Locality Sensitive Hashing

Barna Saha

Outline

Approximate Near Neighbor Search

Near Neighbor Problem

- Given a set of points V, a distance metric d and a query point q, is there any point x close to query point q: $d(x, q) \leq R$.

Easy in low dimension. Complexity increases exponentially in dimension.

Approximate Near Neighbor Problem

- Given a set of points V, a distance metric d and a query point q, the (c, R)-approximate near neighbor problem requires if there exists a point x such that $d(x, q) \leq R$, then one must find a point x' such that $d(x', q) \leq cR$ with probability $> (1 - \delta)$ for a given $\delta > 0$.

The technique that we will be using to solve it is Locality Sensitive Hashing

Locality Sensitive Hashing

A family of hash functions $\mathcal{H}$ that is said to be (c, R, p_1, p_2)-sensitive for a distance metric d, when:

1. $\Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \geq p_1$ for all x and y such that $d(x, y) \leq R$
2. $\Pr_{h \sim \mathcal{H}}[h(x) = h(y)] \leq p_2$ for all x and y such that $d(x, y) > cR$

For $\mathcal{H}$ to be LSH $p_1 > p_2$.

Locality Sensitive Hashing

Example

Let $V \subseteq [0,1]^n$ and $d(x,y) =$ Hamming distance between x and y. Let $R << n$ and $cR << n$, define $\mathcal{H} = \{h_1, h_2, ..., h_n\}$ such that $h_i(x) = x_i$. $p_1 \geq 1 - \frac{R}{n}$ and $p_2 \leq 1 - \frac{cR}{n}$.

Locality Sensitive Hashing for solving (c,R)-NN problem

- LSH $\mathcal{H} : (c, R, p_1, p_2)$-sensitive
- $h_{i,j} \sim \mathcal{H}, i \in [1, K], j \in [1, L]$
- define $g_j = \langle h_{1,j}, h_{2,j}, ..., h_{K,j} \rangle$ for all $j \in [1, L]$

Locality Sensitive Hashing for solving (c,R)-NN problem

Preprocessing For all $x \in V$ and for all $j \in [L]$, add x to $bucket_j(g_j(x))$.
Time$=O(NLK)$
Query(q)

- for j=1,2,..,L
-   for all $x \in bucket_j(g_j(q))$
-     if $d(x, q) \leq cR$ then return x
- return none

Time$=O(KL + NLF)$ where F is the probability for any given j that a point x is hashed to the same bucket by g_j as q but $d(x, q) > cR$.

Locality Sensitive Hashing for solving (c,R)-NN problem

How much is F ?
Given x and y with $d(x,y) > cr$,

$$F = \Pr\left[g_j(x) = g_j(y) \mid d(x,y) > cR\right]$$

$$\prod_{j=1}^{K} \Pr\left[h_{i,j}(x) = h_{i,j}(y) \mid d(x,y) > cR\right] \leq p_2^k$$

Hence query time $O(KL + NLp_2^k)$.

Locality Sensitive Hashing for solving (c,R)-NN problem

Success Probability

$$
\begin{aligned}
&\Pr[\exists j \ \ \text{s.t.} g_j(x) = g_j(q) | d(x,q) < cR] \\
&\geq \Pr[\exists j \ \ \text{s.t.} g_j(x) = g_j(q) | d(x,q) < cR] \\
&\geq 1 - (1 - p_1^K)^L
\end{aligned}
$$

Locality Sensitive Hashing for solving (c,R)-NN problem

How to choose K and L

- set $L = \frac{1}{p_1^K}$. Success probability becomes $1 - \frac{1}{e}$. If $\delta = \frac{1}{e}$-happy!
- To minimize query cost : O(L): $Np_2^K = 1$

We have

$$N = \frac{1}{p_2^k} = \left(\frac{1}{p_1}\right)^{k\frac{\log 1/p_2}{\log 1/p_1}} = L^{\frac{\log 1/p_2}{\log 1/p_1}}$$

We have $L = N^\rho$, $\rho = \frac{\log 1/p_1}{\log 1/p_2}$

Example

$p_1 = 0.1, p_2 = 0.01$ leads to $\rho = 0.5, L = \sqrt{N}, K = O(\log N)$.
Preprocessing time$=O(N\sqrt{N}\log N)$, Query time$=O(\sqrt{N}\log N)$.