Probabilistic analysis of data structures for locality-sensitive hashing functions

Gao Shuxia[1, a, *], Li Chaochao[1, b],

[1]China Science and Technology University Press, No. 96 Jin Zhai Road, Hefei, Anhui Province,230026, China
a.shuxiagao_1992_0528@yahoo.com, b.chaochao_ms_0099@icloud.com
***Corresponding Author**

Abstract: For the nearest neighbor search problem in high-dimensional spaces, Locality-Sensitive Hashing (LSH)has shown excellent performance in terms of query cost and disk space utilization. Under the traditional analysis model, LSH is considered a randomized algorithm, with the only uncertainty being the choice of the hash function. In this research, the collision probability obtained under this model is referred to as the hash-function-based collision probability. In this paper, a different analysis model is used to conduct a theoretical analysis of LSH. The motivation for this work is twofold:1) Under the existing analysis model, users must generate random data structures for each query point to achieve the theoretical effect, which is impractical in real applications.2) The performance metric that users are concerned with is the expected collision probability of a random query point in a single data structure. Based on this, this paper derives the collision probability of random point pairs under the Hamming distance for any single hash function. The collision probability derived under this model is referred to as the query-based collision probability. It is also proven that in the Hamming space, the two types of collision probabilities are identical.

Keywords: Locality-Sensitive Hashing (LSH), Collision Probability, Probabilistic Analysis of Algorithms

1. Introduction

As an efficient and high-quality nearest neighbor search method in high-dimensional spaces, Locality-Sensitive Hashing (LSH)has been widely applied in many fields, including web clustering, computer vision, and bioinformatics [1-2]. The core idea of LSH is to design hash functions in different metric spaces so that the collision probability of point pairs with closer distances is greater than that of point pairs with farther distances. To date, various families of hash functions have been developed for multiple similarity metrics, such as Hamming distance and distance(s∈[0,2])、Jaccard similarity and Arccos similarity, among others.

2. Collision probability of LSH algorithms and practical application challenges

It has been found that LSH-based algorithms generally have stable error probabilities and excellent practical performance[3-6].However, all LSH algorithms rely on the following fact: given a point pair with a distance of r ,the collision probability of this point pair under a randomly selected hash function(denoted as P r $_H$(r)will decrease as r decreases P r $_H$(r)This probability is referred to as the hash-function-based collision probability. It has also been found that for approximate nearest neighbor search of a point q, the LSH algorithm can guarantee a success rate of at least P. However, according to existing literature [4,7-9], P r $_H$(r), The only source of uncertainty in the derivation is the choice of the hash function. Thus, it can be precisely stated that: given a query point q, the probability of finding the approximate nearest neighbor of q (by randomly selecting a sufficiently large number of LSH data structures, denoted as n will asymptotically approach P a n tends to infinity. In other words, to achieve the optimal theoretical performance, users would have to generate a large number of independent random LSH data structures for all query points, which is clearly not practically feasible.

In practical applications, LSH-based algorithms typically operate as follows. First, a set of hash functions is independently and randomly generated. Then, using this set of hash functions, data points are mapped to corresponding hash buckets to form the data structure. For each query point, the data structure is

https://doi.org/10.62852/csa/2025/133

accessed and returns the approximate nearest neighbor. However, in most cases, what users are concerned with is the expected collision probability of a random query point in a single data structure [3,5-6,10], which means there is a certain difference from the traditional interpretation mentioned above. In database applications, once the hash functions are randomly generated, the data structure is determined, while the distribution of data points is constantly changing for this fixed data structure [11].

From the above research, it is evident that there is an urgent need to conduct an alternative probabilistic analysis of LSH data structures. Under this analysis model, once the hash functions are randomly selected, LSH can be regarded as a deterministic data structure. In this paper, for the Hamming space, the collision probability of random point pairs on a single hash function (denoted as P r_h(r)), The research also demonstrates that this result is consistent with the one obtained under the traditional model. P r $_H$(r)identical.

3. Analysis model

As mentioned in the introduction, in P r $_H$(r) During the derivation process, LSH is regarded as a randomized algorithm. However, in practical applications, once the LSH functions are randomly selected in the preprocessing stage, they are fully determined. This indicates that there is a need to change the perspective in studying the probabilistic analysis of LSH data structures [12].

In this paper, the focus is on the derivation of the collision probability of random point pairs on a single hash function (P r_h(r)), The reason for this is P r_h(r) is that this is the fundamental starting point for the performance analysis of LSH-type algorithms. In fact, once P r_h(r)the data structure is determined and the hash functions are randomly selected, the subsequent analysis methods will be based on P r $_H$(r)the analysis methods based on [the traditional model] are already completely the same. It is obvious that, P r_h(r) it depends on the distribution of the data points. To make the analysis feasible and credible, the research assumes that the data points are randomly distributed. For a specific data distribution, P r_h(r) and P r $_H$(r) the discussion on the relationship can be found in the literature [6], and will not be elaborated here.

3.1. Preparations

To address ther-nearest neighbor search problemIndyk and MotwaniIn the literature [4], the concept of Locality-Sensitive Hashing (LSH)functions was introduced. The underlying mechanism of LSH functions is to select specific hash functions such that the collision probability of points that are close in distance is greater than that of points that are farther apart under these hash functions, thereby identifying the neighbors of a query point. Subsequently, this study uses H to denote the family of hash functions that map from R^d to some space U. For any two points o and q, when a hash function h is randomly selected from H, if the collision (h(q)=h(o)) probability satisfies the following conditions, then H can be referred to as a Locality-Sensitive Hashing (LSH)function family.

Definition: For any two points o, q∈ R^d, If the following two conditions are both satisfied, the function family H can be called (r, c r, P1, P2)-sensitive:

$$\text{If } \|q-o\| \leq r,\text{then } Pr_H[h(q)=h(o)] \geq P_1;$$

$$\text{If } \|q-o\| \geq r,\text{then } Pr_H[h(q)=h(o)] \leq P_1;$$

In order for the hash function family to be practically meaningful in design, it must satisfy P_1>P_2。In practical applications, researchers have designed different families of hash functions based on various metric spaces. This paper mainly focuses on the Hamming space.

3.2. Hamming space

In the Hamming space, data points are represented using Hamming codes, that is, binary encodings. The encodings have the same length, with each bit being either 0 or 1. The distance between data points is measured by the Hamming distance, which is defined as the number of positions at which the corresponding bits are different between two encodings. Subsequently, the notation $\{0,1\}^n$ can be used to represent the Hamming code, where n denotes the length of the encoding. For the Hamming space, Indyk and Motwania corresponding LSH function family has been constructed, namely h_i(o)=o[I]。Here I ∈[1,2,…,n] I is a

https://doi.org/10.62852/csa/2025/133

randomly selected index, and o is the encoding of the data point. It can be seen that once I is determined, the hash function h_i is also determined, that is, the value at the corresponding position of the encoding. From the definition of the function family, it is not difficult to deduce that for any two points o and q ,the collision probability (is denoted as Pr_H^b) is equal to the proportion of the number of positions where the values are the same between the two points out of the total length,a value that is ultimately determined by the Hamming distance between the two points.Assuming that the Hamming distance between o and q is r ,the research yields:

$$\mathrm{Pr}_{\mathrm{H}}^{\mathrm{b}}(\mathrm{r}) = 1 - {}^{\mathrm{r}}/_{\mathrm{n}}$$

4. Analysis and derivation

Without loss of generality, assume that the\(k\)-t h bit of the encoding is selected as the hash function. Now, for two points in the Hamming space, o_1 and o_2, Let event A denote o_1 and $o_2$collision, That is to say o_1 and $o_2$thek-th bit is the same. Let event Bdenote o_1 and $o_2$the distance between them is r .Obviously $Pr_h^b(r) = Pr(A|B)$.Also, according to the previous conclusion $Pr_h^b(r) = 1 - \mathrm{r/n}$,The following theorem will prove $Pr_h^b(r)$ and $Pr_H^b(r)$the equivalence between them.

Theorem $\mathrm{Pr(A|B)} = 1 - \mathrm{r/n}$

Proof According to Bayes' theorem, we can obtain:

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}}$$

In an n-dimensional Hamming space, the number of distinct point pairs is$2^n \times (2^n - 1)$.Assuming that the k -t h bit is the same for all point pairs, then the number of distinct point pairs is $2 \times 2^{\mathrm{n}-1} \times (2^{\mathrm{n}-1} - 1)$.Therefore, we can further obtain:

$$\mathrm{P(A)} = 2 \times \frac{2^{\mathrm{n}-1}}{2^{\mathrm{n}}} \times \frac{2^{\mathrm{n}-1} - 1}{2^{\mathrm{n}} - 1} = \frac{2^{\mathrm{n}-1} - 1}{2^{\mathrm{n}} - 1}$$

o_1 and o_2 the distance between them is r, which means o_1 and $o_2$there are r bits that are different between them, from which we can obtain P(B). The formula for P(B) is expressed as:

$$\mathrm{P(B)} = \frac{\mathrm{C_n^r} \times 2^{\mathrm{r}} \times 2^{\mathrm{n}-1}/2}{\mathrm{C}_{2^{\mathrm{n}}}^2}$$

Assumingo_1 and o_2 collide, then the probability that the k -t h bit is the same between them is:

$$\mathrm{P(B|A)} = \frac{\mathrm{C_{n-1}^r} \times 2^{\mathrm{r}} \times 2^{\mathrm{n}-1}/2}{\mathrm{C}_{2^{\mathrm{n}-1}}^2}$$

Substituting equations (2), (3), and (4) yields:

$$\mathrm{P(A|B)} = \frac{\mathrm{n} - \mathrm{r}}{\mathrm{r}} \times \frac{2^{\mathrm{n}} \times (2^{\mathrm{n}} - 1)}{2^{\mathrm{n}-1} \times (2^{\mathrm{n}-1} - 1) \times 2} \times \frac{2^{\mathrm{n}-1} - 1}{2^{\mathrm{n}} - 1} = 1 - \mathrm{r/n}$$

The theorem is proved.

5. Conclusion

In this paper, we have provided a probabilistic analysis based on fixed LSH functions. Specifically, for the commonly used Hamming space, we have derived the collision probability of random data point pairs with a distance of r under a single hash function. (P r_h (r)). During the derivation process, it was also proven that P r_h(r) and the collision probability obtained under the traditional model are completely identical. P r $_H$(r) they are completely identical.

https://doi.org/10.62852/csa/2025/133

6. References

[1] ANDONI A, INDYK P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Commun [J].ACM,2008,51(1):117-122.

[2] KE Yan, SUKTHANKARR, HUSTON L. An efficient parts-based near-duplicate and sub-image retrieval system [C]//ACM Multimedia. New York, NY, USA:ACM,2004:869-876.

[3] GAN Junhao, FENG Jianlin, FANG Qiong, et al. Locality-sensitive hashing scheme based on dynamic collision counting[C]//SIGMOD. Scottsdale, AZ, USA:ACM,2012:541-552.

[4] INDYK P, MOTWANI R. Approximate nearest neighbors: Towards removing the curse of dimensionality[C]//STOC. Dallas, Texas, USA:ACM, 1998:604-613.

[5] LV Qin, JOSEPHSON W, WANG Zhe, et al. Multiprobe ls h: Efficient indexing for high-dimensional similarity search[C]//VLDB. Vienna, Austria:ACM,2007:950-961.

[6] WANG Hong Ya, CAO Jiao, SHU LC, et al. Locality sensitive hashing revisited: Filling the gap between theory and algorithm analysis [C] //CIKM. San Francisco, CA, USA: ACM, 2013:1969-1978.

[7] BRODER AZ, CHARIKAR M, FRIEZE AM, et al. Min-wise independent permutations (extended abstract) [C]//STOC. Dallas, Texas, USA:ACM,1998,60(3):327-336.

[8] CHARIKAR M. Similarity estimation techniques from rounding algorithms [C]//STOC. Montreal, Quebec, Canadapages: ACM,2002:380-388.

[9] DATAR M, IMMORLICA N, INDYK P, et al. Locality-sensitive hashing scheme based on p-stable distributions [C]//So CG. Brooklyn, New York, USA:ACM,2004:253-262.

[10] TAO Yufei, YI Ke, SHENG Cheng, et al. Quality and efficiency in high dimensional nearest neighbor search [C]//SIGMOD. Providence, Rhode Island, USA:ACM,2009:563-576.

[11] SUNDARAM N, TURMUKHAMETOVA A, SATISH N, et al. Streaming similarity search over one billion tweets' using parallel locality-sensitive hashing[J]. PVLDB, 2013,6(14): 1930-1941.

[12] MITZENMACHER M, UPFAL E. Probability and computing- randomized algorithms and probabilistic analysis[M]. Cambridge: Cambridge University Press,2005.

[13] BRODER AZ, GLASSMANSC, MANASSE MS, et al. Syntactic clustering of the web[J]. Computer Networks,1997,29(8-13): 1157-1166.

[14] KLEINBERGJM. Two algorithms for nearest-neighbor search in high dimensions[C]//STOC. El Paso, Texas, USA:ACM,1997:599- 608.

https://doi.org/10.62852/csa/2025/133