Course: Comtemporary Algorithms T.II/2019-20

Lecture 5: Approximate Nearest Neighbor (ANN)

22 January 2020

Lecturer: Dr. Kanat Tangwongsan *Scribe: Suchanun P.& Suchanuch P.*

1 Approximate Nearest Neighbor (ANN)

Exact NN is hard (and often not necessary) so an approzimation of NN is good enough.

Problem (ANN): Given a collection of D points, $ANN(q, r, c)$ is defined as:

- If $\exists x \in D$ such that $d(q, x) \leq r$, report any $y \in D$ such that $d(q, y) \leq cr$.
- If $\nexists x \in D$ such that $d(q, r) \leq r$, report fail.
- Otherwise, either report a point $\leq cr$, or report fail.

Idea: bin points – points close together are in the same bin while points far apart are in different bins.

1.1 Locality Sensitive Hashing (LSH)

Def: (LSH) For parameter $c > 1$, probability values $p_1 > p_2$ and distance $r \geq 0$, a hash family H is said to be (r, cr, p_1, p_2) - sensitive if $\forall q, x, y$:

- If $d(x, q) < r, \mathbb{P}[h(x) = h(q)] \geq p_1$
- If $d(y, q) > cr, \mathbb{P}[h(y) = h(q)] \leq p_2$

where $h \sim H$ at random.

Ex: (Hamming) For $x, y \in \{0, 1\}^k, d(x, y) \, \|x - y\|$

$$H = \left\{ h_i \mid h_i(\vec{x}) \text{ returns the i}^{th} \text{ bit of } \vec{x} \right\}$$

Easy to see: $\mathbb{P}_{h \sim H}[h(\vec{x}) = h(\vec{y})] = 1 - \frac{d(\vec{x}, \vec{y})}{k}$

- If $d(x, q) < r, \mathbb{P}[h(\vec{x}) = h(\vec{q})] \geq 1 - \frac{r}{k} = p_1$
- If $d(y, q) > cr, \mathbb{P}[h(\vec{y}) = h(\vec{q})] \geq 1 - \frac{cr}{k} = p_2$

Ideas:

- AND - drive $p_2 \to 0, p_1 \to$ somewhere reasonable
- OR - drive $p_1 \to 1, p_2 \to$ somewhere reasonable
- Parallel copies: run the algorithm many times, only require it to succeed once, e.g. if $p_1 = \frac{1}{3}$, run $2 \ln n$ times, $\mathbb{P}$ [succeed once] = $1 - p^{2 \ln n} = 1 - \frac{1}{n^2}$

Given a hash family H,

$$h_1, h_2, ..., h_k \sim H$$

where h is $c, cr, p_1, p_2)$ - sensitive and $h : \mathbb{P} \to \mathbb{U}$ (e.g. $\mathbb{R} \to \{0, 1, 2, 3, ...\}$)

1.1.1 AND

AND creates a new hash family H'

$$H' = H \times H \times ... \times H$$

Construction: $g \in H', g(x) = \langle h_1(x), h_2(x), ..., h_K(x) \rangle$ that is $g : \mathbb{P} \to \mathbb{U}^K$

$$g(x) = g(y) \iff \forall j, h_j(x) = h_j(y)$$

Thus, if $d(x, q) < r$,

$$\mathbb{P}[g(x) = g(q)] = \prod \mathbb{P}[h_j(x) = h_j(q)] \geq p_1^K$$

If $d(y, q) > cr$,

$$\mathbb{P}[g(y) = g(q)] = \prod \mathbb{P}[h_j(y) = h_j(q)] \leq p_2^K$$

1.1.2 OR

Construction: Draw $h_1(x), h_2(x), ..., h_L(x) \sim H$ For each point x, send x to binds $h_1(x), h_2(x), ..., h_L(x)$. Thus, if $d(x, q) < r$,

$$\begin{aligned} \mathbb{P}[x, q \text{hashed to the same bin }] &= 1 - \mathbb{P}[x, q \text{ hashed to none of the same bins}] \\ &\geq 1 - (1 - p_1)^L \end{aligned}$$

If $d(y, q) > rc$,

$$\begin{aligned} \mathbb{P}[y, q \text{hashed to the same bin }] &= 1 - \mathbb{P}[y, q \text{ hashed to none of the same bins}] \\ &\leq 1 - (1 - p_2)^L \end{aligned}$$

1.1.3 AND-OR

$$H \xrightarrow{\text{AND}} H' \xrightarrow{\text{OR}} h'_1, h'_2, h'_3, ..., h'_L$$

where for $h' \sim H', h' = \langle h_1, h_2, ..., h_K \rangle$

1.1.4 Choosing K and L

Goal: $\mathbb{E}$[# bad collisions] = 1

If $d(y,q) \geq cr$, K-ANDs

$$\mathbb{P}[\text{\# collisions}] \leq p_2^K = e^{K \ln p_2} = \frac{1}{n}$$

Use $K = \frac{\ln 1/n}{\ln p_2}$,

$$\mathbb{E}[\text{\# collisions}] \leq \frac{1}{n} n = 1$$

If $d(x,y) < r$,

$$\mathbb{P}[\text{x, q collided}] \geq p_1^K = e^{k \ln p_1} = \left(e^{\ln \frac{1}{n}}\right)^{\frac{\ln p_1}{\ln p_2}} = \frac{1}{n^\rho} \quad \text{where } \rho = \frac{\ln p_1}{\ln p_2}$$

Use $L = n^\rho$,

$$\mathbb{E}[\text{\# collisions}] = L$$

Querying: Have a query point q. Suppose we have $g_1, g_2, ..., g_L \sim H'$. Hash the point $q : b_1 = g_1(q), b_2 = g_2(q), ..., b_L = g_L(q)$. Look at bins $b_1, b_2, ..., b_L$.

- If found a point $\leq cr$ apart, report it and quit.
- If looked at $4L$ point already, report None and quit.
- If ran out of points in bin, move to nexti bin.

Probability of success:

Can fail if

- Look at $4L$ points already and still fail, $\mathbb{P}$[bad] $< \frac{1}{4}$
- q doesn't get put to another bin with a point close to it $\mathbb{P}$[this happends] $= \left(1 - \frac{1}{n^\rho}\right)^{n\rho} \leq \frac{1}{e}$

So $\mathbb{P}$[success] $= 1 - \frac{1}{4} - \frac{1}{e} \geq \frac{1}{3}$
To increase the chance of success, report many times: report m times $\mathbb{P}$[fail] $= \left(1 - \frac{1}{3}\right)^m$
If $m = (3\text{lg}_{\frac{2}{3}} n)^{n\rho}$ the $\mathbb{P}$[fail] $= \frac{1}{n^3}$ WHP. This reports in $O(n^\rho \text{lg} n)$ time.