Johnson - Lindenstrauss Lemma

THM: Let X be a set of N points in $\mathbb{R}^n$ and $0 < \varepsilon < 1$. Consider an $m \times n$ matrix A whose rows are independent, mean-zero, isotropic, and sub-gaussian rand. vectors in $\mathbb{R}^n$.

Rescale, A by defining (random projection) $P := \frac{1}{\sqrt{m}} . A$

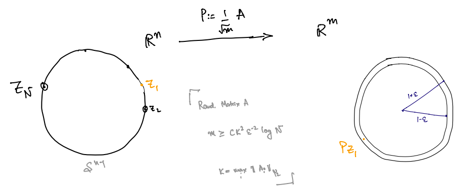
Assume that $m \geq C\varepsilon^{-2} \log N$, where C is a large enough constant (which depends on the sub g norm of A_i-s).

Then, whp, the map P preserves all pairwise distances in X, up to ε relative error.

$$(1-\varepsilon)\|x-y\| \leq \|Px - Py\|_2 \leq (1+\varepsilon)\|x-y\| \quad \text{for all } x, y \in X.$$

Johnson - Lindenstrauss (JL) Lemma

$X \subseteq \mathbb{R}^n \qquad |X| = N$



[Rand. Matrix A

$m \geq CK^2 \varepsilon^{-2} \log N$

$K = \max_i \|A_i\|_{\psi_2}$]

with high prob, for all $z \in X$

$\|Pz\| = (1 \pm \varepsilon)\|z\|$

$$\mathbb{E}\left[\max_{z \in X} \left|\|Pz\| - \|z\|\right|\right] \leq \varepsilon$$

for $m \gtrsim K^2 \varepsilon^{-2} \log N$

$$\mathbb{E}\left[\max_{z} \left|\|Az\| - \sqrt{m}.\|z\|\right|\right] \leq \varepsilon K \sqrt{\log N}$$

Applications

✓ ① Pairwise Euclidean Distances

② Streaming Alg. for F_2- estimation } This Lecture

③ Approximate Nearest Neighbor

Streaming Algorithm for F_2- estimation

[Space Constrained Environment]

Pioneering Work of. Alon, Matias, & Szegedy

* Streaming Model of Computation

- Given a stream of N indices $i_1, i_2, \ldots i_N$ $\quad i_t \in \{1, 2, \ldots, n\}$

Ex $\quad n = 3 \quad N = 6$

$[1, 1, 3, 3, 1, 2]$

- Maintain fq. vector $f_j := |\{t : i_t = j\}|$

$f = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$

[Trivial in $O(n \log n)$ space. Here, will use $O(\text{poly} \log n)$ space]

- Goal: Approximate the ℓ_2 norm $\|f\|_2$

Idea: Instead of storing f, maintain a projection of f

$y = P.f$

[$P \leftarrow$ fast JL transform]

At time t, alg receives $i_t = j$ $\longrightarrow$ @t $f \leftarrow f + e_j$

Update $y \leftarrow y + P.e_j$

(add the jth column of P to y)

Analysis

Via JL we have that, whp,

$$(1-\varepsilon)\|f\| \leq \|Pf\| \leq (1+\varepsilon)\|f\|$$

Setting $m \gtrsim \frac{1}{\varepsilon^2}$, we have this bound.

(with $m \gtrsim \frac{1}{\varepsilon^2} \log N$ this guarantee is achieved for all N steps).

Approximate Nearest Neighbor

Nearest Neighbor (Exact)

Input: Points $P = \{p_1, p_2, \ldots p_N\} \subset \mathbb{R}^n$

Objective: Preprocess P (find a data structure) such that given any query point $q \in \mathbb{R}^n$, we can quickly find

$$\arg\min_{p \in P} \|p - q\|$$

Considerations

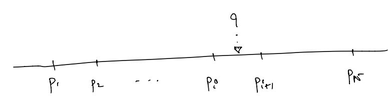
- Space (required to store data structure)
- Query Time

Exhaustive Search

- $O(nN)$ time

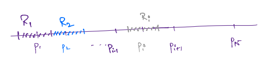
For low dimensions ($n = 1, 2$) efficient solutions are known.

$\boxed{n=1}$

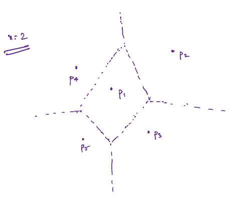


- Store given points $P = \{p_1, p_2 \ldots p_N\}$ in increasing order — Space Required $O(N)$
- Binary Search to find $p_i \leq q \leq p_{i+1}$. Output the closer of p_i, p_{i+1} — Query Time $O(\log N)$

This is the simplest instantiation of a Voronoi Diagram.



$n=2$



For $n=2$, via involved constructs, can achieve $O(N)$ space, $O(\text{poly} \log N)$ query time.

In high dimensions $\boxed{\text{no}}$ significant improvements over the naive method.
(Relevant Data Structure Kd-Trees)

Curse of Dimensionality: The failure of low-dimensional methods when applied in high dimensions

Solution by allowing randomization & approximation

ε- Approximate Nearest Neighbor
(ε- ANN)

Objective: Given a query point $q \in \mathbb{R}^n$, find $p \in P$ such that

$$\|p-q\| \leq (1+\varepsilon) \min_{p' \in P} \|p'-q\|$$

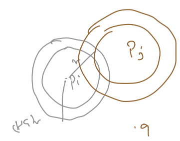
We solve ε- ANN via a reduction to a "gap" problem.

ε- PLEB
(ε- Point Location in Equal Balls)

ε- PLEB
(ε- Point Location in Equal Balls)

Input: N Euclidean balls of radius r, centered at points $P = \{p_1, p_2 \dots p_N\} \subset \mathbb{R}^n$

$B(p,r)$: Euclidean ball of radius r centered at p

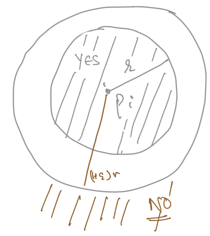


Output: Given a query $q \in \mathbb{R}^n$ we must answer as follows

- If there exists $p_i \in P$ such that $q \in B(p_i, r)$, then we must answer YES and output any point p_j with $q \in B(p_j, (1+\varepsilon)r)$

- If there does not exist p_i with $q \in B(p_i, (1+\varepsilon)r)$ we must say NO

- Otherwise (i.e., $r < \|p_i - q\| \leq (1+\varepsilon)r$) we can say either YES or NO (if we say YES, we must output a point p_j with $q \in B(p_j, (1+\varepsilon)r)$)



[YES always returned with a certificate p_j, which can be verified]

Reducing ε- ANN to ε- PLEB

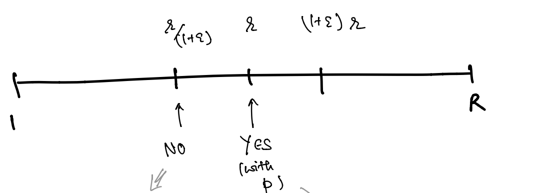
R:= max inter-point distance in P

$1 \leq \|p-p'\| \leq R$

Scaling.

- For every $r = (1+\varepsilon)^0, (1+\varepsilon)^1, (1+\varepsilon)^2, \dots, R$ construct data structure for PLEB(r)

- Given a query point $q \in \mathbb{R}^n$, via binary search, find min r s.t. PLEB(r) returns a YES for q.
  (Let $\hat{p}$ be the point returned by PLEB(r))



There does not exist a point $\hat{p}$ s.t. $\|\hat{p}-q\| \leq \frac{r}{(1+\varepsilon)}$

$\|\hat{p}-q\| \leq (1+\varepsilon)r$

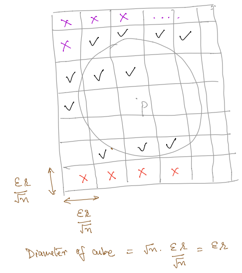
Therefore, $\hat{p}$ satisfies

$$\|\hat{p}-q\| \leq (1+\varepsilon)^2 \|p'-q\| \quad \text{for all } p' \in P$$

✓ $\hat{p}$ is a soln. for 3ε - ANN

Solving ε-PLEB

- We discretize the space & use a hash table.
  - Partition the space into n-dimensional cubes of side length $\frac{\varepsilon r}{\sqrt{n}}$ [Each cube is identified by the min point in it]



Preprocessing

- We create a hash table (initially empty)

For each cuboid C & $p_i \in P$ if $B(p_i, r) \cap C \neq \emptyset$
then insert (C, p_i) into the hash table

Query Time:

1. Given query $q \in \mathbb{R}^n$, find C that contains it
2. Look up in C in the hash table.

$\boxed{\text{No}}$ match implies that $\nexists\, B(p, r)$ which intersects with C

Therefore, $q \notin B(p, r)$ for any $p \in P$.

Hence, we can safely report $\underline{\text{NO}}$ (following PLEB requirement)

3. Say C is in the hash table with $B(p_j, r)$, (i.e. $B(p_j, r) \cap C \neq \emptyset$)

By triangle ineq.
$$\|p_j - q\| \leq r + \text{diam}(C) = r + \varepsilon r = (1+\varepsilon) r$$

Therefore, we can return p_j with $\|p_j - q\| \leq (1+\varepsilon) r$

Time and Space Analysis

Query Time $O(n)$ - Find the cube
- Hash.

Space Requirement (& Preprocessing time) Exponential in n ✗

Curse of dimensionality

Volume of a ball of radius r $= \frac{2^{cn} \cdot r^n}{n^{n/2}}$

Volume of a cuboid with side length $\frac{\varepsilon r}{\sqrt{n}}$ $= \left(\frac{\varepsilon r}{\sqrt{n}}\right)^n$

Number of cubes that intersect with the ball $\approx \left(\frac{1}{\varepsilon}\right)^n$

JL allows us to circumvent the curse of dimensionality

Apply JL to map all given points P & query q
into $m = O(\varepsilon^{-2} \log N)$ space

- Now, the space (and preprocessing time) requirement is

$$\left(\frac{1}{\varepsilon}\right)^{m} \sim N^{-\varepsilon^2 \log(1/\varepsilon)}$$

[PLEB $\longrightarrow$ ANN

Time complexity goes up by a factor of $O\left(\frac{\log R}{\varepsilon}\right)$.]

- Query time $\underbrace{O(mn)}_{JL} + \underbrace{O(m)}_{\text{Find cube \& hash.}}$

$$= O(\varepsilon^{-2} n \log N)$$