Approximate Nearest Neighbor Search

Given points $p_1 \dots p_n \in \mathbb{R}^d$

Preprocess data set to answer nearest neighbor queries:

Given query point $q \in \mathbb{R}^d$, find $\arg\min_{p_i} d(p_i, q)$

	Brute Force	BF + dim red^n
Storage	nd	$n \log(n)/\varepsilon^2$
Preprocessing	—	$n\, d \log(n)/\varepsilon^2$
Query	nd	$n \log(n)/\varepsilon^2$

Can we answer queries in sublinear time

NNS in high dim^n: many data structures have exponential dependence on dimension

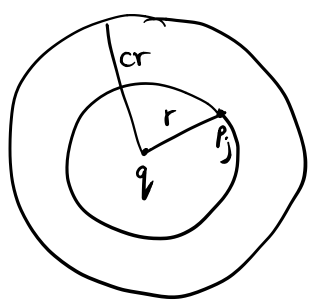
— Curse of dimensionality

JL dimension reduction not good enough

Approximate NNS

Relaxed Goal: Return p_i so that

$$d(q, p_i) \leq c \cdot \min_{p_j} d(q, p_j) \qquad c \geq 1$$



Intrinsic dimensionality of data:

Complexity of data could be low even if points lie in high dim^n

eg. low dimensional manifold.

Doubling dimension $\dim(X)$ of metric space (X, d) is min value of ρ st. every ball in X can be covered by 2^ρ balls of half the diameter.

$X = \mathbb{R}^K$ with any norm $\dim(X) = \theta(K)$

Result: [Krauthgamer, Lee '02]

Navigating Nets

Size: $2^{O(\dim(S))} \cdot n$

$(1+\varepsilon)$- approx nearest neighbor in time

$$2^{O(\dim(S))} \cdot \log \underset{\uparrow}{\Delta} + \left(\frac{1}{\varepsilon}\right)^{O(\dim(S))}$$

Aspect ratio $\Delta = \dfrac{d_{max}}{d_{min}}$

Claim: Exact nearest neighbor in time

$$2^{O(\dim(S))} \cdot \log n$$

[KOR '98] [IM '98]

$(1+\varepsilon)$ approx nearest neighbor in polylog(n) query time

poly(n) preprocessing & storage.

$\hookrightarrow n^{O(1/\varepsilon^2)}$

[IM '98] Locality Sensitive Hashing :

Defn: Family of Hash fn is (r, cr, p_1, p_2) - LSH

with $p_1 \geq p_2$, $c > 1$ if

(a) $\Pr[h(x) = h(y)] \geq p_1$ when $d(x,y) \leq r$ (close points)

(b) $\Pr[h(x) = h(y)] \leq p_2$ when $d(x,y) \geq cr$ (distant points)

Can be used to design algo for approximate NNS

Focus on following problem:

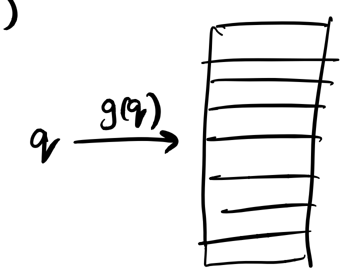
If $\exists$ point within distance r of q, return a point within cr

To solve NNS, multiple copies for different values of r

Sample K hash f^n from (r, cr, p_1, p_2)-LSH family
concatenate to get hash value

$g(x) = h_1(x)\, h_2(x) \ldots \quad h_K(x)$

$g(x) = g(y) \qquad h_i(x) = h_i(y) \;\forall i$



$d(x,y) \leq r \qquad Pr[g(x) = g(y)] \geq p_1^K$

$d(x,y) \geq cr \qquad Pr[g(x) = g(y)] \leq p_2^K$

Choose K so that $p_2^K = \frac{1}{n} \qquad K = \frac{\log n}{\log(\frac{1}{p_2})}$

$\mathbb{E}[\text{distant point collisions}] \leq 1$

$p_1 = p_2^{\rho} \qquad \rho = \frac{\log(1/p_1)}{\log(1/p_2)}$

$p_2^K = \frac{1}{n} \Rightarrow p_1^K = \frac{1}{n^{\rho}}$

$Pr[\text{good point collision}] \geq \frac{1}{n^{\rho}}$

Repeat n^{ρ} times

n^{ρ} hash tables

n^{ρ} query time $\qquad (r, cr, p_1, p_2)$-LSH

$n^{1+\rho}$ storage

ρ : best parameter over LSH schemes for all r

ρ : f^n of c and metric d

Hamming Metric:

$p_i \in \{0,1\}^d$

hash f^n : pick random coord

$d(x,y) \le r \quad Pr[h(x)=h(y)] \geq 1-\frac{r}{d} \approx e^{-r/d}$

$d(x,y) \geq cr \quad Pr[h(x)=h(y)] \le 1-\frac{cr}{d} \approx e^{-cr/d}$

$p_1 = p_2^{1/c} \qquad \rho = \frac{1}{c}$

Best possible for Hamming

l_1 norm : same result

What is the hash f^n? Pick random coord i, threshold t

$I_{\{x_i \geq t\}}$

$\rho = \frac{1}{c}$

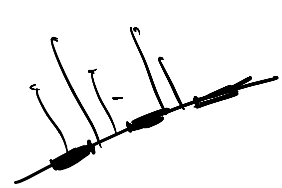
Euclidean space: l_2 norm

l_2 embeds into l_1 isometrically, so we can get $\rho = \frac{1}{c}$

Can do better

[DIIM'04] map to random line, split into buckets of width w

$h_l(x) = \left\lfloor \frac{x \cdot l + \alpha}{w} \right\rfloor \qquad \alpha \in_R [0, w)$



[AI '06] map $\pi : X \longrightarrow \mathbb{R}^d$

random seq $s_1\ s_2\ \ldots\ \in \mathbb{R}^d$, fix radius δ

$h_s(x) = \arg\min_{i>0}\ \pi(x) \in B_\delta(s_i)$

index of first ball containing $\pi(x)$

$\rho \longrightarrow \frac{1}{c^2}$ as $d \to \infty$

$\rho = \frac{1}{c^2}$ optimal

Data Dependent Hashing

[AR '15] Can achieve $\rho = \frac{1}{2c-1}$ for Hamming

$= \frac{1}{2c^2-1}$ for Euclidean

decompose data set into pseudo-random sets.

Other Hash families:

Collection of sets

$$\mathrm{sim}(A,B) = \frac{|A \cap B|}{|A \cup B|}$$ Jaccard coeff.

min-hash scheme [Broder '97]

$f : U \to 2^{64}$ (assume no collisions)

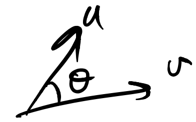
$$h(A) = \min_{a \in A} f(a)$$

$$\Pr\Big[\underbrace{\min f(A)}_{h(A)} = \underbrace{\min f(B)}_{h(B)}\Big] = \frac{|A \cap B|}{|A \cup B|}$$

"distance" fn $1 - \frac{|A \cap B|}{|A \cup B|}$

Sketches of docs for estimating similarity

Sim-Hash



unit vectors with angular distance

$$h(u) = \text{sign}(\langle r, u \rangle) \qquad r: \text{random vector}$$

$$\Pr[h(u) \neq h(v)] = \frac{\theta}{\pi} \qquad \theta = \angle(u, v)$$

used at Google

Cross-Polytope Hash [Andoni et al '15]

unit vectors.

pick random rotation & return index of largest magnitude coordinate

Achieves $\rho \approx \frac{1}{2c^2 - 1}$