CSE 264 Approx. Algs

CSE 142 Machine learning

CSE 185E Tech. writing

ECE 175 Energy ctrl + gen.

Vasgos

Approximation Algorithms

NP-Hard algorithms, polynomial time, where not known

We do a presentation, read papers, and this is under theoretical CS

Aproximating efficient values

class is reading papers, examining problems, presenting

Approximate solution for drive to SJ.

Given $G(V, E, w)$ $w > 0$ ← what does this mean. w means weight

$A \xrightarrow[\text{path}]{\text{shortest}} B$ BFS is a sln.

Dijkstra generalizes BFS V (n) $=$ # vertices, M (E) $=$ # edges

→ $O(m \log n)$

if $n << m$ then → $\approx \tilde{O}(m)$

→ 3 properties?

Polynomial Time ①, computes optimal solution ②, Always correct ③

→ "efficient algorithm" $O(n)$, $O(n^2)$, $n \log n$, n^3 not 2^n or n^n

You can relax all 3 properties.

like 1 → slight exponential time: $1.3^n << 2^n$

For 2 → maybe approximate the optimal solution,

APX → 120, opt = 100

$apx / opt = 1.2$

One method for
the multiplicitive factor appx: $apx \leq \alpha(opt)$
↳ common
↳ multiplicitive factor

& additive factor appx: $apx \leq opt + B$ ↳ additive factor
↳ not scalar, not popular

③ Relaxes the probability of being correct. Related to analysis like worst-case $(O(n))$

any graph

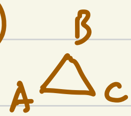
→ Dijkstra $\forall G \leq O(m \log n)$

But most places care for specific, not generic, instances.

Average case analysis, on random graphs. (heads or tails on each node/edge existing). $G(V, E, w)$. Cryptography looks at average case, & many algorithms care about this.

(social network)

Some situations have high clustering coefficients



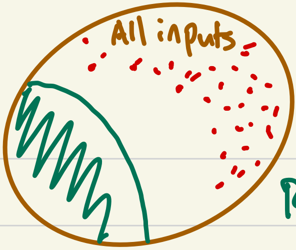
while random graphs are 1/2 (heads/tails exists)

How many friends in networks, for best algos, $E(deg) = p(n-1)$

This is power scaling, popular will have higher degree.

So for some products avg or worst case might not be best.

Beyond worst-case analysis (WCA) uses assumptions for inputs to creating algorithms.



Very bad cases (NP hard/complete)

Real world input space, typical inputs.

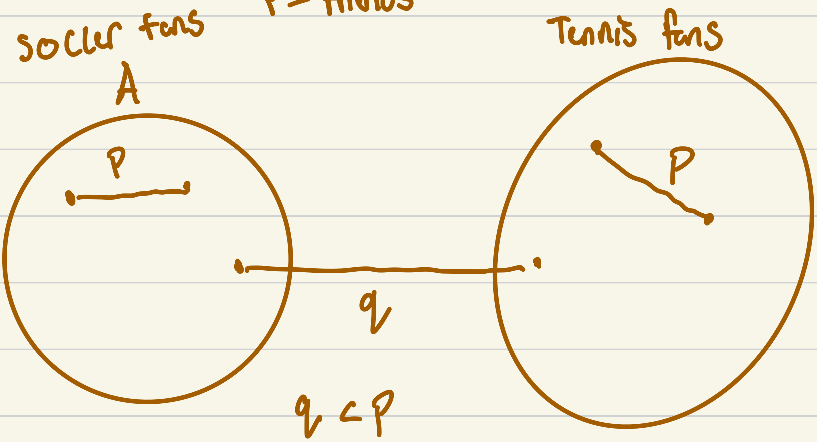
We're using this information for approx. algorithms

These are called 'planted' solutions, like a password to brute force

(example)

A popular model is called stochastic block model for community detection. Can you recover the 2 communities?

p = friends



Can you recover the communities, and can you do it in polynomial time (effic.).

G(n,p)

$G(n,p)$

↳ edge present w/ probability p

Recovery means finding all nodes inside A, and all nodes in B.

Exact recovery would be finding all the people in each community based on connections $(p = 0.8, q = 0.6)$. for example.
(for an outsider guessing person in A or B).

Approx recovery of communities would label a subset and find it.
algos exist such that if $p - q = x$, then can find solution $(x < .2?)$
We want to recover w/ better probability than $n/2$ ($1/2$ guess).

What is satisfiability?

3 SAT $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_7 \vee x_8 \vee \bar{x}_9) \wedge \ldots$

↑ or ↑ and

↳ NP-complete, All np-complete are a 3 SAT special case, vice-versa.

Read 1/2 papers, write a report, do research. and present to the class, and then a final exam.

Lecture 2, Oct 2nd

Get add. code?

Beyond WCA?

How does google search work?

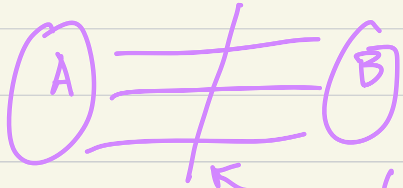
Heirarchical clustering.

Find partition of dataset into similar looking data points.

Clustering. Given points $S = \{x_1, x_2, \dots x_n\}$
Find similar points.

Max-cut is a clustering problem. Points have a distance, separating into 2 groups is based on distance

$G(V, E, w)$



max $w(\text{cut})$

weight is maximizing the cut of connections.

K center, k median, k means, k shape optimize different objective functions for slightly different things.

means:
minimize all points to their center of group

K-median: puts it closer to most points
K means cares about distance squared

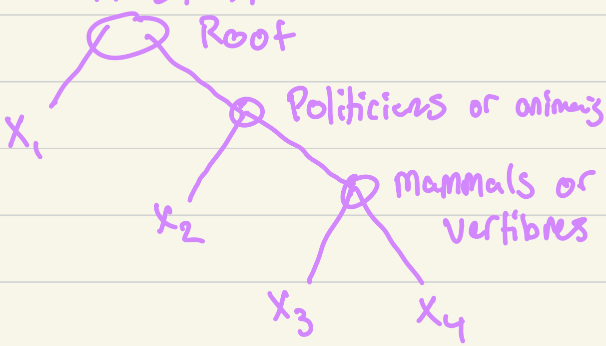
K can be any number.

Hierarchical clustering goal is to get hierarchy of data points
How do you rank nike shoes? Instead of 4 clusters for sandals, running shoes, formal, kids, etc (like graph partitions, or kmeans) we use hierarchical clustering.

How do you find the # of clusters effectively ↑ 3 vs 5 vs 100.

Hierarchical clustering

n: $x_1, x_2, x_3, \dots x_n$



what do you do to split this effectively, what objective function (having similarity score).

Q (objective function):

$$\text{cost}(T) = \max_{x_i, x_j} \left(\frac{\text{dissimilarity}(x_i, x_j)}{\#\text{ hops}} \right)$$

Everything is relative to rest of hierarchical, my idea was x_2 (adding all dissimilarity scores) compared to x_3.

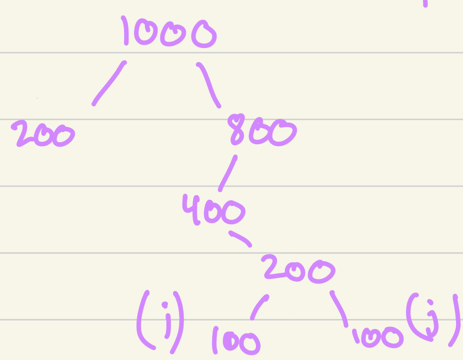
A good|popular cost function: → from STOC'16 journal

penalize if similar things (papers) were separated early on the tree versus later on the tree.

$$\text{cost}(T) = w_{i,j} \left| \frac{\#\text{ data points present when } i,j \text{ got split}}{n} \right|$$

similarity of pair $\cdot \left(\frac{200}{1000} \right)$

split @ 200



$$\text{cost}(T) = \sum_{i,j \in V} \underbrace{w_{i,j}}_{\substack{\text{similarity}\\ \text{score}\\ \text{(positive)}}} \underbrace{\frac{|\#\text{ present}|}{n}}_{\substack{\text{fraction of points}\\ \text{present in split}}}$$

Two types of clustering algorithms

- Linkage algorithms (single linkage, average linkage, complete linkage). (popular);

All points are separate, and merge documents that are highly similar. Bottom-up approach

- Divisive (top-down). Take all points together, and find the best 'cut'/'split' to separate into clusters (balance-cut, sparsest-cut).

Represent as a convex optimization problem

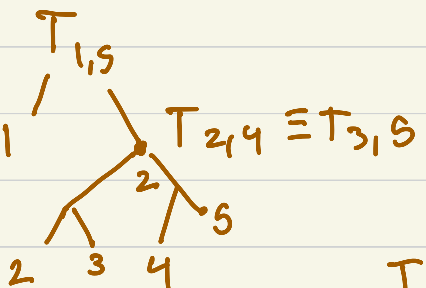
$$\mathrm{cost}(T) = \sum_{i,j \in V} w_{i,j} \left| \frac{\#\text{ points present during cut}}{n} \right|$$

Setting: $G(\overset{\text{vertices}}{V}, \overset{\text{edges}}{E}, \overset{\text{weights}}{w \geq 0})$ $\quad |V| = n$

Find (binary) tree minimize: $\mathrm{cost}(T) = \sum_{i,j \in E} w_{i,j} \underbrace{\left| \#\text{ Trees of } T_{i,j} \right|}_{\text{Penalty}}$

$$\sum_{i,j \in E} w_{i,j} |T_{i,j}|$$

↳ want the tree as low as possible when similarity is high (w).



T_{ij} is the subtree rooted at the LCA(i,j)

How do we write the objective function as vectors and use convex optimization relaxation.

(continuous optimization is easier than discrete optimization)

$$\sum_{i,j \in E} w_{i,j} \, |T_{i,j}|$$

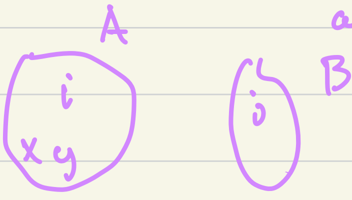
Convex Relaxations

1) Assign a boolean variable for each pair of nodes

$$x_{i,j} = \{0,1\}$$

2) (assign vectors)
All data points are now vectors w/ their features

Aside: easier example. Variable that says whether 2 points are separated or not.



$X_{ij} = 1$, $X_{xy} = \emptyset$

max-cut maximizes weight of edges cut.
↳ Output is partition to split in 2.
We're given $\{1$ if x_{ij} split from cut, $\emptyset$ if not cut$\}$

This obj function cuts it:

ϕ or 1

$$\max_{\text{assigned} \to x} \sum_{ij \in E} w_{ij} \cdot x_{ij}$$

$$x_{ij} = \begin{cases} 1 & \text{if split} \\ \phi & \text{if not split} \end{cases}$$

---

my idea to implement 4 for the main problem

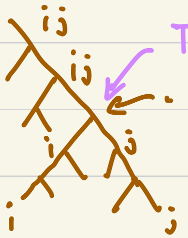
T_{ij} = subtree minimize size

• similarity

x_{ij} = split (1) nosplit (0)

$$\text{minimize}_{\text{split}} \left[x_{i,j} \cdot T_{ij} \cdot (1 - \text{similarity}) \right]$$

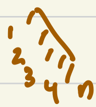
Solution:



T_{ij} = Lowest common ancestor

• lets say here i & j split here

• we need 1 variable per level to describe this pair i,j

worst case of binary tree is (n-1) , best is log(n)?

we define 'level' t →

$t = n$

$t = 1$

$x_{i,j}^{t}$ is $\{0,1\}$ if i,j are together at level t. (1 if split)

T_{ij} = Lowest common Ancestor

$t=n$	0
$t=n-1$	0
⋮	
$t=T_{ij}$	1
⋮	
$t=1$	1

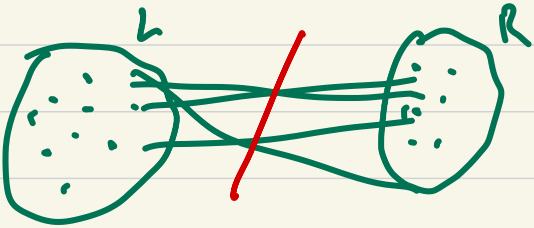
once 1 always 1.

Lecture oct 6 (late)

Talked about relaxation / NP-hard
Hierarchical clustering & semi-definite programming.

ex) graphs w/ vertices, edges, weights.

$G(V, E, w)$ given this split into L & R so the edges cut between both groups is maximized



$w \geq 0$

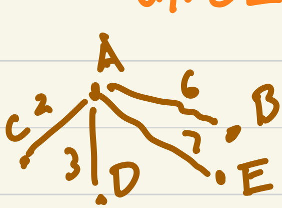
MAX-CUT

Given vertix u we know to keep on left or right side.

$A: V \to \{0,1\}$ u $\nearrow$ L, $\searrow$ R

$$\max \sum_{u,v \in E} w_{uv} \mathbb{1}\{A(u) \neq A(v)\}$$

↳ what does this & u mean?



A goes to left, C & D go left, B on right w/ E

We used to use greedy

If all edges are cut, then bipartide then:
(all separated?)

max function (sum of all weights) = 1

→ back then greedy would take A & go from there w/ opt ≥ 1/2

Even worst case on any graph would mean result in a 50% split (random split).

Bipartide graph splits everything (100%)

50% of splitting or not cutting

Semi-definite programming was to improve the random split

One idea was to rewrite MAX-CUT:

For every vertex $v \in V$: a variable $X_v \in \{0,1\}$ ← left or right

For every edge $e \in E$: $Z_e \in \{0,1\}$ ← cut edge y/N

How to rewrite using ↑

$$\max \sum_{u,v \in E} w_{u,v} \cdot Z_e$$

weight ($w_{u,v}$), cut y/n (Z_e)

This would cut everything so we need to have constraints so → is not = 1

Constraints for partitions:

1) $Z_{u,v}$ is less then $X_u + X_v$, so if both u & v

are on the same side then their connection is __not__ cut:

$z_{uv} \leq x_u + x_v$ $\quad x \in \{0,1\}$

$z_{uv} \leq 2 - x_u - x_v$

People thought to solve over $[0,1]$ interval instead of $\{0,1\}$ set.

This is called the Linear Programs (LP)

↳ linear objectives & constraints for optimal solution

what is relaxation

But bc a cut is not 0 & 1 but can be 0.7 or 0.3 then the solution a weakness:

You can set all variables (L/R) to 0.5 & all z's to 1 (cut) which gives us the same response for all graphs.

LPs are not useful for MAX-CUT / this program.

We'll try another formulation to lead us to Semi Definite programming.

Discrete Problem

$\forall$ vertex v encodes left or right.

$y_v \in \{-1, 1\}$ ← additional variable

$$\max \sum_{u,v \in E} w_{u,v} \cdot \left(\frac{1 - y_u \cdot y_v}{2} \right)$$

if same side cut $1-(1)\cdot(1)=0$ ✓ $1-(-1)(-1)=0$

if different sides then $1-(1)(-1)=2$

Then divide by $2 = 2$

For every vertex we add a vector of n dimensions

↳ we allow $y_v \in \mathbb{R}^2$, but must be unit vector (L^2 norm $= 1$) with constraint:

What is an inner product → $\vec{y}_v \cdot \vec{y}_u = 1$, $\forall v \in V$ $\|y_v\|_2^2 = 1$

$$\max \sum_{u,v \in E} w_{u,v} \cdot \left(\frac{1 - \vec{y}_u \cdot \vec{y}_v}{2} \right)$$

w/ unit vector if same then $= 1$, if diff $= -1$

These are SDPs (semi def pos).

matrix x can't be $\emptyset$

$A\vec{x} = \lambda \vec{x}$ Then λ is eigen value & $\vec{x}$ is eigen vector

we can rewrite $\vec{y}_u \cdot \vec{y}_v$ to find eigen values easily & see if pos/neg.

Gaussian elimination runs in n^3 for n matrix to find eigenvalues quickly & see if all are positive.

inner product

$A_{uv} = y_u \cdot y_v$

$\begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$

A_{uv} must have diagonals (1) & non-neg eigenvalues. (pos only.)
↳ something gives us this eff.

$$A = \begin{bmatrix} 1 & v_2 \cdot u_2 & v_1 \cdot v_3 & \\ 1 & & & \\ & 1 & & v_i \cdot u_j \\ & & 1 & \end{bmatrix}$$

$A = v_i \cdot u_j$

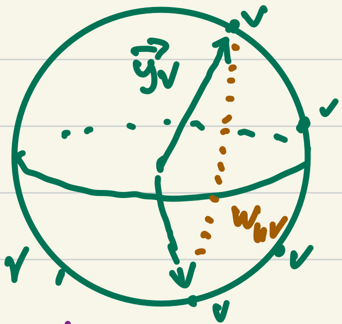
This is SDP and the best approx existing for max-cut.

anything more proves P=NP

Here it is SDP:

"embed"

Put all vertices onto unit sphere of $\mathbb{R}^n$ dimensions



if weights large then opposite sides of sphere. if opposite sides then split.

CVX solver in python

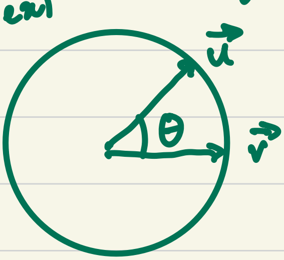
↳ og algo: Random hyperplane rounding algorithm

Take a random hyperplane to cut the sphere to cut sphere. With pos or neg norm: $>=0$ $<=0$

Algorithm $\geq \sim 0.878$ (optimal)

↳ This algorithm creates ~0.878 or better of optimal

ex)



Find $\Pr(u \text{ splits from } v)$

↳ $(2 \cdot \theta/360)$ or θ_{uv}/π

$$w_{uv} \cdot \frac{\theta_{uv}}{\pi}$$

A_{uv} from SDP is $\cos(\theta_{uv})$ so $\dfrac{w_{uv} \cdot (1-\cos\theta_{uv})}{2}$

$$w_{uv} \cdot \frac{\theta_{uv}}{\pi} \geq 0.878 \cdot \left(w_{uv} \cdot \frac{(1-\cos\theta_{uv})}{2} \right)$$

$$\Pr[i/j] = \frac{\theta_{ij}}{\pi} \geq 0.878 \cdot \frac{1-\cos\theta}{2}$$

Missed lass class, now Oct 13th.

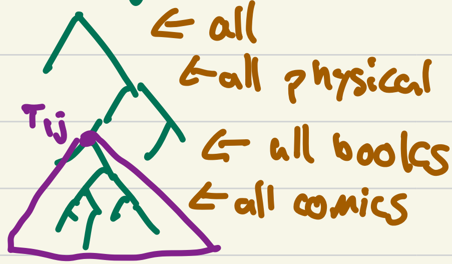
Hierarchical clustering

$G(V, E, w)$

$$\max \sum_{ij \in E} w_{ij} (n - |T_{ij}|)$$

$n =$ everything (nodes) Outside

$T_{ij} =$ inside tree



compared to Max-cut, which is how 2 are separated, Here HC, everything is eventually split.

What is an indicator function? $\mathbb{1}$

when k is separated from i, j (dolphin, cat, dog)

$$(n - |T_{ij}|) = \sum_{k \neq i,j} \left(\mathbb{1} \left(\begin{matrix} k \text{ was first to} \\ \text{separate from } i,j,k \end{matrix} \right) \right)$$

$$\max \sum_{i,j \in E} \sum_{k \neq i,j} \left(\mathbb{1} \left\{ \begin{matrix} k \text{ was first to} \\ \text{be separated among} \\ i,j,k \end{matrix} \right\} \right)$$

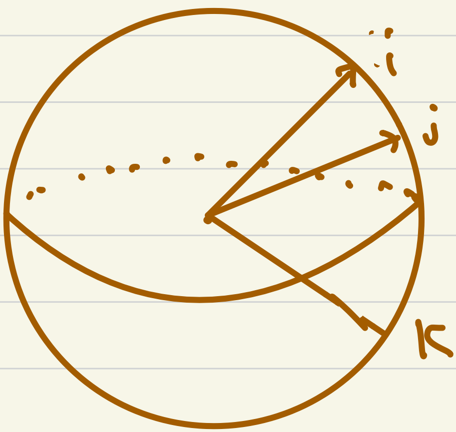
1/3 1/3

Summation over all weights

$$\text{random} = \frac{1}{3} \sum_{i,j \in E} (n-2)$$

Using similar method to max-cut SDP method

1) solve SDP for hierarchical clustering
2) Draw hyperplane through origin
3) This beats random



θ_{ij} $\Pr[k \text{ first separated}] \equiv$

θ_{ik} $\Pr[ij \mid k]$

θ_{jk}

$$\frac{\max(\theta_{ik}, \theta_{jk}) - \theta_{ij}}{\pi}$$

2·min

$$\frac{(\theta_{ik} + \theta_{jk}) - \theta_{ij}}{\pi}$$

SDP:

$$X \quad \max(\theta_{ij}, \theta_{jk}) - \theta_{ij} / \pi$$

$$\frac{(\Theta_{ik} + \Theta_{jk}) - \Theta_{ij}}{2\pi}$$ [is Prob of splitting k from i & j]

$\Pr[ij | k] = x$
$\Pr[ik | j] = y$
$\Pr[jk | i] = z$

3 by 3 linear system

$x + y = \frac{\Theta_{ik}}{\pi}$ ← i & k got separated

$y + z = \Theta_{ij}/\pi$
$x + y = \Theta_{jk}/\pi$
$x + z = \Theta_{ik}/\pi$

Semi-definite Program:

$$\text{SDP}: \quad \max \sum_{i,j \in E} \sum_{t=1}^{n} w_{ij}(1 - x_{ij}^t)$$

edges levels whether x_{ij} separated at level t.

$$x_{ij}^t = 1/2 \| v_i^t - v_j^t \|_2^2$$

↳ every vertex (i) has n vectors for each level.

After SDP we get n vectors for each vertex for

each size at most t

level t looks like:

all clusters, we've cut

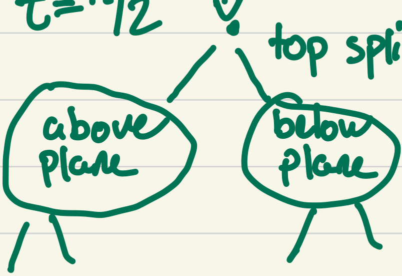
at n, we've cut a ton, and at 1 still a large amount of graph isn't split.

A good candidate is n/2 since we have already split a good amount w/o losing structure.

Then taking the n/2 → level resulting vector from SDP, we run random hyperplane approx. algorithm.

so after we look at t=n/2

top split



This results in an SDP graph.

cont. again →

This is like .3336 · opt
still better.

max cut is SDP

Max bisect algorithm is an SDP that is 0.6 · opt. each level needs to have exactly half and half resulting output of graph.

1) Run SDP 2) hyperplane rounding @ $t = n/2$

The opt soln is either:

1) $\text{Opt} < \underbrace{(1-\varepsilon)}_{\text{some } 80\% \text{ or whatever}} \overbrace{(n-2) \sum_{i,j \in E} w_{ij}}^{\text{can't cut more than edges of graph}}$

How can we beat 1/3 approx?

$$\text{Random} = \frac{(n-2)\left(\sum w_{ij}\right)(1-\varepsilon)}{3(1-\varepsilon)}$$

✓ beats opt in cases where opt is not large

in these cases just choose 1/3.

2) How about high opt solutions $\text{opt} \geq (1-\varepsilon)(n-2)\sum_{i,j \in E} w_{ij}$

Then we use what we made?

lecture oct 15

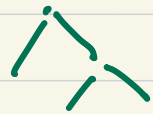
We use objective, SDP, Algos, Geometry, Analysis

Goal is to beat 1/3 greedy baselines = 1/3

This soln gets ≥ 0.336 optimal

which proves we can improve

$G(V, E, w \geq 0)$ $\quad \sum_{ij \in E} w_{ij}(n - |T_{ij}|)$

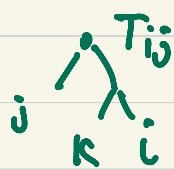


Equal to

$$\sum_{ij \in E} w_{ij} \sum_{k \neq i,j} \mathbb{1}\{k \text{ not a leaf of } T_{ij}\}$$

only if k got separated first

$$\underbrace{y_{ijk}}_{\text{random variable}} = \left[w_{ij} \mathbb{1}\{k \text{ not a leaf of } T_{ij}\} \right] \quad y_{ij} = \sum_{k \neq ij} y_{ijk}$$



Alg, (k got separated first) $\quad (i, j, k) \cdot \Pr[ij|k] = 1/3$

Random partition recursive

Alg_2 SDP first random next

1) solve SDP for hierarchical clustering
   ↳ gives vectors for all levels ?
2) Look at vectors at $\lceil n/2 \rceil = t^* - 1$.
   $x^*_{ij} = X_{ij}^{t^*}$
3) Do the hyperplane rounding for partition $(S, \bar{S})$
4) Run random always (S)

so SDP | hyperplane first, then random always (Alg_1) is a subroutine to run after. $\text{Alg}_2 \geq 0.336$ opt
$\text{Alg}_1 \geq 0.333$ opt

Max uncut bisection is a more optimal solution.

We have (spreading) and (monotonicity)

x_{ij}^t. if 1, then ij separated @ level t.
level t has <u>at most</u> clusters of size t.

SDP obj: $\max \sum_{t=1}^{n-1} \sum_{ij \in E} w_{ij} (1 - x_{ij}^t)$

spreading: 

$\sum_{j \neq i} x_{ij}^t \geq n - t, \ \forall i, \forall t$

monotonicity : $x_{ij}^{t+1} \leq x_{ij}^{t} \quad \forall ij \in E, \forall t, \quad x_{ij}^{t} = 1$

$$\text{SDPobj: } \max \sum_{t=1}^{n} \sum_{ij \in E} w_{ij} (1 - \underbrace{x_{ij}^{t}})$$

$\hookrightarrow x_{ij}^{t} = \frac{1}{2} \| v_i^{t} - v_j^{t} \|_2^2$

$$\text{OPT} < (1-\varepsilon_1) \underbrace{\left[(n-2) \sum_{ij \in E} w_{ij} \right]}_{\text{max of all weights (valid upper bound).}} \leftarrow w$$

$$\text{Random Always} = \frac{1}{3}(n-2) \sum_{i,j \in E} w_{ij} = \frac{1}{3} \frac{(1-\varepsilon_1)}{(1-\varepsilon_1)} (w)$$

so $\text{random always} > \frac{1}{3(1-\varepsilon_1)} \text{OPT}$

now when using: $\text{SDP}_{val} \geq \text{OPT} \geq (1-\varepsilon_1) w$

Analysis: Events to analyze:

understand top cut? $\rightarrow$

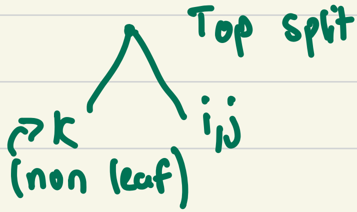
$\mathcal{E}_{ij}$ = i,j together aft. 1st cut.

$\mathcal{E}_{ijk}$ = i,j,k together aft. 1st cut

$\mathcal{E}_{ij|k}$ = ij together aft 1st cut and k seperated everyone remains together (top split didn't cut)

if k is leaf or i or j split first then this is zero

$$E[y_{ij}|k] = \frac{w_{ij}\cdot \Pr[\mathcal{E}_{ijk}]}{3} + w_{ij}\cdot \Pr[\mathcal{E}_{ij}|k]$$



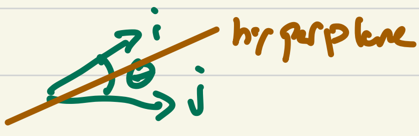
$$E[y_{ij}|k] = \frac{w_{ij}}{3}\Pr[\mathcal{E}_{ij}] + \frac{2w_{ij}}{3}\Pr[\mathcal{E}_{ij}|k]$$

prob ij not split is ijk + ij|k

so:

$$E[y_{ij}] = \sum_{k\neq ij} E(y_{ijk}) = \frac{w_{ij}}{3}\left[(n-2)\Pr[\mathcal{E}_{ij}] + 2\sum_{k\neq ij}\Pr[\mathcal{E}_{ij}|k]\right]$$

$$\Pr[\mathcal{E}_{ij}] = 1-\theta_{ij}/\pi$$



$x_{ij}^{b} = \cos\theta_{ij}$, the vertices together are

$$\theta_{ij} \le \bar{\theta}$$

$$\bar{\theta} = \arccos(1-\varepsilon_2)$$

$$\min: \sum_{k \neq i,j} \Pr[\mathcal{E}_{ij} \mid k]$$

$$\min: \frac{1}{2\pi} \sum_{k \neq i,j} \theta_{ik} + \theta_{jk} - \overline{\theta} \geq (n-2)\left(\frac{1}{4} - \frac{\overline{\theta}}{2n}\right)$$

$$\text{Subj to: } \sum_{k \neq i} \cos\theta_{ik} \leq n/2 - 1$$

$$\sum_{k \neq j} \cos\theta_{jk} \leq n/2 - 1$$

20-25 min presentations → week 8 or 9

10-15 mins technical, start w/ introduction to prob.
Don't jump directly into technical specifics.

Slides for presentation. Present paper, part of proof (in a self contained way).

We have a stochastic model:

rankings, correlation clustering, hierarchical clustering

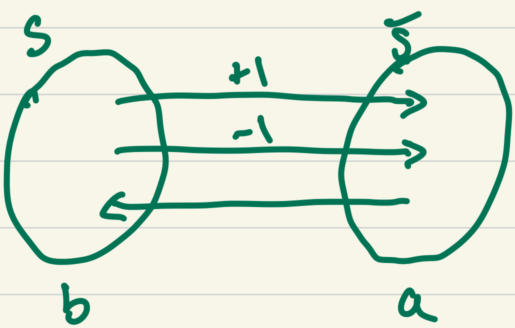
Sample 2 items: $a \, ? \, b$: which is smaller — $(1-\varepsilon)$ correct, ε wrong

3 items: $(a, b, c) \rightsquigarrow$ $1-\varepsilon$ right, ε (2 wrong answers)

For solving $a < b$, we could set it up as

$$\begin{tikzcd} a \arrow[r, shift left, "+1"] & b \arrow[l, shift left, "-1"] \end{tikzcd}$$

lets setup weighted / directed max-cut:



we will only care about left to right edges, Then include both weights

Alg: max cut first on S & $\bar{S}$, then random permutations of S & $\bar{S}$.

random (L) random (R)

Maximize satisfied constraints is goal, how does W(L,R) get to that?

max # SAT constraints.

graph has +/- 1, and directed.

We have a total of m constraints.

Beyond worst case

m_s is satisfied by first cut (L,R)

m_v violated by L,R $\quad -1 \quad +1$

m_u unaffected

weight of cut $W(L,R) = m_s - m_v$

$$Alg = m_s + \tfrac{1}{2} m_u$$

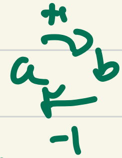
$$= m_s + \tfrac{1}{2}(m - m_s - m_v)$$

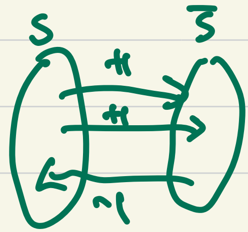
weight of any cut in graph

$$= \tfrac{1}{2}m + \tfrac{1}{2} m_s - \tfrac{1}{2} m_v = \frac{m}{2} + \frac{1}{2}\underbrace{W(L,R)}_{\text{np hard Problem}}$$

opt cut ≥ median cut ≥ $m/2$

How about approx alg?





The graph (total) has ∅ avg

We're doing Max-Cut w/ Directed & negative weights.

(typically)

E(SDP + hyperplane rounding) ≥ 0.878 optimal

• One solution, typically bad but good here, we can add a coeff to make everything positive (no neg weights).

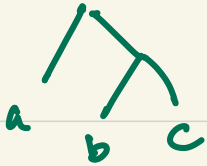
Only for regular max cut,

for directed + neg → $E\left(\begin{smallmatrix}\text{SDP}\\ \text{hyper round}\end{smallmatrix}\right) \geq .857\,\text{opt} - 0.143\,|w^-|$

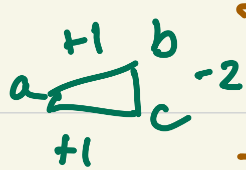
if un-directed then

$$E\left(\begin{smallmatrix}\text{SDP}\\ \text{hyp}\end{smallmatrix}\right) = 0.878\,\text{opt} - 0.122\,|w^-|$$

$$\geq 0.857\,\text{opt} - 0.143\,m \geq \underbrace{\frac{0.857\,m}{2} - 0.143\,m}_{} = \frac{.571\,m}{2} = .286\,m$$



For these cases encoded as:



total weight (always) need to be ϕ.

Alg should be $\geq 1/3\, m$ (random)

$w(L, R) = 2m_s - m_v$

$Alg = m_s + 1/3(m - m_s - m_v)$

$= 1/3\, m + 1/3(2m_s - m_v)$

$= 1/3\, m + 1/3\, w(L, R)$

Nov 5th → wk 6

→ wk 8?

Nov 17th Monday (ian soham, Adi)

~ 20 minutes with slide decks

Draw graphs too

show actual self-contained proofs

How Hard is Inference for Structured Prediction

describe in our own words ↳ vaggos's advisor.

Embeddings, preserve comparison.

$\delta(x, y) < \delta(w, z)$

This is enough.

we can embed points in $\in \mathbb{R}^d$ where $d \leq n$

& $n/2 \leq d \leq n$

Euclidean norm assumed

$\|\ldots\|_2 = \sqrt{x_1^2 + \ldots x_d^2}$

$\|\varphi(x) - \varphi(y)\|_2 <$?

$\|\varphi(w) - \varphi(z)\|_2$

$\|x\|_1 = |x| + |x_2| \pm \ldots |x_d|$

↳ L1 norm

↳ conv. way to write is $\operatorname{sign}(\vec{x}) \cdot \vec{x}$ ↑ +−

$\|x\|_\infty = \max_{i \in [d]} |x_i|$

L∞ norm

$(1, -1) \cdot (x_1, x_2)$

↳ $x_1 - x_2$

↳ so to preserve data we need minimum $n/2$ dimensions & n max dim.

The # ordinal embeddings is less than potential things trying to capture ~~n~~ and less than $n/2$

Terminal embeddings

(ord. embed.) (VIP)

How do we keep only the important embed. / nodes.

$T = \{t_1, t_2, \ldots t_k\}$ Terminal nodes

↳ k terminals

$V \setminus T = \{v_1, v_2, \ldots v_{n-k}\}$

all other nodes

we want to keep T and preserve those over $V \setminus T$.

↳ $(t, ?) < (t', ?)$ we want to compare distances of this form.

↓ any node

$t, t' \in T$

Upper bound is k dimensions, these are our VIP nodes (lets say k is small)

if everyone is VIP, you need to preserve every comparison.

→ terminal nodes

How do we preserve key embeddings?

↳ by putting each one in its own dimension w/ L2 norm

Idea: Every vertex $v(\underset{\omega}{\cdot}, \cdot, \cdot, \ldots)$ k coordinates

each coordinate is dedicated to a terminal node

terminal 1: $t_1 (1, 0, 0 \ldots 0)$

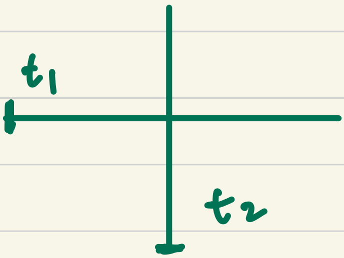
$t_2 (0, 1, 0 \ldots 0)$

instead of doing 1 do $-M$ (large negative)

$$t_1\,(-M, 0, 0, \dots 0)$$
$$t_2\,(0, -M, 0, 0 \dots 0)$$

Pick $M = k^3 n^2$ (large number)

$t_i = -M\vec{e}_i$ lets say k is pretty small.



Creating embeddings based on 'Rank' of vertex to terminal (1 if closest, 2 if 2nd).

$$\text{rank } r\{\overset{\text{term}}{t}, \overset{\text{vert}}{v}\} \in \{1, 2, \dots, k \cdot n\}$$

specifying order in which distance t, v appears w.r.t other vertices.

So what is the embedding for a vertex? (not terminal)

$$\varphi(v) = \left(r(t_1, v), r(t_2, v), \dots r(t_k, v)\right)$$

compare:

Pair (t, v) & (t', v')

↳ $\|\varphi(t) - \varphi(v)\|_2^2 = \sum_{i \neq t} r(i, v)^2 + \left(r(t, v) + M\right)^2$

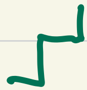
$i = t$

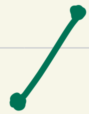
$[0, 0 \dots, -M, 0 0 \dots]$

$$= \sum_{i}^{n} r(i,v) + 2M \cdot r(t,v) + M^2$$

so we

$$\sum_{i} r(i,\tilde{v}) + 2M \cdot r(t',v') + M^2$$

L_1 = manhattan → 

L_2 norm = euclidean ⇒ 

Application: Finding L_1 diameter of a set P of points $\in \mathbb{R}^d$

↪ $x_1, x_2, x_n \in \mathbb{R}^d$

Find pair $p,q \in P$ where $\|p-q\|_1 = \max_{p,q} \|p'-q'\|_1$

Naive is checking all diameters w/ distance calc:

(d is small, n is large) Naive: $O(n^2 \cdot d)$

Better: $O(n \cdot d \cdot 2^d)$

each $f_s(p) = \vec{s} \cdot \vec{p}$

$f(p_1) = [\ \ldots\ldots\ ]$

$f(p_2) = [\ \ldots\ ]$

$\vdots$

$f(p_n) = \underbrace{[\qquad\qquad]}_{2^d \text{ coords}}$

? isometry ?

$\vec{s} = \{-1, +1\}^d$

$$\|p-q\|_1 = \|f(p) - f(q)\|_\infty$$

$f_s(p) = \vec{s} \cdot \vec{p}$

$L_1 \to L_\infty$

isometry from $L_1^d \to L_\infty^{2^d}$

$f(p_1) = [\ \ldots\ldots\ ]$

$f(p_n) = [\ \ldots\ldots\ ]$

Look @ only 1 coordinate & do max minus min & compute for all.

---

ordinal embedding:

Taking embeddings from 1 space to another while maintaining connections b/w nodes.

matrix space (euclidean dist. space or tree space)

Let $X = ([n], \delta)$ be any matrix space. φ = phi

We say $\varphi : X \to \mathbb{R}^d$ is an ordinal embed. if for every $x, y, z \in X$ we have:

$$\delta(x,y) < \delta(z,w) \iff \|\varphi(x) - \varphi(y)\| < \|\varphi(w) - \varphi(z)\|$$

Distortion: if 1 dist. becomes 17 dist, then 17 is distortion.

$$\varphi(x) \;\overset{1}{\rule{4em}{0.4pt}}\; \varphi(y)$$

$$\varphi(z) \;\overset{17}{\rule{6em}{0.4pt}}\; \varphi(w)$$

Def: Ordinal relaxation: A relaxation of 10 means a dist mult. by 10 is still less than other distance, then keep & dont mess it up:

$$\overset{10}{\alpha}\,\delta(i,j) < \delta(k,l) \Rightarrow \delta'(i,j) < \delta'(k,l)$$

'significantly different distances should be preserved'

Some constraints we $n >$ dimensions $> n/2$ for $X = ([n], \delta)$

? & # of ord embed of d vs # distinct metrics $\binom{n}{2}$

if you go from higher dim to lower, we want α to be larger to have less constraints

relaxation balances dimensions & points of comparison.

if $\alpha = 1$ then same, higher allows for some info lost.

Theorem trade offs a vs d

For every int d, every int n:

$\exists$ Matrix space T on n points 'there is a matrix space'

such that <u>the triplet relaxation</u>

↳ special case (i,j) vs (i,k)

of any ordinal embedding of T into d dimension euclidean space.'

↓ matrix space

is atleast:

$$\frac{\log n}{\underbrace{\log d + \log(\log n)}_{} + 8} - 1$$

any embedding will have this loss 'max relaxation'. worst case bound.

↳ getting rid of this is still an open problem?

we want to preserve this relationship: in the new space

we will screw up some embeddings

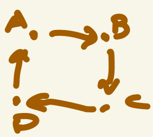
How do we prove this?

1) we need a family of significantly different metrics spaces

we dont have constraints on close stuff like w/i $\log(n)$ or something?

2) # ord embed of dim d $\ll$ # family of sig. diff metrics spaces

Girth G of a graph is shortest cycle



edges in shortest cycle, here is 4. A triangle is 3.

in a pentagon 5  if it has a triangle in it then $\underline{3}$: 

it's important to construct high girth graphs to solve/make this.

Delete edges to make a big family of high girth graphs. Like before A to B was 1, after edge gone, then $g-1$.

it doesn't have to be a,b it could be x,y. we count probability of getting higher girth cycles after cut. we choose graph to be high girth.

vertex
nodes, edges

1) Pick high girth graph $G(V,E)$ $|V|=n$

↳ 'how dense can you make graph while still being high girth.

$$m = \underbrace{|E| \geq \tfrac{1}{4} n^{1+1/g}}_{100 \geq 1/4 \; 20^{1+1/g} \;?}$$

g: girth, E is edges.
↳ shortest cycle.

Pick g: $\dfrac{\log n}{\log d + \log\log n + 8}$ then $|E| > 16 \cdot n \cdot d \cdot \log n$

idk why → (atleast $nd \log n$ edges & has high girth)

2) Subsample edges of G:

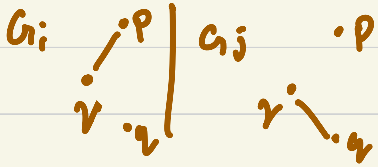
large # N: $[G_1, G_2, \dots G_N]$

(*) ← this sub-property we're making

$\forall\, G_i, G_j : \exists\, v \in V : E_{G_i}(v) \setminus E_{G_j}(v) \neq \emptyset$ and $E_{G_i}(V) / E_{G_j}(V) \neq \emptyset$

↳ for every pair — there exists

what this means is when is v unhappy if we go from G_i to G_j:



in G_i vp is 1

in G_j v_q is 1.

How many G do we choose? to high is hard for ↑

to low doesn't guarantee:

ord embed of dim d $\ll$ # family of sig. diff metrics spaces

so we sample 1/2 of each edge

$G_1 : m$

$G_2 : m$

$\vdots$

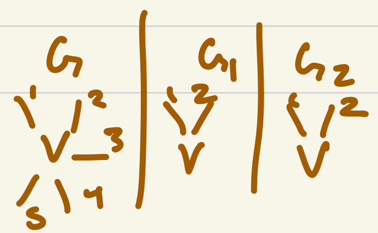
$\underbrace{G_N}$

N pick $N = 2^{bm}$, for $b < \frac{1}{2} \log_2 \frac{4}{3}$

Proof: simplifying assumptions:

1) k-regular graph G

2) independence on v



being a witness for G_j : G_i being for any ? α

Ordinal embeddings → n points

distance(a,b) < distance(a,c)

emb $F \in \mathbb{R}^d$ $\quad \|F(a)-F(b)\| < \|F(a)-F(c)\|$

lets take case of contrastive triplets. m triplets (anchor, pos, neg)

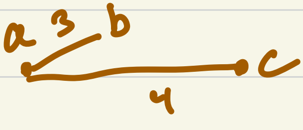
Constraints: m triplets of this form

Realizable instances: no paradoxes like $a > b > c > a$

→ Focus on this.

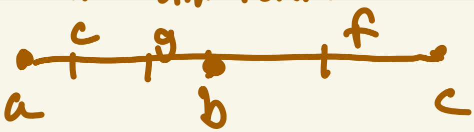
Goal: do this embedding for all triplets w/ dim $O(\sqrt{m})$

we can always do w/ $O(m)$ or $O(n)$ so how?



1) Construct a graph: Tells how ~~triplets~~ points are related to each other. So if I have ⋰ then add a point, I need to ensure constraints are still related.

(a, b, c) → embed points
(e, g, f)

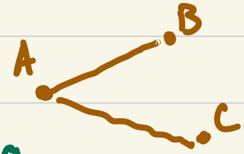


B is fixed, since we need to ensure points are violated (realizable instances).

$\| \|_2^2 \rightarrow$ euclidean.

undirected & unweighted

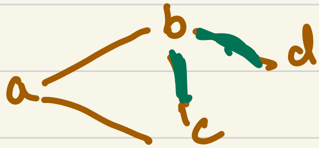
1) (a, b, c)



construct a dependency graph, ~~pet~~ ~~set~~, of n vertices.

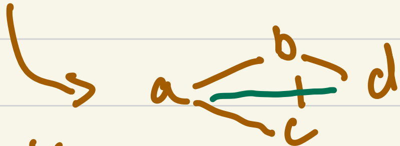
Then place in (b, c, d)

bc, bd



Then place in (a, b, d)

ab, ad

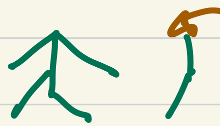


→ edges

$|E| = 2m$ edges.

Arboricity of G: The minimum # of forests in which edges can be partitioned (density).

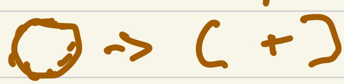
Forest - disconnected tree.



if we get arboricity of 4, then $\text{emb}^F \in \mathbb{R}^{d = 4r}$

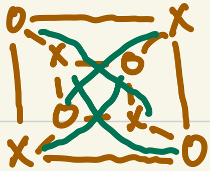
Example: Tree $r = 1$ (how many trees to overlay to get).

Cycle $r = 2$



The bipartide klick of 4,4 nodes: $K_{4,4}$

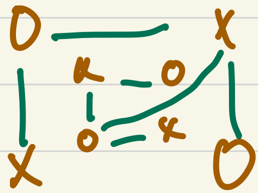
klick on circles vs x's.



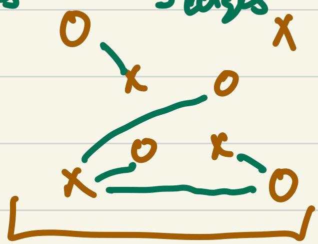
$K_{4,4}$

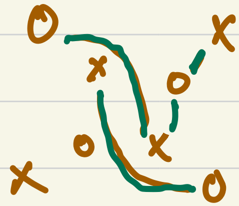
↳ at most 7 edges so $|Tree| \le 7$

(ii)



Tree (i)





Does not have to be connected, can be 2 trees or 1 forest (acyclic graph)

Tree can have ≤ 7 edges or else it would be cyclic. we want a forest which is okay being disconnected, but not cyclic.

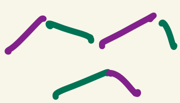
if aboricity low, then tree like, high means likely cycles or dense.

Aboricity: maximize s $\left\{ \dfrac{|\text{Edges in subgraph } s|}{|\text{Vertices in subgraphs } s| - 1} \right\}$

s = subgraphs of G

$$\max_s \left\{ \frac{|E_s|}{|V_s| - 1} \right\}$$

Colorable :



⇐ 2 colorable (bc not cyclic).

Facts: 1) G of arb. $r \rightarrow$ is $2r$ colorable

2) $r \leq 2\sqrt{|E_G|} \leq O(\sqrt{m})$

3) $\exists$ ordering $x_1\ x_2\ x_3 \ldots x_n$

There exists a

Such that every node has at most

$$|N^-(x_i)| \leq 2r-1, \ \forall i$$

↳ backwards ordering

where every previous node has at most $(2r-1)$ arboricity when looking back

$$\|F(a)-F(b)\|_2^2 < \|F(a)-F(c)\|_2^2$$

$$\|F(a)\|^2 + \|F(b)\|^2 - 2F(a)F(b) < \|F(a)\|^2 + \|F(c)\|^2 - 2F(a)F(c)$$

1) Construct a graph

2) Linear algebra trick

By setting to unit circle then

$\|F(b)\|^2 \rightarrow$ goes to 1, same with $\|F(c)\|^2$

$$1 - 2F(a)F(b) < 1 - 2F(a)F(c)$$

$$(x_1\ y_1) \cdot (x_2\ y_2) = x_1 x_2 + y_1 y_2$$

$$F(a)\cdot F(b) < \underbrace{F(a)\cdot F(c)}_{=\binom{n}{2}}$$

I got lost.

$$F: V \to \mathbb{R}^{4r} : \quad F(x) = (\underset{2r}{\hat{x}}, \underset{2r}{\mathring{x}})$$

if dependency graph: $x \quad y$

$\hat{x}\cdot\hat{y} \approx$ rank of distance $d(x,y)$

order $\{x,y\}$ pairs in descending order, in terms of distance

if a liner system has no solution, it is inconsistent. Being liner system, it has a det $= \emptyset$ and therefore adding slight noise (fiddles) to each of the variables|polynomials will make a solution since it will 'escape the roots'.

$$x_1 \quad x_2 \quad x_3 - x_4 \quad x_5 - x_6$$

inner prod (like 3,4) is given by rankings.