Sublinear Time Algorithms for General Metrics

Piotr Indyk

Stanford University

Metric spaces

Problems: Proximity (Nearest Neighbor, Closest Pair), Clustering (MST), Regression, Routing, Matching ...

Metatheorem:

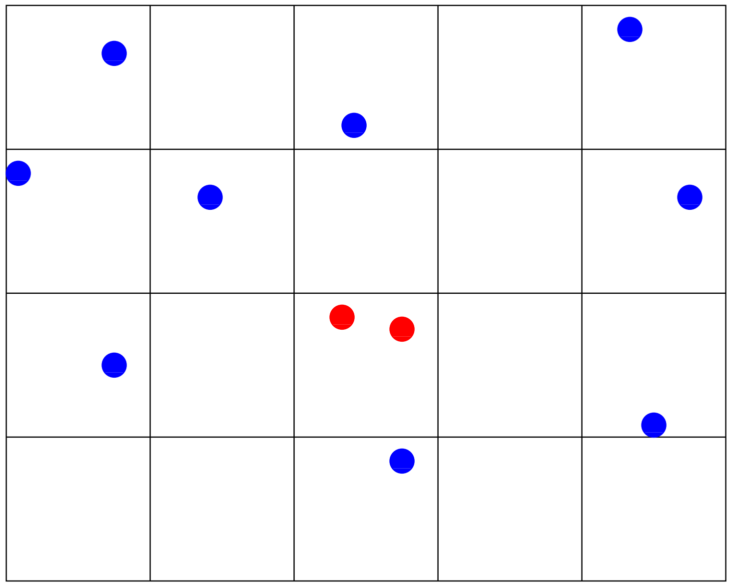
- low dimensional Euclidean space: n polylog n
- general metrics d: $\geq |d| \approx n^2$

Last lecture: $n^{2-\delta}$ for high dimensions

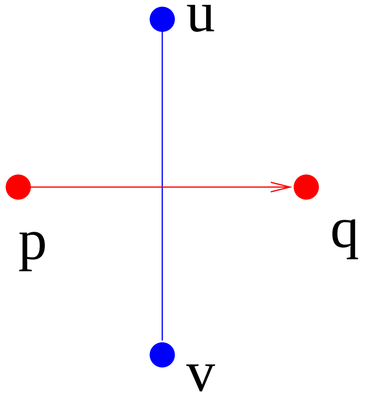
What can we do for general metrics ?

(Counter)examples

Closest pair: $\Omega(n^2)$



Furthest pair: $1/2$-approximation in $O(n)$



Results[I'99]

Problem	Our results		Previous results	
	Approx	Time	Approx	Time
k-median	$(3(2+\alpha), 2\beta)$	$\tilde{O}(kn)$	$(6,1)$ $(1+\gamma, 1+\frac{2}{\gamma})$	$\tilde{O}(n^2)$ [JV'99] $\tilde{O}(n^2)$ [CG'99]
MaxMST,MaxTSP MRCST, Alignment	$1/2$ 2	$\tilde{O}(n)$ $\tilde{O}(n)$	1, 1.32 $1+\epsilon$, 2	$\geq n^2$ $\geq n^2$
1-median* (and more) avg. distance	$1+\epsilon$ $1+\epsilon$	$\tilde{O}(n)$ $\tilde{O}(n)$	1 1	n^2 n^2

*a.k.a. Fermat-Weber Problem.

Related work

- sublinear-time approximation algorithms:
  - k-center [HS,Go] : $\tilde{O}(kn)$
  - MAX-CUT in dense graphs [GGR]:

$$2^{\text{poly}(1/\epsilon)} + n\,\text{poly}(1/\epsilon)$$

- approximate property testing/spot checking
  - their goal: solve the problem for large fraction of the input bits
  - our goal: solve the problem for the whole input

1-median Problem

Let $\Sigma(p) = \sum_q d(p,q)$.

Goal: find $\min_p \Sigma(p)$.

Idea: Approximate comparison of $\Sigma(p)$ and $\Sigma(q)$:

- if $\Sigma(p) > (1+\epsilon)\Sigma(q)$, output q
- if $\Sigma(q) > (1+\epsilon)\Sigma(p)$, output p
- otherwise, any answer is fine

Result: $O(1/\epsilon^5)$-time comparator

Implications

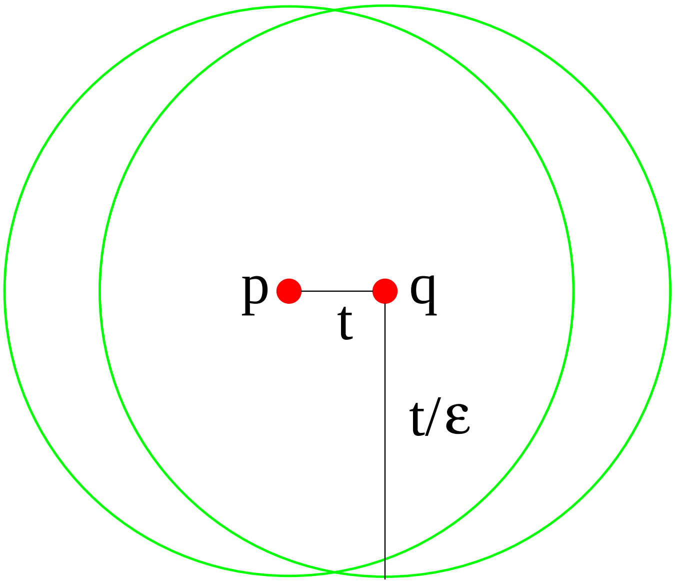
Approximate minimum computation:

Algorithm I: $O(n$ polylog $n)$-time

- set $\epsilon = O(\frac{1}{\log n})$, probability $= 1 - 1/2n$
- tournament tree of depth $\log n$
- $(1 + 1/\log n)^{\log n} \approx e$

Algorithm II: $O(n)$-time by Kleinberg's technique

Approximate comparator



Define $t = d(p, q)$, $S = B(p, t/\epsilon) \cup B(q, t/\epsilon)$

- $|S|/n$ small - any answer is fine
- $|S|/n$ large - we can sample from S:
  - estimate $\sum_{u \in S} d(p, u)$ and $\sum_{u \in S} d(q, u)$
  - output minimum

Sampling works because $d(p, s), d(q, s) \leq t/\epsilon$ and $\max(d(p, s), d(q, s)) \geq t/2$.

Other applications

- "max" instead of "min"

- approximate $\Sigma(p)$ for all p

  - approximate sorting
  - find "breakpoints" (binary search)

- improve metric MAX-CUT [VK'98] to

$$2^{\text{poly}(1/\epsilon)} + \tilde{O}(n)$$

- almost linear-time approximation scheme for 2-clustering [I4'99]

Average distance

Algorithm:

- sample $O(n/\epsilon^{2.5})$ edges
- compute the average

Problem: what if one distance (say $d(p,q)$) is large ?

Idea:

- if $d(p,q)$ is large, then $\max(d(p,s),d(q,s))$ is large, for any s
- sample will include several large edges

Formally, we bound the variance.

k-median

Algorithm:

- sample $s \approx \sqrt{kn}$ points, call it S
- find k medians $c_1 \ldots c_k$ for S
- assign each p to its closest median $c(p)$
- find kn/s points with largest $d(p, c(p))$
- solve k-median for those points
- output all medians

Time: $s^2 + (kn/s)^2 + kn = kn$