Saurabh, Shubham, Karthik, Priya
Big Data Indexing Strategies

Asymmetric Locality Sensitive Hashing (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS)

Anshumali Shrivastava, Ping Li

Table of Contents

MIPS Problem Statement and LSH

- Given a query $q \in \mathbb{R}^D$ and a collection $\mathcal{C}$ of N vectors in $\mathbb{R}^D$, search for p $\in \mathcal{C}$ such that, $p = arg \max_{x \in \mathcal{C}} q^T x$
    - Querying is frequent and N is huge.
- **Can we do better (sub-linear) than the trivial linear method?**
- MIPS is different from Near Neighbor Search $arg \min_{x \in \mathcal{C}} ||q - x||_2^2 = arg \min_{x \in \mathcal{C}} (||x||_2^2 - 2q^T x) \neq arg \max_{x \in \mathcal{C}} q^T x$
    - L_2 norm of x may vary.
- **LSH** $h : \mathbb{R}^D \Rightarrow [0, 1, 2, \ldots, N]$ with the following property: $Pr_h[h(x) = h(y)] = f(sim(x, y))$, where f is monotonically increasing.

Fast Similarity Search with LSH

- Concatenate K independent hash functions to create a meta-hash function: $B_l(x) = [h_1(x);\ h_2(x);\ \ldots;\ h_K(x)]$.
- There are L such $B_l(x)$, l = 1, 2, . . . , L.**(increases recall)**
- **Pre-processing Step:** Assign $x_i \in S$ to $B_l(x_i)$ in the hash table l, for l = 1, 2, . . . , L.
- **Querying Step:** union of all elements from buckets $B_l(q)$, union is taken over all hash tables l, for l = 1, 2, . . . , L.

Table 1

h_1^1	...	h_K^1	Buckets
00	...	00	...
00	...	01	...
00	...	10	Empty
...	...	...	...
11	...	11	...

...

Table L

h_1^L	...	h_K^L	Buckets
00	...	00	...
00	...	01	...
00	...	10	
...	...	...	...
11	...	11	Empty

L_2 LSH

- Given a fixed number r, we choose a random vector $a : a_i \sim N(0, 1)$, and a scalar b generated uniformly at random from [0, r].
- Hash function : $h^{L_2}_{a,b}(x) = \left\lfloor \frac{a^T x + b}{r} \right\rfloor$
- It can be shown that $Pr(h^{L_2}_{a,b}(x) = h^{L_2}_{a,b}(y))$ is monotonic in $||x - y||_2$.
- Unless the given data has a constant norm, $||x - y||_2 = \sqrt[2]{||x||_2^2 + ||y||_2^2 - 2x^T y}$ is not monotonic in the inner product $x^T y$.
- Simplifying the above result to a query and a data point: Pr(h(q) = h(x)) = f($||q||_2^2 + ||x||_2^2 - 2q^T x$)

LSH cannot solve MIPS

- For inner products, we can have x and y, such that $x^T y > x^T x$.
- Self similarity is not the highest similarity
- Under any hash function Pr(h(x) = h(x)) = 1. But we need Pr(h(x) = h(y)) > Pr(h(x) = h(x)) = 1

Solution : ALSH

- Use P(.) for creating buckets.
- Use Q(.) for querying probe buckets. Pr(Q(x) = P(x)) $\neq$ 1
- All we need is Pr(Q(q) = P(x)) to be monotonic in $q^T x$.
- Recall for L_2 LSH: Pr(h(q) = h(x)) = f($||q||_2^2 + ||x||_2^2 - 2q^T x$)
  $\Rightarrow$ It suffices to construct P and Q such that $||Q(q)||_2^2 + ||P(x)||_2^2 - 2Q(q)^T P(x)$ is monotonic (or approximately) in $q^T x$.

ALSH Construction

1. WLOG, let $||q||_2$ = 1. Also let $||x_i||_2 \leq U < 1,\ \forall x_i \in S$. If not divide all x_is by $\underset{x_i \in S}{max}\ \frac{||x_i||_2}{U}$

$$P : \mathbb{R}^D \mapsto \mathbb{R}^{D+m} \qquad P(x) = [x; ||x||_2^2; ||x||_2^4; ...; ||x||_2^{2^m}]$$

$$Q : \mathbb{R}^D \mapsto \mathbb{R}^{D+m} \qquad Q(x) = [x; \tfrac{1}{2}; \tfrac{1}{2}; \ldots ; \tfrac{1}{2}]$$

$$||P(x_i)||_2^2 = ||x_i||_2^2 + ||x_i||_2^4 + \cdots + ||x||_2^{2^m} + ||x||_2^{2^{m+1}}$$

$$||Q(q)||_2^2 = ||q||_2^2 + \frac{m}{4}$$

$$||Q(q) - P(x_i)||_2^2 = ||Q(q)||_2^2 + ||P(x)||_2^2 - 2Q(q)^T P(x)$$

$$= (1 + \frac{m}{4}) - 2q^T x_i + ||x_i||_2^{2^{m+1}}$$

$$||x_i||_2^{2^{m+1}} \to 0 \Rightarrow \underset{x \in \mathcal{C}}{arg\ max}\ q^T x \simeq \underset{x \in \mathcal{C}}{arg\ min} ||Q(q) - P(x)||_2$$

Removing the condition $||q||_2 = 1$

1. $P : \mathbb{R}^D \mapsto \mathbb{R}^{D+2m}$

   $P(x) = [x; ||x||_2^2; ||x||_2^4; ...;||x||_2^{2^m}; \frac{1}{2}; \frac{1}{2}; \dots ; \frac{1}{2}]$

   $Q : \mathbb{R}^D \mapsto \mathbb{R}^{D+2m}$

   $Q(x) = [x; \frac{1}{2}; \frac{1}{2}; \dots ; \frac{1}{2}; ||x||_2^2; ||x||_2^4; ...;||x||_2^{2^m}]$

   $||Q(q)-P(x_i)||_2^2 = \frac{m}{2})+||S(x)||_2^{2^{m+1}}+||S(q)||_2^{2^{m+1}}-2q^t x(\frac{U^2}{M^2})$
   $||S(x)||_2^{2^{m+1}}, ||S(q)||_2^{2^{m+1}} \leq U^{2^{m+1}} \to 0$
2. S(x) = $\frac{U}{M}x$; M = $max_{x_i \in S}||x_i||_2$

The Final Algorithm

- Preprocessing: Scale $\Rightarrow$ Append 3 Numbers $\Rightarrow$ Usual L2-LSH
  - Scale x $\in$ C to have norm $\leq$ 0.83
  - Append $\| x_i \|_2$, $\| x_i \|_4$, and $\| x_i \|_8$ to vector x_i.
  - Use standard L2 hash to create hash tables.
- Querying:
  - Append 0.5 three times to the query q.
  - Use standard L2 hash on the transformed query to probe buckets.

Evaluation metrics

- **Datasets:** Movielens and Netflix.
- Using SVD, we decompose R(Ratings matrix) into user and item latent vectors. The term R(i, j) indicates the rating of user '**i**' for an item '**j**'.
  Latent dimension f = 150 for Movielens data and = 300 for Netflix data
- How the 2 hash functions correlate with the top-T inner products? T=10
  - Compute the top-T inner prods based on the actual inner products $u_i^T v_j$, for every j, where u_i is a query and v_j is a vector point.
  - Rank all items based on $Matches_j = \Sigma\delta(h_t(u_i) = h_t(v_j))$, where δ is the indicator function. The number of times its hash value matches with the hash values of query which is u_i

Precision vs Recall graph

- Compute the precision and recall of the top-T items for any T obtained from the sorted list based on Matches. Start at the top of the ranked item list and walk down in order.
- If we are at the k-th ranked item, we check if this item belongs to the top-T similar vector points' list.
- If it is one of the top-T, then we increment the count of *relevant seen* by 1, else we move to k + 1. By k-th step, the total items seen is k.
- $\textit{Precision} = \frac{\textit{relevant seen}}{k}$
  $\textit{Recall} = \frac{\textit{relevant seen}}{T}$
- For each query choose from K $\in$ {5, 6, . . . , 30} and L $\in$ {1, 2, . . . , 200}. Use m = 3, U = 0.83 and r = 2.5. Around 10,000 experiments

c-Approximate Near Neighbor Data Structure.

- Given : a set of points in a $\mathbb{R}^D$ and parameters $S_0 > 0, \delta > 0$ and a query point q, with probability $1-\delta$, if $\exists$ S_0-near neighbor of q in P(Sim(q, p) $\geq S_0$), it outputs some cS_0-near neighbor of q in P. Popularly c-NN is solved similar to LSH
- **(Locality Sensitive Hashing (LSH))** A family H is called (S_0, cS_0, p_1, p_2)-sensitive if, $\forall$ x, y $\in \mathbb{R}^D$, h chosen uniformly from H satisfies the following : (p1 > p2 and c < 1)
  - if Sim(x, y) $\geq S_0$ then $Pr_H(h(x) = h(y)) \geq p_1$
  - if Sim(x, y) $\leq cS_0$ then $Pr_H(h(x) = h(y)) \leq p_2$
- Given a family of (S_0, cS_0, p_1, p_2)-sensitive hash functions, one can construct a data structure for c-NN with $O(n^\rho \log n)$ query time and space $O(n^{1+\rho})$, where $\rho = \frac{\log p_1}{\log p_2}$ < 1.

There can not exist any LSH family for MIPS.

Proof:

- Let $\exists$ such hash function h.
- For un-normalized inner products, Sim(x, x) = $x^T x$ = $||x||_2^2$
- There may exist another points y, such that Sim(x, y) = $y^T x > ||x||_2^2$ + M, for any constant M.
- Under any single randomized hash function h, the collision probability of the event {h(x) = h(x)} is always 1.
- So if h is an LSH for inner product then the event h(x) = h(y) should have higher probability compared to the event h(x) = h(x)
- This is not possible because the probability can not be greater than 1 .
- Note that we can always choose y with Sim(x, y) = $S_0 + \delta > S_0$ and cS_0 > Sim(x, x) $\forall S_0$ and $\forall$c < 1.

Asymmetric Locality Sensitive Hashing (ALSH)

- A family H, along with $P : \mathbb{R}^D \mapsto \mathbb{R}^{D'}$ (Processing transformation), $Q : \mathbb{R}^D \mapsto \mathbb{R}^{D'}$ (Query transformation) is called (S_0, cS_0, p_1, p_2)-sensitive if for a given c-NN instance with query q, and the hash function h chosen uniformly from H satisfies the following: (p1 > p2 and c < 1)
  - if Sim(x, y) $\geq S_0$ then $Pr_H(h(Q(q)) = h(P(x))) \geq p_1$
  - if Sim(x, y) $\leq cS_0$ then $Pr_H(h(Q(q)) = h(P(x))) \leq p_2$

  Here x is any point in the collection S.
- Given a family of hash function H and the associated query and preprocessing transformations P and Q, which is (S_0, cS_0, p_1, p_2)-sensitive, one can construct a data structure for c-NN with $O(n^\rho \log n)$ query time and space $O(n^{1+\rho})$, where $\rho = \frac{\log p_1}{\log p_2}$ (Proof in next slide)

Proof:

- Use the Fast Similarity Search with LSH with a slight modification.
- While preprocessing, we assign x_i to bucket $B_l(P(x_i))$ in table l.
- While querying with query q, we retrieve elements from bucket $B_l(Q(q))$ in the hash table l.
- By definition, the probability of retrieving an element, under this modified scheme, follows the same expression as in the original LSH.

L_2 LSH Recap

- Given a fixed number r, we choose a random vector $a : a_i \sim N(0, 1)$, and a scalar b generated uniformly at random from [0, r].
- Hash function : $h^{L_2}_{a,b}(x) = \left\lfloor \frac{a^T x + b}{r} \right\rfloor$
- It can be shown that $Pr(h^{L_2}_{a,b}(x) = h^{L_2}_{a,b}(y))$ is monotonic in $||x - y||_2$.
- Unless the given data has a constant norm, $||x - y||_2 = \sqrt[2]{||x||_2^2 + ||y||_2^2 - 2x^T y}$ is not monotonic in the inner product $x^T y$.
- Simplifying the above result to a query and a data point: Pr(h(q) = h(x)) = f($||q||_2^2 + ||x||_2^2 - 2q^T x$)

ALSH Construction Recap

1. WLOG, let $||q||_2$ = 1. Also let $||x_i||_2 \leq U < 1,\ \forall x_i \in S$. If not divide all x_is by $\underset{x_i \in S}{max}\ \frac{||x_i||_2}{U}$

$$P : \mathbb{R}^D \mapsto \mathbb{R}^{D+m} \qquad P(x) = [x; ||x||_2^2; ||x||_2^4; ...; ||x||_2^{2^m}]$$

$$Q : \mathbb{R}^D \mapsto \mathbb{R}^{D+m} \qquad Q(x) = [x; \tfrac{1}{2}; \tfrac{1}{2}; \ldots ; \tfrac{1}{2}]$$

$$||P(x_i)||_2^2 = ||x_i||_2^2 + ||x_i||_2^4 + \cdots + ||x||_2^{2^m} + ||x||_2^{2^{m+1}}$$

$$||Q(q)||_2^2 = ||q||_2^2 + \frac{m}{4}$$

$$||Q(q) - P(x_i)||_2^2 = ||Q(q)||_2^2 + ||P(x)||_2^2 - 2Q(q)^T P(x)$$

$$= (1 + \frac{m}{4}) - 2q^T x_i + ||x_i||_2^{2^{m+1}}$$

$$||x_i||_2^{2^{m+1}} \to 0 \Rightarrow \underset{x \in \mathcal{C}}{arg\ max}\ q^T x \simeq \underset{x \in \mathcal{C}}{arg\ min} ||Q(q) - P(x)||_2$$

collision probability-L_2 LSH

- $Pr(h^{L_2}_{a,b}(x) = h^{L_2}_{a,b}(y)) = F_r(d)$
- $F_r(d) = 1 - 2\Phi(-\frac{r}{d}) - (\frac{2}{\sqrt{2\pi}(\frac{r}{d})})(1 - e^{-\frac{(\frac{r}{d})^2}{2}})$
- $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$, x is the cumulative density function (cdf) of standard normal distribution and d = $||x - y||_2$ is the Euclidean distance between the vectors x and y.

Fast Algorithms for MIPS

- Given a c-approximate instance of MIPS, i.e., Sim(q, x) = $q^T x$, and a query q such that $||q||_2 = 1$ along with a collection S having $||x||_2 \leq U < 1 \; \forall x \in S$.
- We have the following two conditions for hash function $h^{L_2}_{a,b}(x)$:
  - if $q^T x \geq S_0$ then $Pr[h^{L_2}_{a,b}(Q(q)) = h^{L_2}_{a,b}(P(x))] \geq F_r(\sqrt{1 + \frac{m}{4} - 2S_0 + U^{2^{m+1}}})$
  - if $q^T x \leq cS_0$ then $Pr[h^{L_2}_{a,b}(Q(q)) = h^{L_2}_{a,b}(P(x))] \leq F_r(\sqrt{1 + \frac{m}{4} - 2cS_0})$

Fast Algorithms for MIPS - Proof

$$
\begin{aligned}
& Pr[h^{L_2}_{a,b}(Q(q)) = h^{L_2}_{a,b}(P(x))] \\
&= F_r(||Q(q) - P(x)||_2) \\
&= F_r\left(\sqrt{1 + \tfrac{m}{4} - 2q^T x + ||x||_2^{2^{m+1}}}\right) \\
&\geq F_r\left(\sqrt{1 + \tfrac{m}{4} - 2S_0 + U^{2^{m+1}}}\right)
\end{aligned}
$$

- The last step follows from the monotonically decreasing nature of F combined with inequalities $q^T x \geq S_0$ and $||x||_2 \leq U$.
- We have also used the monotonicity of the square root function.
- The second inequality similarly follows using $q^T x \leq cS_0$ and $||x||_2 \geq 0$.

Results

- $p_1 = F_r(\sqrt{1 + \frac{m}{4} - 2S_0 + U^{2^{m+1}}})$
- $p_2 = F_r(\sqrt{1 + \frac{m}{4} - 2cS_0})$
- Given a c-approximate MIPS instance, ρ is a function of 3 parameters: U, m, r. The algorithm with the best query time chooses U, m and r, which minimizes the value of ρ.
- Let $\rho^* = \min_{U,m,r} \frac{\log(F_r(\sqrt{1+\frac{m}{4}-2S_0+U^{2^{m+1}}}))}{\log(F_r(\sqrt{1+\frac{m}{4}-2cS_0}))}$ such that $\frac{U^{2^{m+1}}}{2S_0} < 1 - c$ $m \in \mathbb{N}^+$, r > 0 and U $\in$ (0, 1).

Thank you!