall i$, $S_i \setminus Y$

- k petals (None of them can be empty)
- A core Y



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 to compute this
↳ $|F|, |U|, k$

Proof:- By induction on d, for $d = 1$, singletons
Suppose $d \geq 2$, Let $\mathcal{B} = \{S_1, S_2, \ldots, S_t\} \subseteq F$ be inclusion-wise maximal family of pairwise disjoint sets in F.

→ If $l \geq k$, then G is a sunflower already with atleast k petals

→ Assume $l < k$.

$$S = \bigcup_{i=1}^{l} S_i$$

Then $|S| \leq d \times (k-1)$

Since G is maximal, every set $A \in F$, intersects atleast one set from G $A \cap S \neq \phi$

Therefore, there is an element $u \in U$, which is contained in atleast

$$\frac{|F|}{|S|} > \frac{d!\,(k-1)^d}{d(k-1)} = (d-1)!\,(k-1)^{d-1}$$

Sets from F

→ Take all sets of F containing this element u

Construct F' of sets with cardinality $(d-1)$ by removing u

$$|F| > d!\,(k-1)^d$$

By i.h.

F' contains sunflower $\{S_1', \ldots, S_k'\}$ with k petals

$$\{\{S_1' \cup u\}, \{S_2' \cup u\}, \ldots \{S_k' \cup u\}\}$$

Poly time Alg.

→ Greedily select maximal set
If size is atleast k - done
Else, find u and recurse.

Erdos - Rado - 1960 (London Math Soc.)

Each set in $\mathcal{F}$ has cardinality d. If $|\mathcal{F}| > d!(k-1)^d$, then there is a sunflower

$(\log k)^d$ (Annals. of Math 21, Alweiss et. al)

Terry Tao's blog - Sunflowers Lemma via Shannon's entropy

---

d- Hitting Set
(Application of Sunflower Lemma)

I/p → Family of Sets $\mathcal{A}$ over U
each set has cardinality atmost d,
a non-negative integer k

O/p → whether there is a subset $H \subseteq U$ of size atmost k
s.t. H contains atleast one element from each set.

Proof :-

If $\mathcal{A}$ contains a sunflower,

say $S = \{S_1, S_2, \ldots S_{k+1}\}$ of $\# k+1$

then every hitting set # of A of cardinality

at most k intersects its core (some y)

Reduction Rule

$(U, \mathcal{A}, k) \rightarrow$ Return (U', A, k)

$A' = (A \setminus S) \cup \{y\}$

$U' = U_{x \in A'} x$

Kernel formation algo.

If # sets $> d! \, k^d$, find a sunflower (not having it is not a problem since its already a kernel)

— Apply recursively

Kernel size $\leq d! \times k^d$.

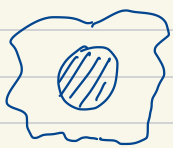
Fixed-parameter algo ??

Randomized Methods in Parametrized Algorithms

Color-coding [Alon, Yuster, Zwick J. ACM, 1995]

↳ Detect small subgraph in a large input graph

Framework:



↳ $G(V, E)$

I/p = n vertex graph G

O/p = n vertex subgraph H of G
(of some particular pattern)
(clique, Ind. set)

Brute-force :) check every subgraph

↳ $O(n^k)$

Goal – $\boxed{2^{O(k)} * n^{O(1)}}$

→ This is achievable for trees

Trees - Treewidth (graphs which are not trees but tree-like)

Q Can we always get something?

- Most likely, no

k-clique

Framework

Randomly color G, in such a way that a "pattern" emerges w.h.p.

k-Simple path

(Remark - It is possible to de-randomize using splitters or pseudo-random generators)

I/p - A graph $G=(V, E)$, +ve integer k

O/P - k-simple path

Idea - Color vertices in V, uniformly at random from $\{1, 2, \dots k\}$ and find a path, if it exists

Hope - this process gives a "colorful" path.

Q Then is a set S of k vertices. what is prob. that all vertices get different color?

$$\frac{k!}{k^k} \geq \frac{1}{e^k} \quad \text{(Stirling appx)}$$

Algorithm

↳ Repeat $e^k \times t$ times

- Pick random f:

$$f: V(G) \to \{1, 2, \dots, k\}$$

- Look for a colorful path

↳ Success - If the algorithm gets a path, then G has a path

Failure - If $\exists$ a k-path but the Alg. doesn't find it

$$1 - \left(\underbrace{1 - \frac{1}{e^k}}_{\text{↳ failure in one round}}\right)^{e^k \times t} \geq 1 - \frac{1}{e^t}$$

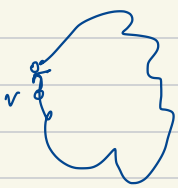
↳ success in one round

This algorithm doesn't give us the path

Remark :- No randomization in colorful path finding

Homework :- $k^{O(k)}$ time alg. for finding colorful path.
(BTES/DTES)

→ Dynamic Programming based on simple observations



$T[S, v] = \begin{cases} \text{True. if } \exists \text{ a path of } |S| \text{ vertices ending in } v \text{ using all elements of } S \\ \text{False otherwise} \end{cases}$

$\rightarrow \bigcup_{u \in N(v)} T[S \setminus f(v), u]$

↳ # of table entries

$= 2^k \times n^2$

↳ $O(n)$ time for filling up one entry

# of runs $= e^k \times \epsilon$

Total time → $O\left((2e)^k \times n^2\right)$

Derandomization

↳ De-randomize

Q How can we make the random coloring deterministic?

- Let $F = f_1, f_2, \ldots, f_k$ be a family of f^s with $f_i : V(G) \to \{1, 2, \ldots k\}$

→ F is a universal hash family if for every set $S \subseteq V(G)$ of size k there is $f_i \in F$ s.t. f_i makes S colorful.

→ Construct a k-perfect hash family F.
For each $f \in F$, look for a colorful k-path

Total time - $O(t + |F| \times 2^k \times n^2)$

↳ [Naor, Schulman, Srinivasan, FOCS-1995]
↳ Construct k-perfect hash families of size $e^{k + o(k)} \log n$