

Nearest Neighbor Search

$$X_1 = [1100 \dots 01]$$

\xleftarrow{d}

$$X_2 = [010100010]$$

\vdots

$$X_n = [101100010]$$

a query vector

$$q = [110 \dots 1]$$

\xleftarrow{d}

Objective:

find x_i

$$\text{s.t. } x_i = \underset{x_j}{\operatorname{argmin}} \operatorname{dist}(q, x_j)$$

(assume Hamming distance is considered)

Bruteforce: $O(nd)$

Question: what type of Datastructures (indices) to build, so that queries can be answered faster?

For Hamming Dist.: Range Trees.

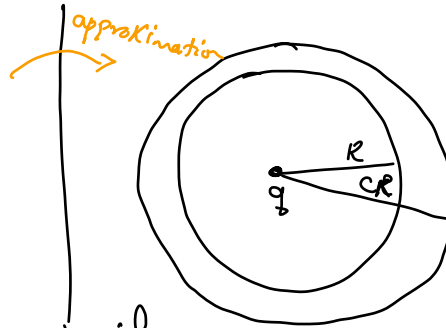
Euclidean dist.: Voronoi Diag.

$$\hookrightarrow O(d \log n)$$

challenge: these indices work for low-D
i.e., d is a small constant

How to answer NN queries in High D spaces?

$\hookrightarrow d$ is not small



if $\exists x_i$ s.t. $\operatorname{dist}(q, x_i) \leq R$

return any point x_j where

$$\operatorname{dist}(q, x_j) \leq CR$$

$$* \begin{matrix} i \in [1, n] \\ j \in [1, n] \end{matrix}$$

• A Binary hash $H: X \rightarrow \{0, 1\}$

is a function that given an input x maps it to a Binary value $H(x)$



• Given a System (X, d)

universe of possible inputs

distance function

a binary hash belongs to the class Local Sensitive hash (LSH)

if

$$\forall \operatorname{dist}(x, y) \leq R:$$

$$P(H(x) = H(y)) \geq P_1$$

$$\forall \operatorname{dist}(x, y) \geq CR$$

$$P(H(x) = H(y)) \leq P_2$$

$$\text{AND } P_1 > P_2$$

for binary vectors:

$H(X) = x_\ell$: The ℓ -th bit of vector X
 → a random index

e.g. $\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ X = [1 & 0 & 0 & 1 & 0 & 1 & 1] \\ Y = [0 & 0 & 1 & 1 & 0 & 1 & 1] \end{matrix}$

$H(X) = X_3$

$H(X) = 0$, $H(Y) = 1$

$$P(H(X) \neq H(Y)) = \frac{\text{dist}(X, Y)}{d}$$

$$\Rightarrow P(H(X) = H(Y)) = 1 - \frac{\text{dist}(X, Y)}{d}$$

$$\forall \text{dist}(X, Y) \leq R$$

$$\frac{\text{dist}(X, Y)}{d} \leq R/d$$

$$P(H(X) = H(Y)) = 1 - \frac{\text{dist}(X, Y)}{d} \geq 1 - R/d \leftarrow P_1$$

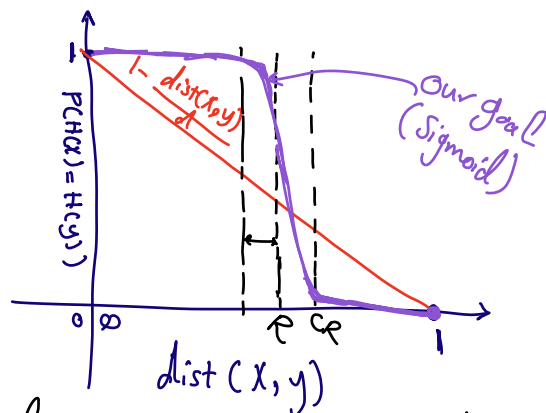
$$\forall \text{dist}(X, Y) \geq CR$$

$$P(H(X) = H(Y)) \leq 1 - \frac{CR}{d} \leftarrow P_2$$

$$P_1 - P_2 \stackrel{?}{>} 0$$

$$\begin{aligned} P_1 - P_2 &= 1 - R/d - 1 + \frac{CR}{d} \\ &= \frac{(C-1)R}{d} > 0 \quad \checkmark \end{aligned}$$

$H \in \text{LSH}$



Goal: Using binary hashes from the class of LSH, build a (non-binary) hash that satisfies:

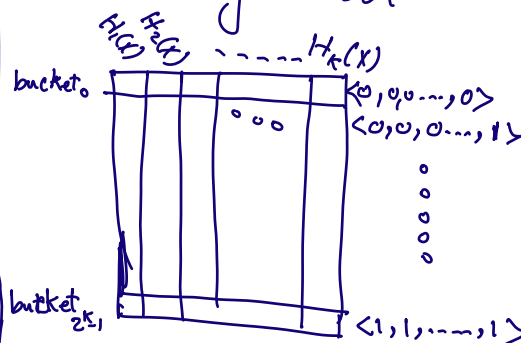
$$\forall \text{dist}(X, Y) \leq R$$

$$P(H(X) = H(Y)) \approx 1$$

$$\forall \text{dist}(X, Y) \geq CR$$

$$P(H(X) = H(Y)) \approx 0$$

Step 1: Build a bucket of binary hashes



$$H(X) = \langle H_1(X), H_2(X), \dots, H_k(X) \rangle$$

e.g. $\begin{matrix} 1 & 2 & 3 \\ X = [0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1] \\ Y = [0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0] \end{matrix}$

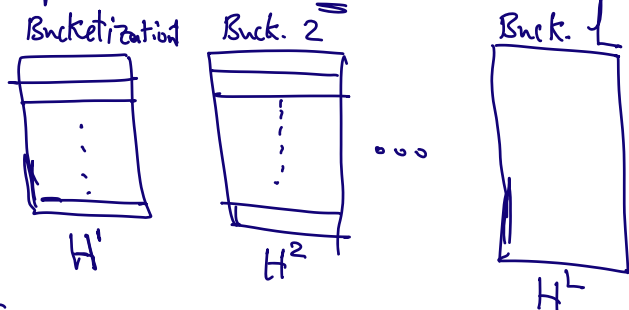
$$H_1 = X_2, H_2 = X_5$$

$$H(X) = \langle X_2, X_5 \rangle = \langle 0, 1 \rangle$$

$$H(Y) = \langle Y_2, Y_5 \rangle = \langle 1, 0 \rangle$$

$$P(X \text{ and } y \text{ fall into the Same bucket}) = P(H(y) = H(x)) \\ = \left(1 - \frac{\text{dist}(x,y)}{d}\right)^k$$

Step 2: Build L Buckets



Steps:

given the Query point q ,
return any x_j that falls into
the bucket of q in at least
one of the bucketizations

$$P(\exists B_i \text{ st. } H^i(x) = H^i(y)) =$$

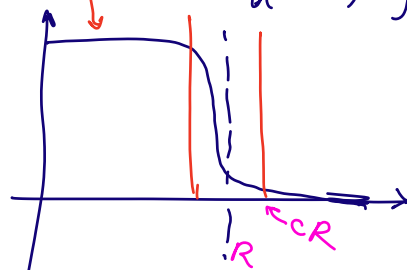
$$P((H^1(x) = H^1(y)) \vee (H^2(x) = H^2(y)) \vee \dots (H^L(x) = H^L(y)))$$

$$= 1 - P\left(\bigwedge_{i=1}^L (H^i(x) \neq H^i(y))\right)$$

$$= 1 - \left(1 - \left(1 - \frac{\text{dist}(x,y)}{d}\right)^k\right)^L$$

A Sigmoid function

$$1 - \left(1 - \left(1 - \frac{\text{dist}(x,y)}{d}\right)^k\right)^L$$



find Proper k, L