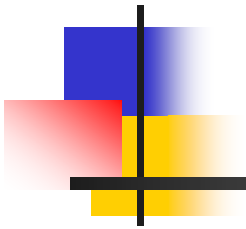
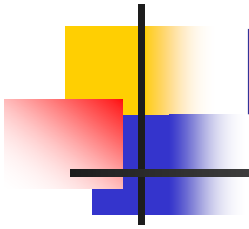


Streaming Algorithms for Geometric Problems

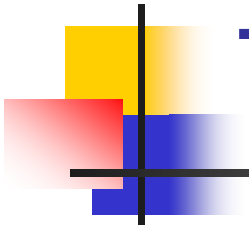
Piotr Indyk

MIT



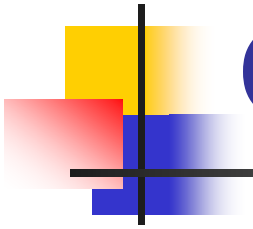
Data Streams

- A data stream is a (massive) sequence of data
  - Too large to store (on disk, memory, cache, etc.)
- Examples:
  - Network traffic (source/destination)
  - Sensor networks
  - Satellite data feed, etc.
- Approaches:
  - Ignore it
  - Develop algorithms for dealing with such data



Talk Overview

- Computational model
- Example problems
- (Short) history of streaming algorithms
- Streaming algorithms for geometric problems
  - Insertions only
  - Insertions and deletions
- Open problems

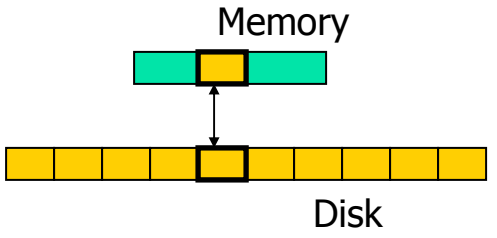


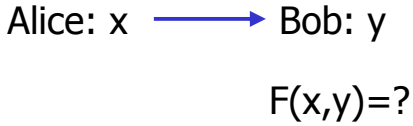
Computational Model

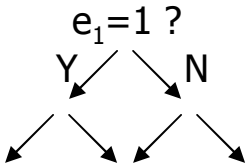
- Single pass over the data: $e_1, e_2, \ldots, e_n$
- Bounded storage
- Fast processing time per element

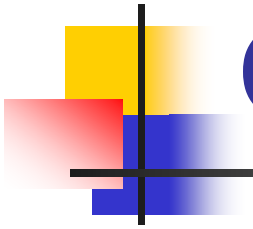
Related Models

- External Memory:
  - Bounded Storage
  - Data Stored on Disk
  - Random Access to Blocks of Data
- Compact Representations of Data and Communication Complexity
- Read-Once Branching Programs





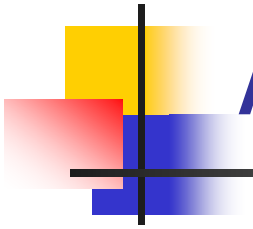




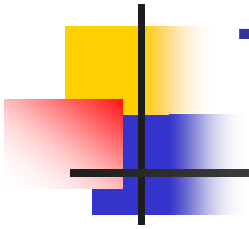
Classic Examples

- Compute the number of distinct elements:
  - Exactly: $\Omega(n)$ bits of space
  - $(1+\varepsilon)$ -approximation: $O(1/\varepsilon^2 * \log n)$ bits
    [Flajolet-Martin, JCSS'85] ,…
- Compute the median
  - Exactly: $\Omega(n)$
  - $(50\% \pm \varepsilon)$ -approximation: $O(1/\varepsilon * \text{polylog } n)$
    [Paterson-Munro, TCS'80] ,…

Brief History of Streaming Algorithms



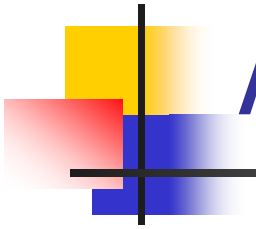
- Ancient times [MP'80,FM'85,Morris,..]
- Middle Ages
- Renaissance [Alon-Matias-Szegedy, STOC'96]
  - Theory
  - DB (Aqua project in Bell Labs)
  - Networking
  - …
- Streaming became mainstream J



Theoretical History

- Vector problems:
  - Stream defines an array of numbers
  - Maintain stats of the array, e.g., median
- Metric problems
  - Clustering
- Graph problems, Text problems
- Geometric Problems [this talk]

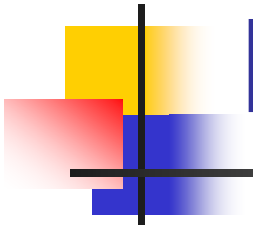
Geometric Data Stream Algorithms as Data Structures



- Data structures that support:
  - Insert(p) to P
  - Possibly: Delete(p) from P
  - Compute(P)

- Use space that is sub-linear in |P|

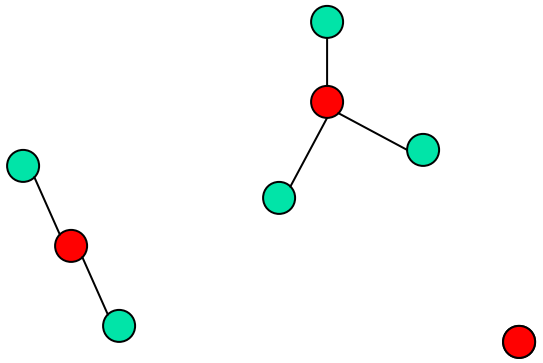
Insertions-only



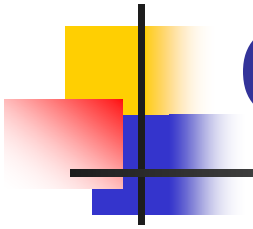
Metric clustering problems

- k-center [Charikar-Chekuri-Feder-Motwani, STOC'97]
- k-median [Guha-Mishra-Motwani-O'Callaghan, FOCS'00, Meyerson, FOCS'01, Charikar-O'Callaghan-Panigrahy, STOC'03]
- Bounds:
  - Poly(K,log n) space
  - O(1)-approximation

k-median/k-center



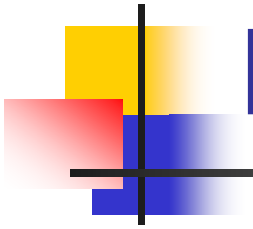
- k is given
- Goal: choose k medians/centers to minimize:
  - k-median: the sum of the distances
  - k-center: the max distance



Geometric Problems

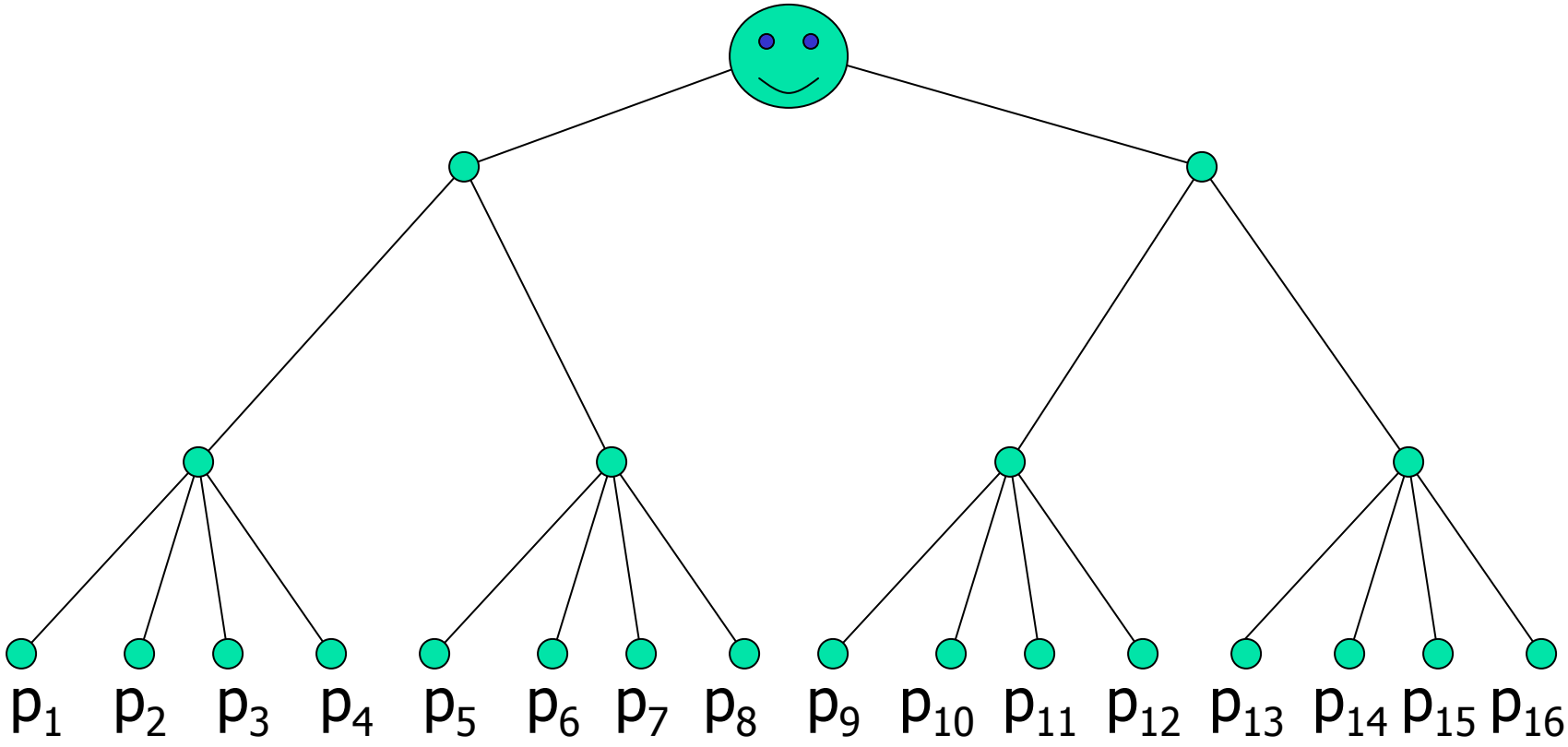
- Diameter, Minimum Enclosing Ball
  [Agarwal-Har-Peled, SODA'01, Feigenbaum-Kannan-Zhang'02 (Algorithmica), Hershberger-Suri, PODS'04]
- K-center [AHP, SODA'01]
- K-median [Har-Peled-Mazumdar, STOC'04]
- Range searching via ε-approximations:
  - [Suri-Toth-Zhou, SoCG'04]
  - [Bagchi-Chaudhary-Eppstein-Goodrich, SoCG'04]

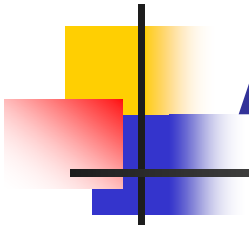
Dominant Approach: Merge and Reduce



- Main ideas:
  - Design an (off-line) algorithm that computes a "sketch" of the input
    - Small size
    - Sufficient to solve the problem
  - A sketch of sketches is a sketch

Tree Computation



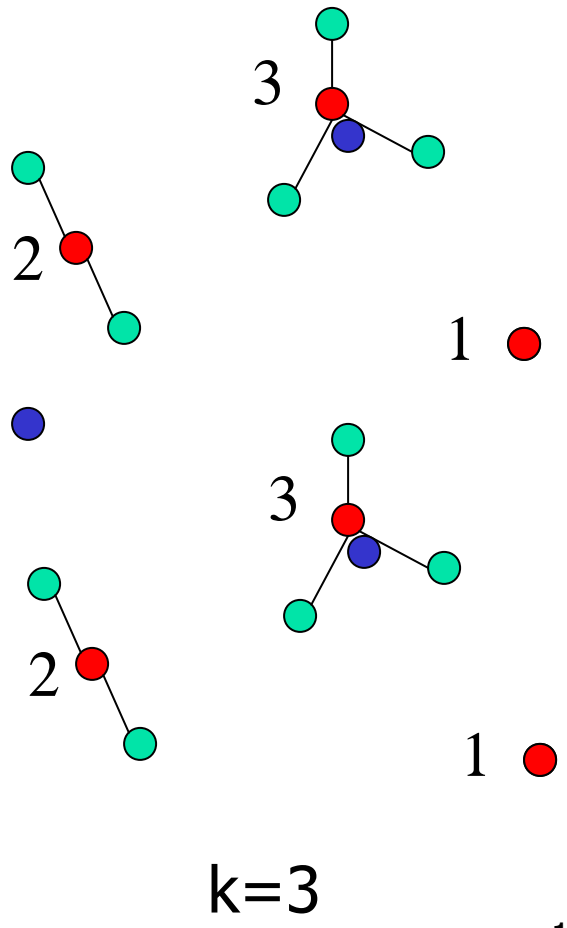


Algorithm

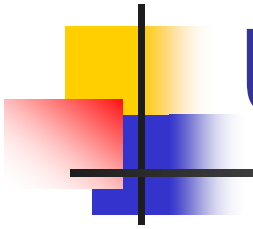
- Space: (sketch size)*log n
- Time: sketch computation time
- Question: Where do sketches come from ?

Idea I: solution=sketch

- Consider k-median
- [GMMO'00] : approximate k-median of approximate weighted k-medians is an approximate k-median
- Result:
  - Constant depth tree
  - Space: kn^{α} , $\alpha>0$
  - O(1) -approximation
  - Works for any metric space



Use the solution, ctd.



- ε-Approximations: find a subset $S \subseteq P$, such that for any rectangle/halfspace/etc R,

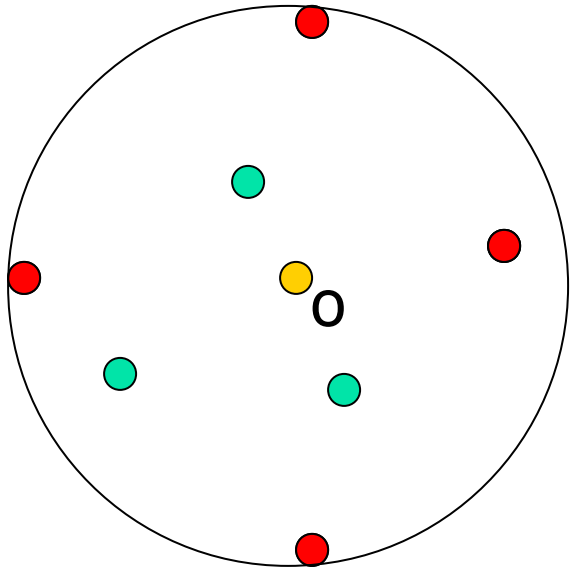
$$|R \cap S|/|S| = |R \cap P|/|P| \pm \varepsilon$$

- [Matousek] : approximation of a union of approximations is an approximation
- [BCEG'04] : convert it into streaming algorithm, applications
  - $\approx 1/\varepsilon^2$ space
- [STZ'04] : better/optimal bounds for rectangles and halfspaces

Idea 2: Core-Sets [AHP'01]

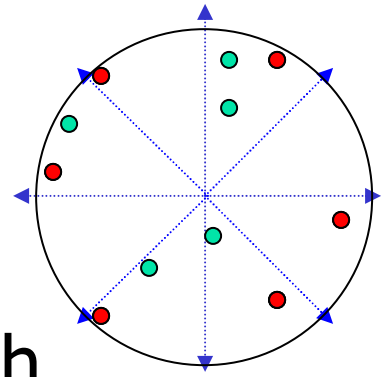
- Assume we want to minimize $C_P(o)$
- $S \subseteq P$ is an ε-core-set for P, if for any o, and a set T:

  $C_{P \cup T}(o) < (1+\varepsilon)\, C_{S \cup T}(o)$
- Note: this must hold for all o, not just the optimal one

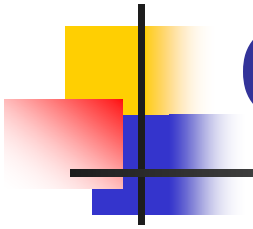


Example: Core-set for MEB

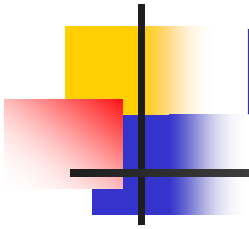
- Compute extremal points:
  - Choose "densely" spaced direction v_1 ...v_k
  - I.e., for any u there is v_i such that $u*v_i \geq ||u||_2 / (1+\varepsilon)$
  - For each direction maintain extremal point
- $k=O(1/\varepsilon)^{(d-1)/2}$ suffice



Stream Algorithms via Core-sets



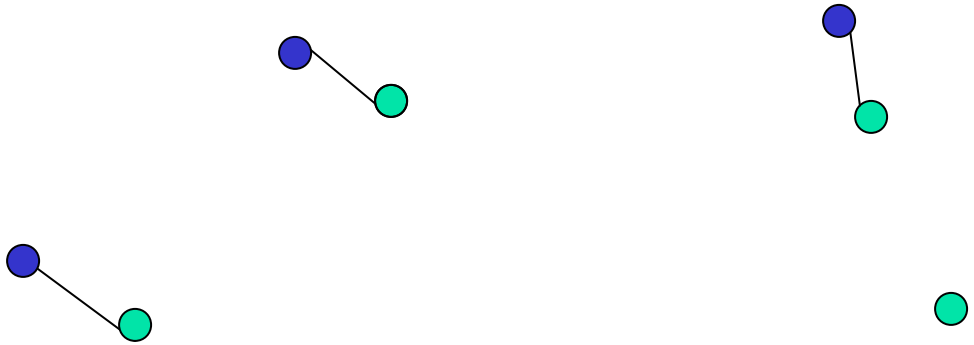
- Diameter/MEB/width: $O(1/\varepsilon)^{(d-1)/2}$ log n space [AHP'01]
- k-center: $O(k/\varepsilon^d)$ log n [HP'01]
- k-median: $O(k/\varepsilon^d)$ log n [HPM'04]
- Faster algorithms and other results: [Chan, SoCG'04], [Suri-Hershberger'03]



Limitations

- Small core-sets might not exist (see next slide)
- Do not support deletions

Minimum Weight Bi-chromatic Matching



- Estimate the cost of MWBM

Insertions and Deletions

Streaming Algorithms for Vector Problems

- Norm estimation:
  - Stream elements: (i,b) , i=1…m
  - Interpretation: $x_i = x_i + b$
  - Want to maintain $||x||_p$
- Why ? Examples:
  - $||x||_p^p = \Sigma_i x_i^p =$ #non-zero elements in x, as $p \rightarrow 0$
  - …

Dimensionality reduction

- L_2: Johnson-Lindenstrauss Lemma:
  - x is an m-dimensional vector
  - A is a random m times k matrix, each entry independently drawn from e.g. Gaussian distribution, $k=O(\log N/\varepsilon^2)$
  - Then with probability 1-1/N

$$||x||_2 \leq ||Ax||_2 \leq (1+\varepsilon)||x||_2$$

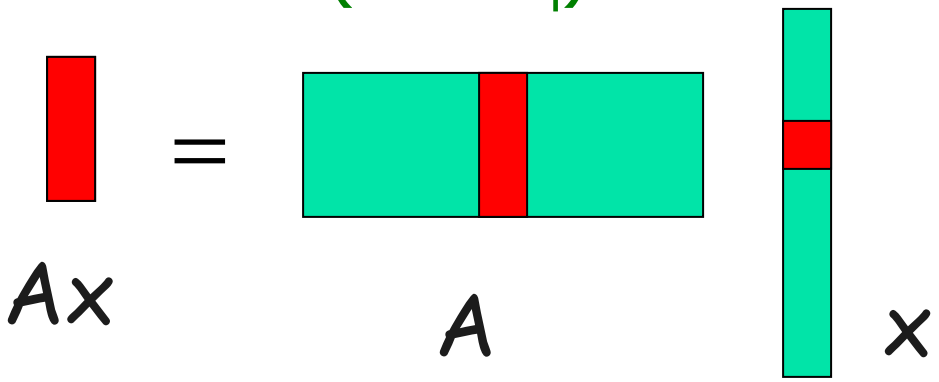
  - A can be pseudo-random [AMS'96]*

*Using slightly different method for norm estimation

What it means

- To know $||x||_2$, suffices to know Ax
- Can maintain Ax when the coordinates are incremented:

$$A(x+ be_i)=Ax+ bA\ e_i$$

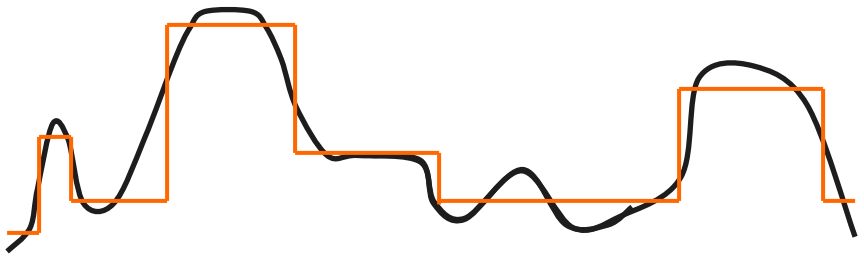


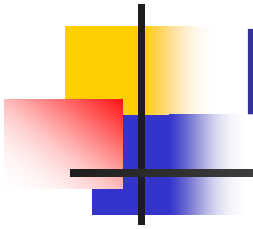
- Can maintain approximate L_2-norm of x
- Similar approach works for $p \in (0,2]$

[Indyk, FOCS'00]

Histograms

- View x as a function x:[1…n] $\rightarrow$ [1…M]
- Approximate it using piecewise constant function h, with B pieces (buckets)
- Problem can be formulated in 2D as well (buckets become rectangular tiles)



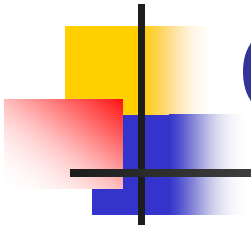


Results: 1D

- [Gilbert-Guha-Indyk-Kotidis-Muthukrishnan-Strauss, STOC'02]:
  - Maintains h with B pieces such that
    $$||x-h||_2 \leq (1+\varepsilon)||x-h_{OPT}||_2$$
  - Under increments/decrements of x
  - Space: $\text{poly}(B,1/\varepsilon,\log n)$
  - Time: $\text{poly}(B,1/\varepsilon,\log n)$

Results: 2D

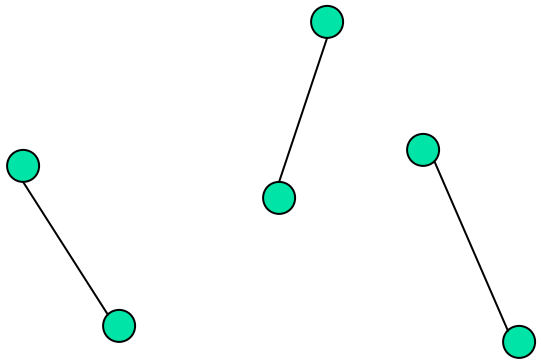
- [Thaper-Guha-Indyk-Koudas, SIGMOD'02] :
  - Maintains h with B log (nM) tiles such that
    $$||x-h||_2 \leq (1+\varepsilon)||x-h_{OPT}||_2$$
  - Under increments/decrements of x
  - Space/Update time: poly(B,1/ε,log n)
  - Histogram reconstruction time: poly(B,1/ε, n)
- [Muthukrishnan-Strauss, FSTTCS'03] :
  - Maintains h with 4B tiles
  - Time: poly(B,1/ε, log(nM))



General Approach

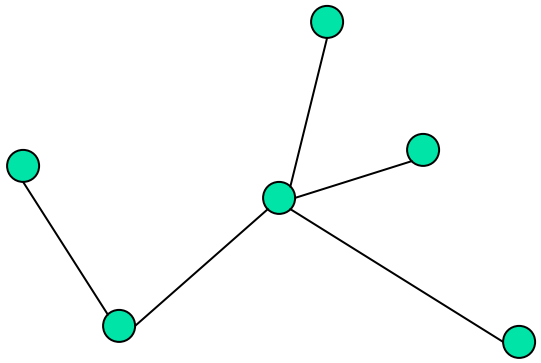
- Maintain sketches Ax of x
- This allows us to estimate the error of any given h, via ||x-h|| ≈ ||Ax-Ah||
- Construct h:
  - Enumeration
  - Greedy
  - Dynamic Programming

Minimum Weight Matching



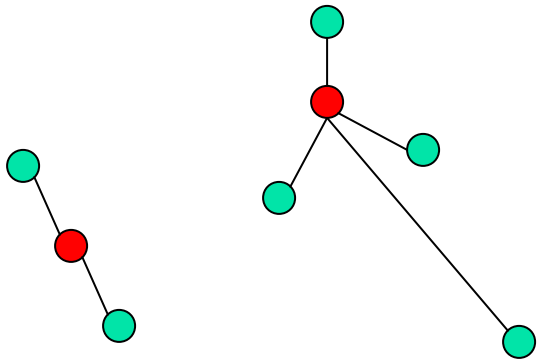
- Estimate the cost of MWM

Minimum Spanning Tree



- Estimate the cost of MST

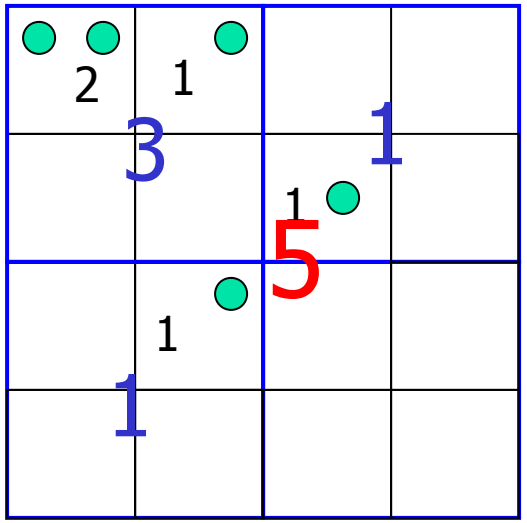
Facility Location



- Goal: choose a set F of facilities to minimize the sum of the distances to nearest facility plus the number of facilities times f
- Again, report the cost

Approach

- Assume $P \subseteq \{1 \ldots \Delta\}^2$
- Reduce to vector problems
- Impose square grids $G_0 \ldots G_k$, with side lengths $2^0, 2^1, \ldots, 2^k$, shifted at random.
- For each square cell c in G_i, let $n_P(c)$ be the number of points from P in c.
- The algorithms will maintain certain statistics over $n_P(.)$, which will allow it to approximately solve the problems



Estimators

- MST: $\sum_i 2^i \sum_{c \in G_i} [n_P(c)>0]$
- MWM: $\sum_i 2^i \sum_{c \in G_i} [n_P(c) \text{ is odd}]$
- MWBM: $\sum_i 2^i \sum_{c \in G_i} |n_G(c)-n_B(c)|$
- Fac. Loc.: $\sum_i 2^i \sum_{c \in G_i} \min[n_P(c), T_i]$
- K-median: $\sum_i 2^i \sum_{c \in G_i - B(Q, 2^i)} n_P(c)$
  (const. factor)

Maintain #non-zero entries in n_P [FM'85]

Maintain L_1 difference [I'00]

Results [Indyk'04]

Problem	Appr.
MST	$\log \Delta$
MWM	$\log \Delta$
MWBM*	$\log \Delta$
Fac.Loc.	$\log^2 \Delta$

Space: $(\log \Delta + \log n)^{O(1)}$

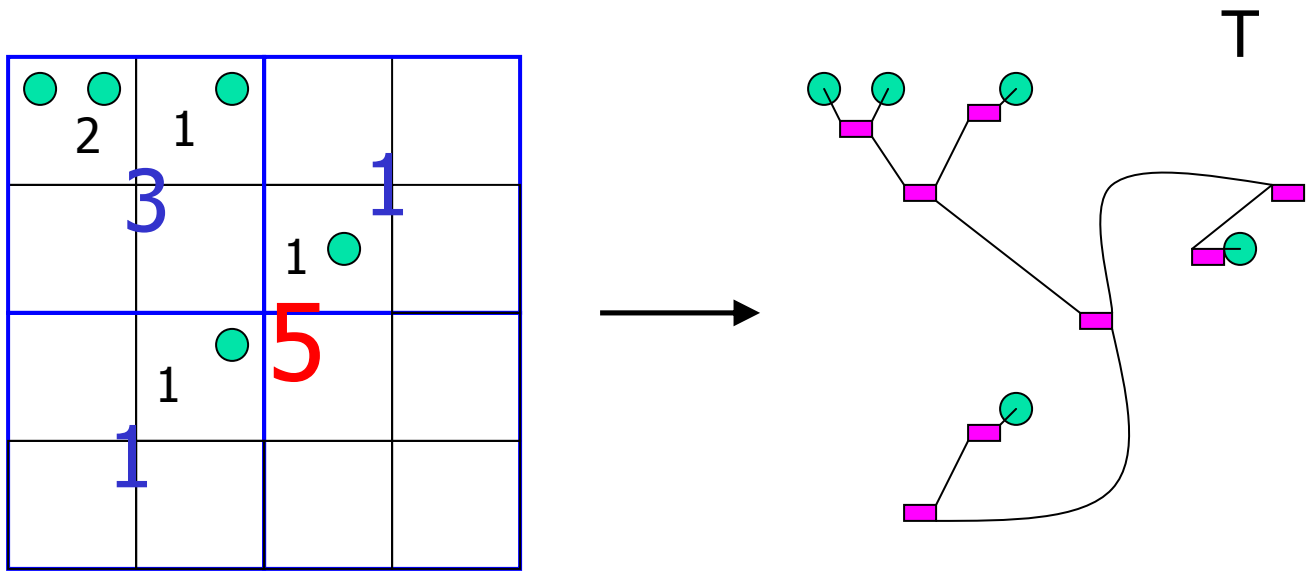
*follows from Charikar, STOC'02; also Agarwal-Varadarajan, SoCG'04 and Indyk-Thaper'02

Results: K-median

Computation Time	Approximation
$\Delta^{O(k)}$ poly(log n+1/ε)	1+ε
Δ^2 poly(log n+log +k)	O(1)
poly(log n+log Δ +k)	[1+ε , log n log Δ]

Space: $(K+\log + \Delta \log n)^{O(1)}$

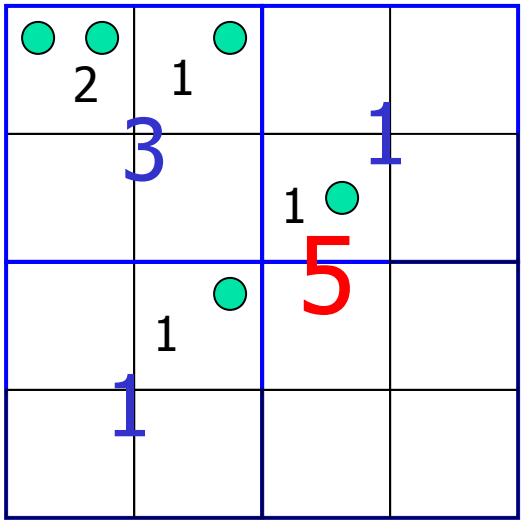
Probabilistic embeddings into HST's

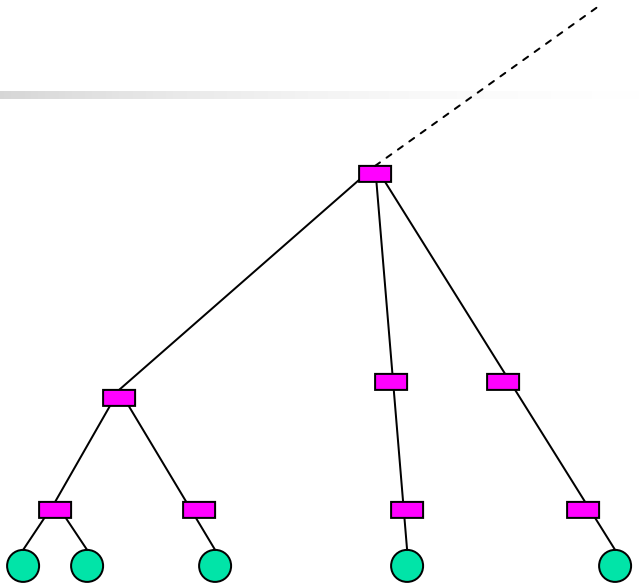


Known [Bartal, FOCS'96, Charikar-Chekuri-Goel-Guha-Plotkin,STOC'98]:

- $||p-q|| \le D_{tree}(p,q)$
- $E[\ D_{tree}(p,q)\ ] \le ||p-q|| * O(\log \Delta)$

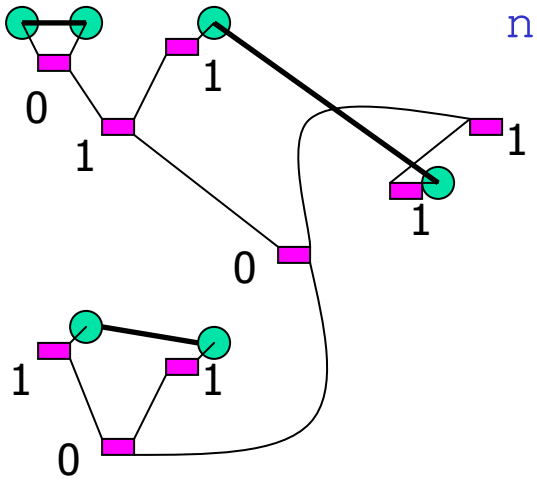
MST



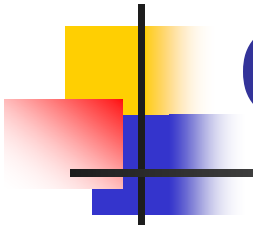


- E[Cost(MST in T)] ≤ O(log Δ) Cost(MST)
- Cost(MST in T) ≈ Cost(T)
- How to compute Cost(T) ?
  - Sum over all levels i, of the #nodes at i, times 2^i
  - Node c exists iff $n_i(c)>0$

Matching

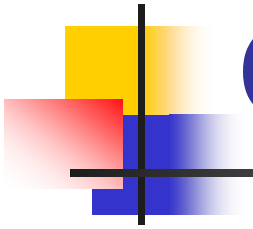


- Algorithm:
  - Match what you can at the current level
  - Odd leftovers wait for the next level
  - Repeat
- Optimal on the HST
- Cost= $\sum_i 2^i \sum_{c \in Gi} [n_P(c) \text{ is odd}]$



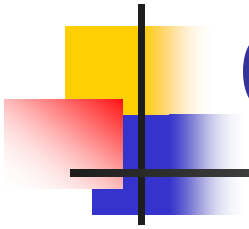
Conclusions

- Algorithms for geometric data streams
  - Insertions-only: merge and reduce
  - Insertions and deletions: randomized linear embeddings



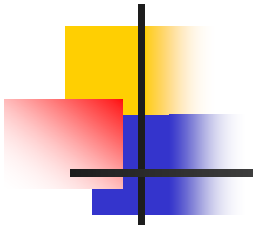
Open Problems

- High dimensions:
  - Diameter:
    - $2^{1/2}$-approx, $O(d^2 n^{1/2})$ space, follows from [Goel-Indyk-Varadarajan, SODA'01]
    - c-approx, $O(dn^{1/(c^2 - 1)})$ [Indyk, SODA'03]
    - Conjecture: $\approx 2^{1/2}$-approx, $O(d \text{ polylog } n)$ space
  - Min-width cylinder: 18-approx, $O(d)$ space [Chan'04]
  - Other problems ?



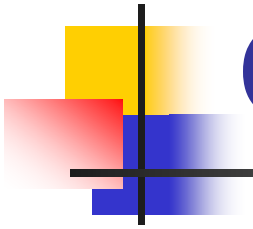
Open Problems

- Range queries:
  - General lower bounds ? (Not just for ε-approximations)
    - $\Omega(1/\varepsilon^2)$ -bit bound for general queries follows from LB for dot product [Indyk-Woodruff, FOCS'03] , and is tight (for randomized algorithms)
    - What about e.g., half-space queries ? $O(1/\varepsilon^{4/3})$ is known [STZ'04]
  - Other problems [STZ'04]



Open Problems

- Matchings, Facility Location, etc:
  - Replace $\log \Delta$ by $O(1)$ or even $1+\varepsilon$
  - Possible for MST [Frahling-Indyk-Sohler'??]
  - Related to computing bi-chromatic matching [Agarwal-Varadarajan'04]
- Min-sum clustering ?



Open Problems

- Better core-sets
  - k-median: $1/\varepsilon^d \rightarrow 1/\varepsilon^{(d-1)/2}$? Possible for $d=1$ [Indyk]
  - k-center: $1/\varepsilon^d \rightarrow 1/\varepsilon^{(d-1)/2}$ Possible for $k=1$ (this is minimum enclosing ball)
- Insertions and deletions ?
  - k-median: $\text{poly}(\log n+\log\Delta+k+1/\varepsilon)$ space/time, $(1+\varepsilon)$ –approximation ?

The End – Thank you !