COMS 4995-8: Advanced Algorithms (Spring'21) Feb 09, 2021

Lecture 9: Nearest Neighbor Search: Locality Sensitive Hashing

Instructor: *Alex Andoni* Scribes: *Haoran Pu, Rodolfo Raimundo*

1 Review: Lecture 8

Last time we discussed the problem of Nearest Neighbor Search using as a foundation for our possible solutions the Johnson-Lindenstrauss lemma from Lecture 7.
We then started discussing solutions for the c-ANN problem as described below:

Definition 1. *[c-approximation of r-near neighbor, c-ANN]: Given $c > 1, r > 0$, data set $D \subset \mathbb{R}^d, n = |D|$, build a data structure on D such that it can answer given $q \in \mathbb{R}^d$:*

- *if there exists $p^* \in D$ such that $||p^* - q|| \leq r$, then report $p \in D$ such that $||p - q|| \leq c \cdot r$*
  (in case of randomized algorithms, p is reported with probability at least 90% per point)
- *if doesn't exist $p^* \in D$ such that $||p^* - q|| \leq r$, then may or may not report anything*

After a solution adopting dimension reduction, we concluded that in some cases it is enough to use a sketching map instead of a proper dimension reducing map[1]. We then followed with a solution using sketching for ℓ_1 and the Hamming space[2].

2 Solution 1: KOR Theorem and Sketching

As described in the **Section 1**, last time we stated a theorem that could lead us to a possible solution of the c-ANN problem using sketching. Now, we will prove the theorem and better define the solution. We should first recall the Kushilevitz-Ostrovsky-Rabani Theorem:

Theorem 2. *[Kushilevitz-Ostrovsky-Rabani]: Fix $d \geq 1$ and $c = 1 + O(\varepsilon)$. Then, there exists $\theta > 0$ such that $\forall r \geq 1$ there exists a distribution φ over $\{0,1\}^d$ defined as $\varphi : \{0,1\}^d \to \{0,1\}^k$, with $k = O(\frac{\log n}{\varepsilon^2})$ such that:*

- *if $\|p - q\| \leq r$ then $\Pr[\|\varphi(p) - \varphi(q)\| \leq \theta k] \geq 1 - \frac{1}{n^3}$*
- *if $\|p - q\| > cr$ then $\Pr[\|\varphi(p) - \varphi(q)\| > \theta k] \geq 1 - \frac{1}{n^3}$*

As one should notice, the distribution φ acts like a dimension reduction mapping for a scale r.
We can now proceed with the proof of the theorem as follows:

[1] Scribe 8: Section 3
[2] Scribe 8: Section 4

Proof. First, we should define φ as transformation in the given space, thus we will have that φ takes the following form:

$$\varphi(p) = \begin{bmatrix} \cdots & u_1 & \cdots \\ \cdots & u_2 & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & u_d & \cdots \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_d \end{bmatrix}$$

The matrix M is defined such that for each u_i we have that $\Pr[u_i = 1] = \alpha = \frac{1}{2r}$ and we have that the size of the matrix M of the transformation is $d * k$ with d and k as defined previously.
We should also remember that since we defined the mapping in $\{0,1\}^d$, we have that all operations are in the Galois Field with modulo 2 ($\mathbb{F}_2$), hence, the sum operation behaves similarly to Exclusive OR ($\oplus$). Taking these properties into account, we can proceed with our proof.
For a vector p and a matrix M as defined above with $v_i = (u_1^i, u_2^i, \ldots, u_d^u)$[3], we have:

$$\varphi(p) = (v_1 \cdot p, v_1 \cdot p, \ldots, v_k \cdot p)$$

Now, for another vector q, since we have that $(+) \equiv (\oplus)$, we have:

$$\|\varphi(p) - \varphi(q)\| = \sum_{i=1}^{k} \mathbb{1}[v_i \cdot p \neq v_i \cdot q]$$

We can then take the expectation of such variables to prove our theorem.

$$\begin{aligned} \mathbb{E}[\|\varphi(p) - \varphi(q)\|] &= \mathbb{E}\left[\sum_{i=1}^{k} \mathbb{1}[v_i \cdot p \neq v_i \cdot q]\right] \\ &= k * \Pr[v_i \cdot p \neq v_i \cdot q] \end{aligned}$$

Further, we have:

$$\begin{aligned} \Pr[v_i \cdot p \neq v_i \cdot q] &= \Pr_u[u \cdot p \neq u \cdot q] \\ &= \Pr_u\left[\oplus_{i=1}^{d} u_i \cdot p_i \neq \oplus_{i=1}^{d} u_i \cdot q_i\right] \\ &= \Pr_u\left[\oplus_{i=1}^{d} u_i \cdot (p_i \oplus q_i) \neq 0\right] \end{aligned}$$

Since the operation Exclusive OR depends only on the coordinate, we have that the number of iterations of i in which $(p_i \oplus q_i) = 1$ is $\|p - q\|$. Now, since we have established that each $u_i = 1$ with probability $\frac{1}{2r}$, we can then define the distribution of u_i as follows:

- $u_i \in \{0,1\}$ with probability $\frac{1}{r} = 2\alpha$.
- $u_1 = 0$ with probability $1 - 2\alpha$

[3]For the purpose of simplification, we will be dropping the superscript on u^i, since the operations are analogous in every column of M.

Therefore, $u_1 = 1$ with probability α as defined previously, further, we have:

$$\begin{aligned}\Pr_u\left[\oplus_{i=1}^{d} u_i \cdot (p_i \oplus q_i) \neq 0\right] &= 1 - \Pr_u\left[\oplus_{i=1}^{d} u_i \cdot (p_i \oplus q_i) = 0\right] \\ &= 1 - (1-2\alpha)^{\|p-q\|} - (1-(1-2\alpha)^{\|p-q\|}) * \frac{1}{2} \\ &= \frac{1}{2}(1-(1-2\alpha)^{\|p-q\|})\end{aligned}$$

Now, we can look back at the cases described in the KOR Theorem:

1. if $\|p-1\| \leq r$, for $(1-x) \approx e^{-x}$ for small x, we will have:

$$\begin{aligned}\Pr[u \cdot p \neq u \cdot q] &\leq \frac{1}{2}\left(1 - \left(1 - \frac{2}{2r}\right)^{r}\right) \\ &= \frac{1}{2}(1 - e^{-1}) \\ &= \frac{1}{2} - \frac{1}{2e}\end{aligned}$$

2. if $\|p-1\| > cr$, for $c = 1+\varepsilon$, we will have:

$$\begin{aligned}\Pr[u \cdot p \neq u \cdot q] &\leq \frac{1}{2}\left(1 - \left(1 - \frac{2}{2r}\right)^{r*(1+\varepsilon)}\right) \\ &= \frac{1}{2}(1 - e^{-1-\varepsilon}) \\ &= \frac{1}{2}(1 - e^{-1}(1-\varepsilon)) \\ &= \frac{1}{2} - \frac{1}{2e} + \varepsilon * \frac{1}{2e}\end{aligned}$$

We can now use the Chernoff Bound on each case, hence we will have:

1. $$\|\varphi(p) - \varphi(q)\| \leq \frac{1}{2} - \frac{1}{2e} + \frac{\varepsilon}{4e}$$

2. $$\|\varphi(p) - \varphi(q)\| \leq \frac{1}{2} - \frac{2}{2e} + \frac{\varepsilon}{2e} - \frac{\varepsilon}{5e}$$

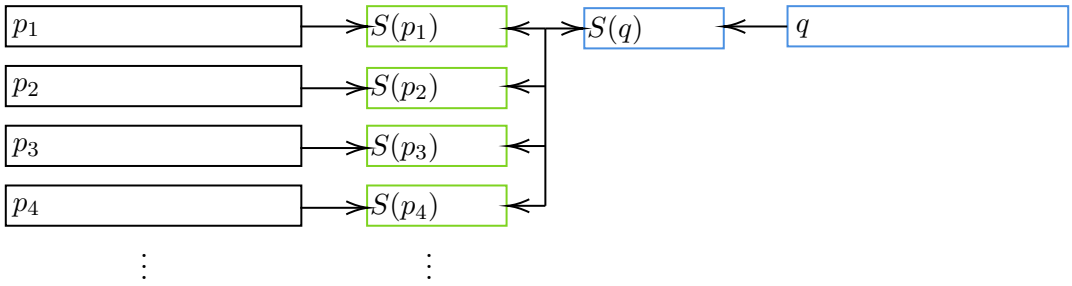
With probability $\sigma \geq 1 - \frac{1}{n^3}$. Thus, we have concluded our proof. □

As a result of our proof, we then arrive in the following corollary:

Corollary 3. *There is a solution for a $(1+\varepsilon) - ANN$ in $\{0,1\}^d$ using space $O(n*k) = O(n*\frac{\log n}{\varepsilon^2})$ and query time $O(d*k) + O(n*\frac{\log n}{\varepsilon^2})$.*

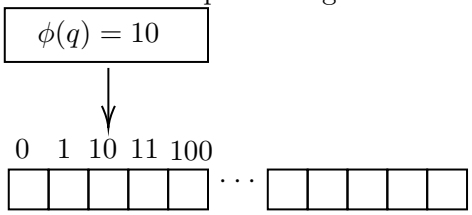
3 Solution 2: Compare Query Sketch & Dataset Sketch

The issue for **Sol 1** is that the query time contains term $O(n\ln(n))$, which is bad. And here we are trying to reduce the n term.



Assumption 4. *The sketch is $\{0,1\}^k$, which has 2^k distinct values. This can be achieved by doing the sketch in **sol 1**.*

The key trick here is the sketch itself can be a pointer of an address for the sketch. This means, if we just store all the possible answers in the address, upon computing the query sketch, we have $O(1)$ retrieval time and therefore preventing the n term. Here is an diagram showing the sketch-address trick.



And here the algorithm becomes:

Algorithm 1 Sketch Algorithm

Require: $\phi : \{0,1\}^d \to \{0,1\}^k$ created by **Sol 1**, , a $n \times \{0,1\}^d$ dataset D, a 2^k length array A and a query $q \in \{0,1\}^d$. By J-L theorem, $k \in O(\frac{\ln(n)}{\epsilon^2})$

1: **for** $x \in D$ **do**
2: `compute` $\phi(x)$ `and store it to address` σ `if it is an answer for query sketch` σ
3: **end for**
4: compute query sketch $\phi(q)$
return $A[\phi(q)]$

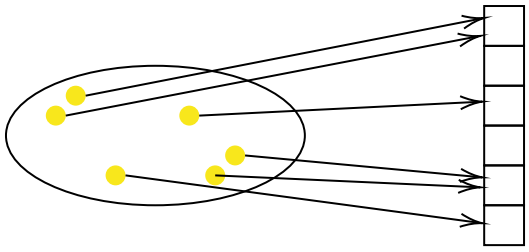
3.1 Sketch Algorithm Space & Query Time Analysis

Space complexity is based one the size of the array A: $2^k = 2^{O(\frac{\ln(n)}{\epsilon^2})} = n^{O(1/\epsilon^2)}$. This means, if one wants to have better approximate (lower ϵ), the space complexity will be higher degree polynomial ($\epsilon = 0.05$ will give $O(n^{400})$).

Query Time: Since retrieving answer is $O(1)$, most works are done for computing $\phi(q)$, which is $O(dk) = O(\frac{d\ln(n)}{\epsilon^2})$, please note that here we do eliminate the n term from **Sol 1**.

4 Solution 3: Locality Sensitivity Hashing (LSH) algorithm

The issue of **Sol 2** is that eventually the space will be a higher order polynomial of n, which makes the storage cost too large to be considered practical. The core idea is to sacrifice some query time (still under linear) while keeping space close linear as well. And this can be achieved by the application of Locality Sensitivity Hashing (LSH). Such hashing function has the property that, if two points are closed in their space, then the hash of those two points shall have high probability under the same bucket and vice versa. Here is an example show the expected behavior of locality sensitive hashing function.



Definition 5. *Fix $r, c > 0$, family $\mathcal{H}$ of $h : \mathbb{R}^d \to U$,where U is countable set, is called (r, cr, P_1, P_2)-LSH if $\forall p, q \in \mathbb{R}^d, ||p - q|| \leq r \Rightarrow P(h(p) = h(q)) \geq P_1$, and $||p - q|| \geq cr \Rightarrow P(h(p) = h(q)) \leq P_2$*

The understanding of this algorithm is quite similar with K-Mean problem where the differences are 1): U is independent of dataset D, and 2): unlike K matters in K-Mean problem, we don't care about the cardinally of U. And the algorithm becomes:

Algorithm 2 LSH Algorithm

Require: A (r, cr, P_1, P_2)-LSH hash function h, a dictionary data structure V, and query q

1: **for** $x \in D$ **do**
2: `Prepossess h(x) and store it in` V
3: **end for**
4: **for** $p \in V(h(q))$ **do**
5: `compute` $||p - q||$ `and return` p `if` $||p - q|| \leq cr$
6: **end for**

In future lecture we will show such algorithm has query time $O(n^\rho) \leq O(n^1)$ and space close to $O(n)$

4.1 $(r, cr, 1, \frac{1}{n})$ - LSH Correctness & Performance Analysis

Here lets do some best case scenario and use the best LSH we can imagine to see the upper bound of the performance. Since $||p - q|| \leq r \implies P(h(p) = h(q)) = 1$, the bucket must have the correct answer if such answer exists. And for expectation, we will compute its expectation, since the data is in dim d:

$$\mathbb{E}(\sum_{x \in V(h(q))} Query(x)) \leq d\mathbb{E}(\sum_{i=1}^{n} Q(x) I(x \in V(h(q)))) \leq d \times n \times cP_2 \in O(d)$$

Here $Query(\cdot)$ means query for entire vector, $Q(\cdot)$ means query for one coordinate, and c is computation constant between two numbers.

4.2 Impossibility of $(r, cr, 1, \frac{1}{n})$ - LSH

TL;DR: if exists, $P_1 = 1$ will enforce h has only one bucket, which contradicts the P_2 requirement. Although the performance analysis is promising, such thing is over optimistic and just mathematically impossible. Here is a gist of how such thing cannot be possible in $\mathbb{R}^1$. (You can alter interval $(a-r, a+r)$ into volume in higher dimensions to show the generic impossibility as well)

Claim 6. *LSH - $(r, cr, 1, \frac{1}{n})$ does not exist for $\mathbb{R}^1$*

Proof. We will prove it by contradiction and assume such thing exists.
Fix value a and r, we know: $\forall x \in (a-r, a+r), h(x) = h(a)$.
By using induction, this implies, for any natural number $n \in \mathbb{N}$, if $x \in (a-nr, a+nr)$, then $h(a) = h(x)$.
Recall Archimedes theorem, for any $m > 0$, there exists $n \in N$ such that $n \geq \frac{m-a}{r}$.
This means, $\forall x \in \mathbb{R}^1$, there exists a natural number $n \in \mathbb{N}$ such that
$x \in (a - nr, a + nr)$ and so $h(x) = h(a)$.
This means we have only one bucket and therefore
$\forall x$ such that if $||x - a|| \geq cr$, $P(h(x) = h(a)) = 1 > \frac{1}{n}$ for $n > 1$.
This contradicts the assumption that $||x - a|| \geq cr$, $P(h(x) = h(a)) \leq P_2 = \frac{1}{n}$.
Given the contradiction, we hereby show our assumption is false and therefore such thing does not exist.

□

And such impossibility implies P_1 must be strictly less than 1, which will be discussed in next lecture.