Approximations and Streaming Algorithms for Geometric Problems

Piotr Indyk
MIT

Geometric Data Stream Algorithms as Data Structures

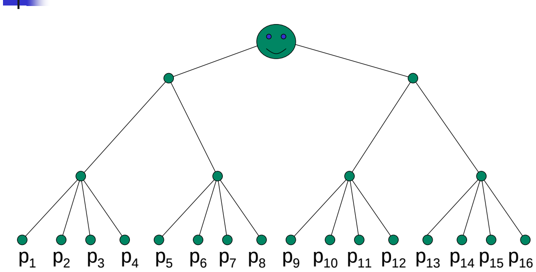
- Data structures that support:
  - Insert(p) to P
  - Possibly: Delete(p) from P
  - Compute(P)
- Use space that is sub-linear in |P|

Insertions-only

Dominant Approach: Merge and Reduce

- Main ideas:
  - Design an (off-line) algorithm that converts the input into a “sketch”:
    - Small size
    - Sufficient to solve the problem
    - A sketch of sketches is a sketch
  - Partition the input in a tree-like fashion
  - Simulate tree computation in small space
- Technique can traced back to ancient times i.e., 80’s [Munro-Paterson’78]

Tree Computation

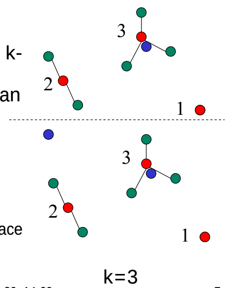


Analysis

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

Idea I: solution=sketch

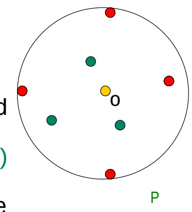
- Consider k-median
- [GMMO'00] : approximate k-median of approximate weighted k-medians is an approximate k-median
- Result:
  - Constant depth tree
  - Space: kn^{α} , $\alpha>0$
  - $O(1)$ -approximation
  - Works for any metric space



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:

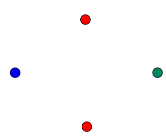
  $C_{P \cup T}(o) = (1 \pm \varepsilon)\, C_{S \cup T}(o)$
- Note: this must hold for all o's, not just the optimal one



Interlude

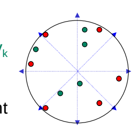
- Question from the audience: can we remove "T" from the definition of the core-set ?
- I.e., for the streaming algorithm to work, can we just require

  $C_P(o) = (1 \pm \varepsilon)\, C_S(o)$?
- Answer: NO
  - Consider $C_P(o)$ to be the diameter of P (o is a dummy argument)
  - Consider P={● ● ●}. Assume we see it in a stream first.
  - The two red points would be a core-set S for P (their diameter is equal to the diameter of P)
  - However, we could later see T={●}
  - Unfortunately, the diameter of TuP is quite different from the diameter of TuS
  - Therefore, S alone is not a good enough representation of P



Example: Core-set for MEB

- Compute extremal points:
  - Choose "densely" spaced directions $v_1 \dots v_k$
  - I.e., for any u there is v_i such that $angle(u,v_i)<\alpha$
  - For each direction maintain extremal point
- Claim:
  - The resulting set is an $O(\alpha^2)$-core-set for MEB
  - In R^d, $k=O(1/\alpha)^{d-1}$ directions suffices
    - Easy to see for $d=2$



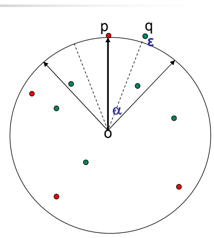
The resulting set is an $O(\alpha^2)$-core-set for MEB

- Assume radius=1
- Then we have

  $\cos(\alpha/2) \le 1/(1+\varepsilon) \approx 1-\varepsilon$

  ("=" if o-p-q angle is $\pi/2$)
- From calculus

  $\varepsilon=O(\alpha^2)$



Core-sets

- Diameter/MEB/width: $O(1/\varepsilon)^{(d-1)/2}$ space [AHP'01]
- k-center: $O(k/\varepsilon^d)$ [HP'01]
- k-median:
  - $O(k/\varepsilon^d)$ [HPM'04]
  - $O(k^2/\varepsilon^d)$ [HPK'05]
  - $O(k^2 d \log^6 n/\varepsilon)$ [Chen'05]
  - $O(d^3/\varepsilon^7)$, k=1 [Indyk'05]
- Line-clustering etc [Feldman-Fiat-Sharir'06]
- See the [Agarwal-Har-Peled-Varadarajan] survey for more

Limitations

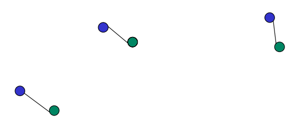
- Small core-sets might not exist
- Do not support deletions

Insertions and Deletions

Insertions and Deletions

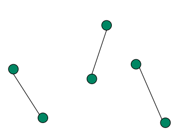
- Technique:
  - Reduction of geometric problems to vector problems
  - Use of randomized linear embeddings
- Problems:
  - Maintaining histograms of the data
  - Classic geometric problems (matching, MST, clustering etc)

Minimum Weight Bi-chromatic Matching



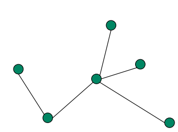
- Estimate the cost of MWBM

Minimum Weight Matching



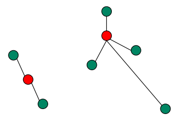
- Estimate the cost of MWM

Minimum Spanning Tree



- Estimate the cost of MST

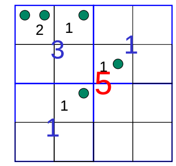
Facility Location



- 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 [Indyk'04]

- 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: $\sum_i^L 2^i \sum_{c \in G_i} [n_P(c)>0]$
  - L is the smallest level with exactly one non-zero entry in the count vector (see also later)
- 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: $\sum_i 2^i \sum_{c \in G_i - B(Q, 2^i)} n_P(c)$ (given medians Q)

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

Maintain L_1 difference [I'00]

Results

Problem	Appr.	
MST	$\log \Delta \rightarrow 1+\varepsilon$	[Frahling-Indyk-Sohler'05]
MWM	$\log \Delta$	[..., Charikar'02, ...]
MWBM	$\log \Delta$	
Fac.Loc.	$\log^2 \Delta$	
K-median	XYZ $\rightarrow 1+\varepsilon$	[Frahling-Sohler'05]

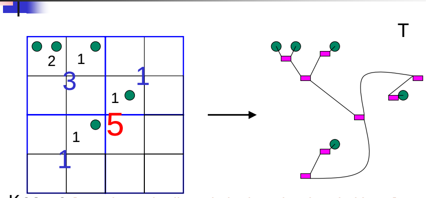
Space: $(\log \Delta + \log n + K)^{O(1)}$

XYZ

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

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

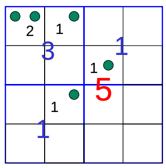
Probabilistic embeddings into HST's

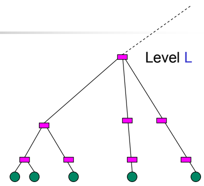


Known [Bartal'96, Charikar-Chekuri-Goel-Guha-Plotkin'98]:

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

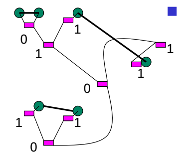
MST





- 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=$\sum_i 2^i \sum_{c \in G_i} [n_P(c) \text{ is odd}]$

Conclusions

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