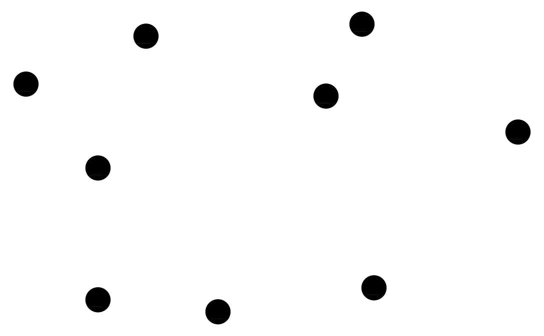
Approximate Algorithms for High-dimensional Geometric Problems

Piotr Indyk

MIT

Types of problems



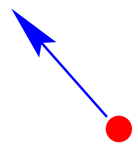
- on-line: build (or maintain) a data structure supporting queries, e.g. Near(est) Neighbor, Furthest Neighbor, ...

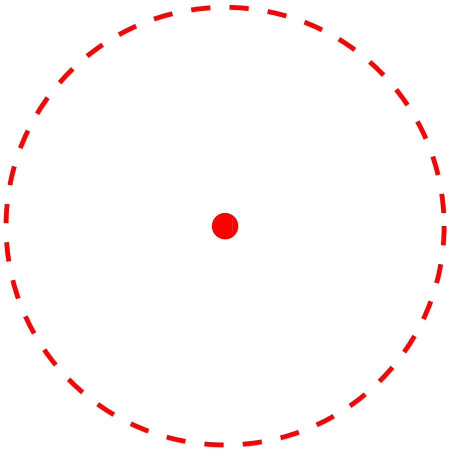
- off-line: solve a computational problem, e.g. Close(st) Pairs, Minimum Spanning Tree (MST), Diameter, ...

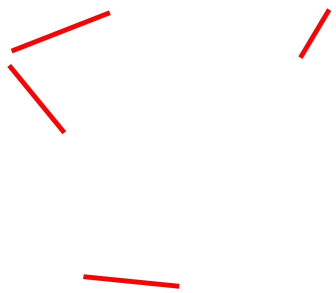
High dimensional data sets.

What are "high" dimensions ?

- Intuitively:
  - 2^d is too much
  - d smaller than n, e.g., d^2 more desirable than n^2
- Formally: $d \in [\omega(\log n), n^{o(1)}]$
- In practice: $d \geq 20$





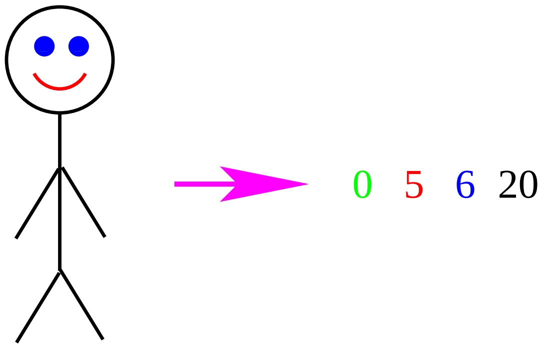


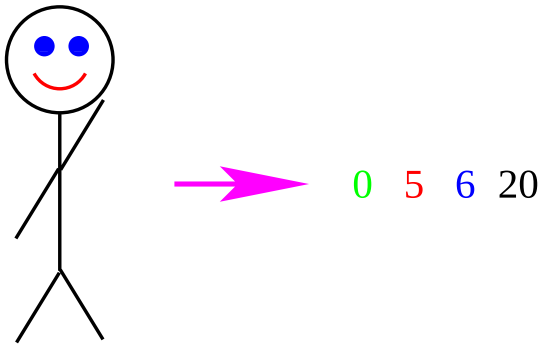
The applications

- similarity search:
  - find an object similar to a given one (Information Retrieval, Visual IR, ...), or
  - find all pairs of similar objects (e.g., Copyright Detection)
  - steps:
    - feature extraction (e.g., vectors)
    - distance measures (norms, other metrics)

- pattern recognition - "Nearest Neighbor Rule":
  - objects are labeled
  - given a new object, assign to it the label of its nearest neighbor

- compression/vector quantization
  - a set of representative objects
  - replace each object by its nearest representative


0 5 6 20


0 5 6 20

History

Computational Geometry

For $d = 2$:

- nearest neighbor: $\log n$ time, n space [Shamos'75, Lipton-Tarjan'77]
- closest pair: $n \log n$ [Shamos-Hoey'75]
- minimum spanning tree, diameter, etc

Problem: exponential dependence on the dimension, e.g.

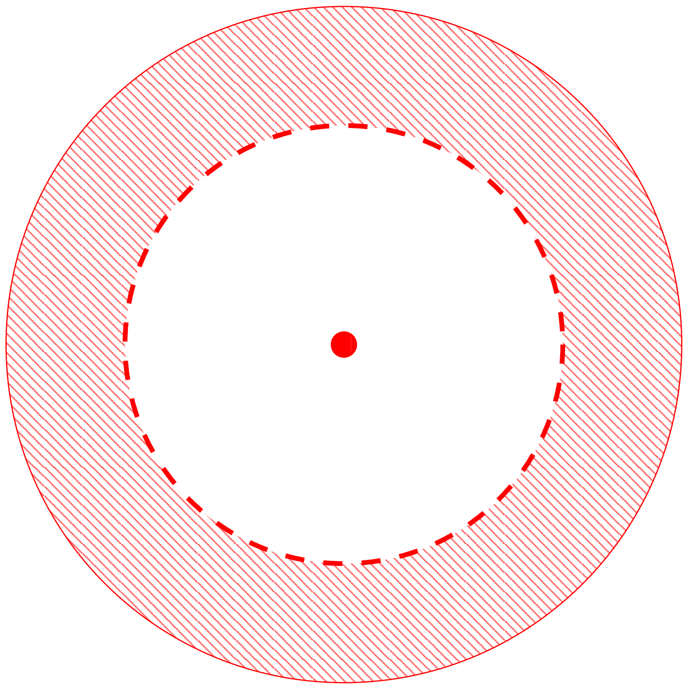
- nearest neighbor: $d^3 \log n$ time, $n^{d/2}$ space [Clarkson'88, Meiser'93]
- closest pair: $\exp(d) n \log n$ [Bentley-Shamos'76]

Databases: K-D-B-tree, R-tree, TV-tree, SS-tree ...

"Curse of dimensionality"

Conjecture: Any algorithm for the near(est) neighbor problem in high dimensions must use either $n^{\omega(1)}$ space or have $d^{\omega(1)}$ query time.

- Can be stated for any l_p norm; in fact, it is likely to hold even for $\{0,1\}^d$
- Can be made stronger by allowing:
  - Subexponential space (not just superpolynomial)
  - Sublinear in n query time
- So far, proved for query time roughly $\Theta(d)$ [Barkol-Rabani'00, Beame-Vee'02]



Approximate Computing in High Dimensions

1. Approximate near neighbor algorithms for normed spaces - focus on sublinear time

2. Algorithmic reductions to the above from nearest neighbor, closest pair, MST, diameter, matching, facility location etc - sublinear/subquadratic time

3. Extensions to not-normed spaces (shape metrics)
   - Embeddings
   - Product metrics

4. Open problems

Omitted topics

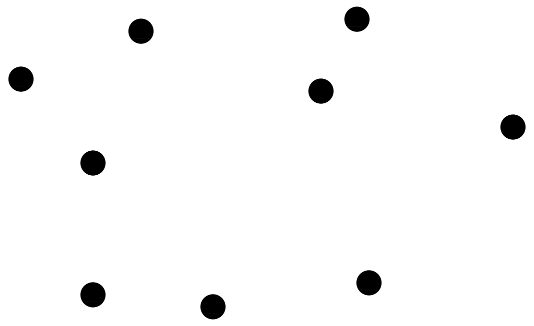
- Algorithms for general metric spaces
  [Clarkson'97, Karger-Ruhl'02]

- Recent developments in high-dimensional clustering
  - Reconstructing mixtures of Gaussians
    [Dasgupta'99, Dasgupta-Schulman'00, Arora-Kannan'01, Vempala-Wang'02]
  - $(1+\epsilon)$-approximate k-median
    [Ostrovsky-Rabani'00, Badoiu-HarPeled-Indyk'02]
  - $(1+\epsilon)$-approximate minimum enclosing ball
    [BHI'02, Badoiu-Clarkson'03, Kumar-Mitchel-Yildrim'03, Bulatov-Jambawalikar-Kumar-Sethia'03]
  - $(1+\epsilon)$-approximate k-flats
    [BHI'02, HarPeled-Varadarajan'02, BC'03, HarPeled-Varadarajan'03]

Part I: Near Neighbor

(Dynamic) Approximate Near Neighbor:

- insertions/deletions
- if there is a point within distance r from q, return some point within distance $(1+\epsilon)r$ from q
  (r fixed)



Near Neighbor: Euclidean(-like) spaces

Query	Query	Space	Update
Kle'97	$O(n)$	$O(dn)$	$O(d \log n)$
KOR'98 IM'98*	$d \log n$	$n^{O(1/\epsilon^2+1)}$	$n^{O(1/\epsilon^2)}$
IM'98**	$dn^{1/(1+\epsilon)}$	$n^{1+1/(1+\epsilon)}$	$dn^{1/(1+\epsilon)}$
I'00	$dn^{O(\frac{1+\log(1+\epsilon)}{1+\epsilon})}$	$O(nd)$	$d \log n$
Yia'00	$dn^{\gamma(\epsilon)}$ (avg case, heuristic)	$O(nd)$	

**related techniques in [Greene-Parnas-Yao'94], [Karp-Waarts-Zweig'95] and [Broder'98]

Near Neighbor: l_∞ spaces

Query	Query	Space	Update
I'98	$d \log n$	$n^{1+\delta}$ $\epsilon = O(\log_{1+\delta} \log d)$	n^δ
AH'01 H'01	$O(n \log d)$ (avg case)	$O(nd)$	

Locality-Sensitive Hashing

A family $\mathcal{H} = \{h : S \to U\}$ is called (r_1, r_2, P_1, P_2)-*sensitive* for D if for any $q, p \in S$

- if $D(p,q) \leq r_1$ then $\Pr_{\mathcal{H}}[h(q) = h(p)] \geq P_1$,
- if $D(p,q) > r_2$ then $\Pr_{\mathcal{H}}[h(q) = h(p)] \leq P_2$.

We assume $P_1 > P_2$ and $r_1 < r_2$.

Example - Hamming metric $\{0,1\}^d$:

- $\mathcal{H} = \{h(b_1 \ldots b_d) = b_i, i = 1 \ldots d\}$
  (i.e., sample one bit at random)
- $\Pr_{\mathcal{H}}[h(q) = h(p)] = 1 - D(p,q)/d$

Multi-index Hashing

To solve NN with parameters ϵ, r: set $r_1 = r$, $r_2 = (1+\epsilon)r$

Define $\mathcal{G} = \{g | g(p) = h_1(p).h_2(p) \ldots h_k(p)\}$

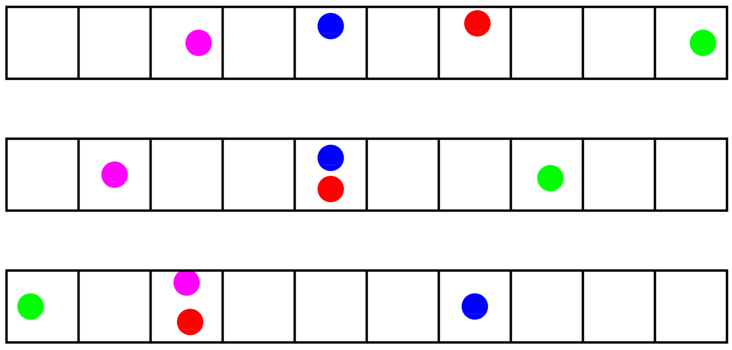
(for Hamming metric - sample k random bits)

Preprocessing: prepare indices for $g_1, \ldots, g_l$

Add p: store p in buckets $g_1(p), \ldots, g_l(p)$
Delete p: remove p from buckets $g_1(p), \ldots, g_l(p)$

Query: check $g_1(q) \ldots g_l(q)$ and report the closest among first (say) $3l$ points

Time: $O(dl)$
Storage: $O(dn + nl)$



LSH: analysis

Question: How many indices l do we need ?

Theorem: Setting $l = n^\rho$ for $\rho = \frac{\log 1/P_1}{\log 1/P_2}$ is sufficient with constant probability.

(Hamming metric $\Rightarrow \rho = 1/(1+\epsilon)$)

LSH: Proof

Define:

- p^* - a point s.t. $D(q,p^*) \le r$
- FAR(q) - all p s.t. $D(q,p) > (1+\epsilon)r$
- BUCKET$_j(q)$ - all p s.t. $g_j(p) = g_j(q)$

Events:

- E_1: $\sum_{j=1}^{l} |\text{FAR}(q) \cap \text{BUCKET}_j(q)| \le 3l$
- E_2: $g_j(p^*) = g_j(q)$ for some g_j, $1 \le j \le l$

Will show: $\Pr[\overline{E_1}] < 1/3$ and $\Pr[\overline{E_2}] < 1/e < 1/2$

Proof: Bad collisions

Let $p \in \mathsf{FAR}(q)$. Then

$$\Pr[p \in \mathsf{BUCKET}_j(q)] \leq P_2^k$$

For $k = \log_{1/P_2} n$

$$\Pr[p \in \mathsf{BUCKET}_j(q)] \leq P_2^{\log_{1/P_2} n} = 1/n$$

Thus

$$E[|\mathsf{FAR}(q) \cap \mathsf{BUCKET}_j(q)|] \leq n \cdot 1/n = 1$$

$$E[\sum_{j=1}^{l} |\mathsf{FAR}(q) \cap \mathsf{BUCKET}_j(q)|] \leq l$$

By Markov inequality

$$\Pr[\sum_{j=1}^{l} |\mathsf{FAR}(q) \cap \mathsf{BUCKET}_j(q)| > 3l] = \Pr[\overline{E_1}] \leq 1/3$$

Proof: Good collisions

For any g_j:

$$\Pr[g_j(p^*) = g_j(q)] \geq P_1^k = P_1^{\log_{1/P_2} n} = n^{-\frac{\log_{1/P_1} n}{\log_{1/P_2} n}} = n^{-\rho}$$

For $l = n^\rho$ we have

$$\begin{aligned} \Pr[\overline{E_2}] &\leq (1 - \Pr[g_j(p^*) = g_j(q)])^l \\ &\leq (1 - n^{-\rho})^{n^\rho} \\ &\leq 1/e \end{aligned}$$

Another example

Let $D(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

Fact [Broder, Glassman, Manasse, Zweig '97]: The family

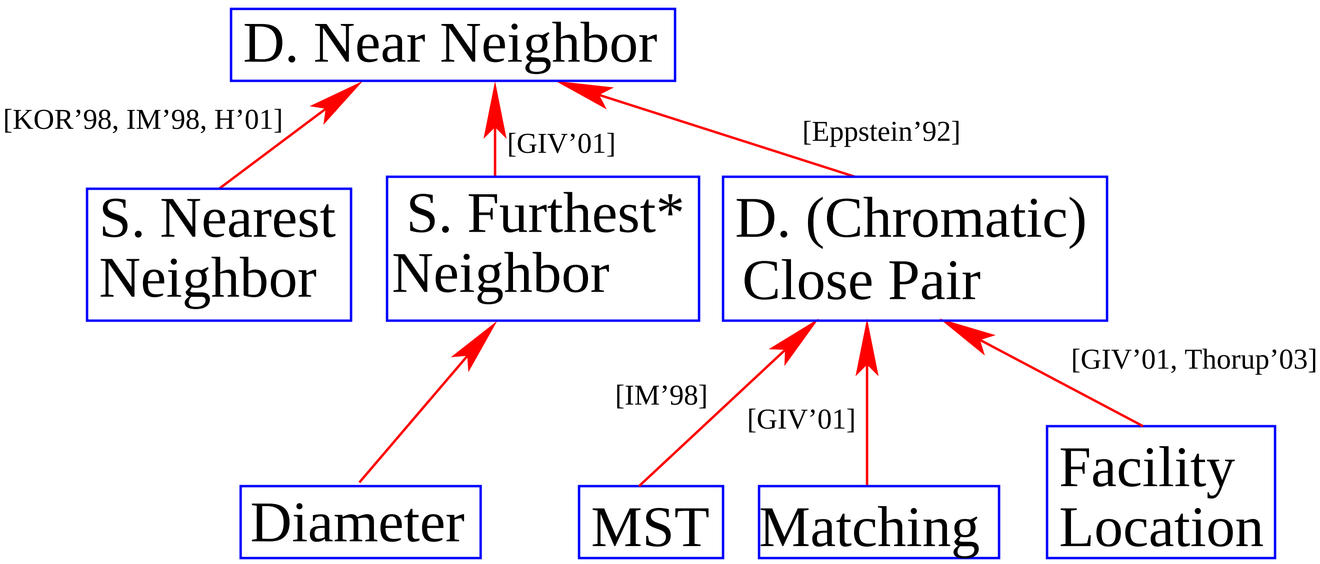
$$\mathcal{H} = \{h_\pi : h_\pi(A) = \max_{a \in A} \pi(a),\ \pi \text{ is a permutation of } S\}$$

is (r_1, r_2, r_1, r_2)-sensitive.

Reason:

$$\Pr_{h \in \mathcal{H}}[h(A) = h(B)] = D(A, B)$$

Part II: Reductions



* l_2 only; superseeded by the $O(dn^{1/(1+\epsilon)^2})$ query time algorithm of [I'03]

- All reductions involve $\log^{O(1)} n$ (or $n \log n^{O(1)} n$) overhead in space/time

Part III: Not-normed metrics

Hausdorff metric: for shapes A,B:

$$\overrightarrow{D}_H(A,B) = \max_{a \in A} \min_{b \in B} D(a,b)$$

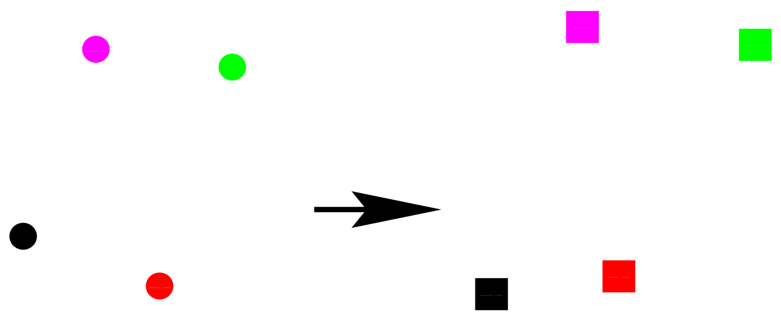
$$D_H(A,B) = \max(\overrightarrow{D}_H(A,B), \overrightarrow{D}_H(B,A))$$

☺

Approach I: Embeddings

A different type of reduction:

"Hard" metric $\Rightarrow$ "Easy" metric (i.e., a norm)



From	To	Distortion	Paper
Hausdorff metric (over pointsets)	l_∞	$1+\epsilon$	[FI'99]
Block edit distance	l_1	$\approx \log d$	[MS'00]
Earth-Mover Distance (or min bi-matching)	l_1	$\approx \log d$	[Cha'02]

Approach II: Product Metrics

- Consider edit distance $D_E(q,p)$ for strings i.e., min # of ins/del/subst transforming q into p
- Not known how to embed into normed spaces
- Can be decomposed:
  - Let $q = q_1 \dots q_k$, $|q_i| = d/k$
  - Then

$$D_E(q,p) = \min_{p=p_1\dots p_k} \sum_j D_E(q_j, p_j)$$

  - The nearest neighbor of q in $\{p^1 \dots p^n\}$ is

$$\min_{p^i} \min_{p^i=p^i_1\dots p^i_k} \sum_j D_E(q_j, p^i_j)$$

  - I.e., it is a nearest neighbor under product metric $\sum_j D_E(q_j, \cdot)$, over a pointset of size $n \exp(k)$

Product metrics: Results

- Assume we have c-approx NN algorithms for metrics $M_i = (X_i, D_i)$, $i = 1 \ldots k$, using space S

- Then we have an approx NN algorithm for metric over $X_1 \times \ldots \times X_k$, where

  $$D[(p_1 \ldots p_k), (q_1 \ldots q_k)] = f[D_1(p_1, q_1) \ldots D_k(p_k, q_k)]$$

- In particular:
  - $f = \max$: $O(c \log \log n)$-approx, $kS + n^{O(1)}$ space, fast query time [I'02]
  - f "smooth": $3c$-approx, $kS + n^k$ space, fast query time [I'0?]

- This implies algorithms for edit distance and Frechet distance (over sequences of points), with approximation factor 3^l and space $n^{d^{1/(l+1)}}$, $l \geq 1$

Open Problems: Near Neighbor

- Improve the $O(dn^{\frac{1}{1+\epsilon}})$ query time (LSH)
  - $O(dn^{\frac{1}{(1+\epsilon)^2}})$ query time for furthest neighbor
  - Furthest $\rightarrow$ nearest reduction
- Derandomization
  - $O(1)$-approximate algorithm, space $n^{O(1/\epsilon^{O(1)})}$ [I'00]
  - A $(1+\epsilon)$-approximate embedding of l_2^d into $l_1^{d'}$, $d' = d^{O(\log d)}$ [I'00]
- $O(1)$-approximate NN in l_∞ using $n^{O(1)}$ space
- Lines instead of points ?
  - As queries
  - In the dataset: $n^{O(\log n)}$ space NN algorithm known [Mag'02]

Open Problems: Not-normed Spaces

- NN under Hausdorff metric for sets of lines
  - Embedding into l_∞ ?
  - Product metrics ?

- Frechet metric
  - Over sequences of points: embedding into l_∞ ?
  - Over polygonal lines: anything ?

- Better algorithms for product metrics
  - $O(1)$-approximate NN for max-products using $n^{O(1)}$ space
  - Get rid of the n^k term in space for sum-products

Open Problems: Other

- Algorithms for very high dimensions
  - Closest pair in l_2^n: $O(MM(n))$
    (Bentley-Shamos conjectured $\approx n^2$)
  - Can obtain $n^{3-\alpha}$-time algorithm for l_∞^n

- Complexity-theoretic lower bounds