Theoretical Analysis on Pruning Nearest Neighbor Candidates by Locality Sensitive Hashing

Tomoyuki Mutoh
Graduate School of Engineering
Osaka Prefecture University
1–1 Gakuencho, Naka, Sakai
599–8531 Japan
Email: mutoh@m.cs.osakafu-u.ac.jp

Masakazu Iwamura
Graduate School of Engineering
Osaka Prefecture University
1–1 Gakuencho, Naka, Sakai
599–8531 Japan
Email: masa@cs.osakafu-u.ac.jp

Koichi Kise
Graduate School of Engineering
Osaka Prefecture University
1–1 Gakuencho, Naka, Sakai
599–8531 Japan
Email: kise@cs.osakafu-u.ac.jp

***Abstract*—Locality Sensitive Hashing (LSH) is one of the most popular methods of the approximate near neighbor search. In applications that require the nearest neighbors of queries in a short time, LSH is sometimes used in pruning of the candidates of nearest neighbors. While the pruning reduces the processing time greatly, it also reduces the chances of retrieving the exact nearest neighbors. However, the pruning of nearest neighbor candidates using LSH has not been considered theoretically. Thus in this paper, we investigate the pruning effect by deriving the formulae of retrieval accuracy and computational cost of distance calculation for uniformly distributed data. Furthermore, we make evaluations on the formulae by comparison between simulation results and the theoretical values.**

I. Introduction

Nearest neighbor search, which finds the closest datum to the query, is one of basic techniques and widely applicable. However, due to its massive computational cost, particularly for large data, it had not been used in real applications. To alleviate the problem, the *approximate near neighbor search* has been proposed. It can greatly reduce computational cost by allowing wrong search in a certain rate.

Locality Sensitive Hashing (LSH) is one of the most popular methods of the approximate near neighbor search [1], [2], [3]. It because time complexity and space complexity are analytically considered.

Recently, several applications have been developed which require high-speed retrieval of similar samples to the query from large amount of stored data.

In the applications, LSH is sometimes used to reduce processing time [4], [5]. In the methods, the approximate near neighbors that LSH selects are regarded as the candidates of nearest neighbors and the distances between the candidates and the query are calculated to find the exact nearest neighbors. While the *pruning* reduces the processing time greatly, it also reduces the chances of retrieving the exact nearest neighbors. Although the performance of LSH is analytically considered, the performance of the pruning of nearest neighbor candidates using LSH has not considered theoretically. Thus in this paper, we investigate the pruning effect by deriving the formulae of retrieval accuracy and computational cost of distance calculation for uniformly distributed data. In order to derive the formulae, we consider the distance between the nearest neighbor and the query. Supposing the distance is known, we construct a hypersphere model and derive the formulae.

II. Locality Sensitive Hashing

Locality Sensitive Hashing (LSH) [1], [2], [3] retrieves close points with high probability and far points with low probability thanks to *locality sensitive* hash functions. In vector space [2], a locality sensitive hash function is defined by

$$h(\boldsymbol{p}) = \left\lfloor \frac{\boldsymbol{a} \cdot \boldsymbol{p} + b}{w} \right\rfloor, \tag{1}$$

where $\boldsymbol{a}$ is a d-dimensional random vector whose elements follow the standard Gaussian distribution independently, b is a real numbers chosen uniformly from the range $[0, w]$, and w indicates the width of bins. The regions defined by a locality sensitive hash function is illustrated in Fig. 1(a). The points existing in the region are regarded as approximate near neighbors.

In high-dimensional space, however, too many points are retrieved as the approximete near neighbors. Thus, as shown in Fig. 1(b), LSH constructs a bucket by combining multiple (say k) hash functions. The region of the bucket is defined by the intersection of the regions of hash functions.

In order to increase the probability of retrieving the exact near neighbors, LSH also uses multiple (say L) buckets as illustrated in Fig. 1(c). The region is defined by the union of the regions of buckets.

Finally, for subsequent discussions, we should care about widths of bins. As in Fig. 2, the widths of bins in the feature space change due to the effect of projection. We call the width in the feature space *apparent width*, which is denoted by $w' = \frac{w}{\|\boldsymbol{a}\|}$.

III. Formulation of pruning effect

In this section, we derive theoretical formulae of the pruning effect in retrieval accuracy and computational cost of distance calculation. Both of them increase as the volume of the buckets (e.g., Fig 1(c)) increase. However, the tendencies of increase are not identical due to following reasons. The retrieval accuracy is determined by whether the exact nearest neighbor is likely to exist in the buckets. Since the exact

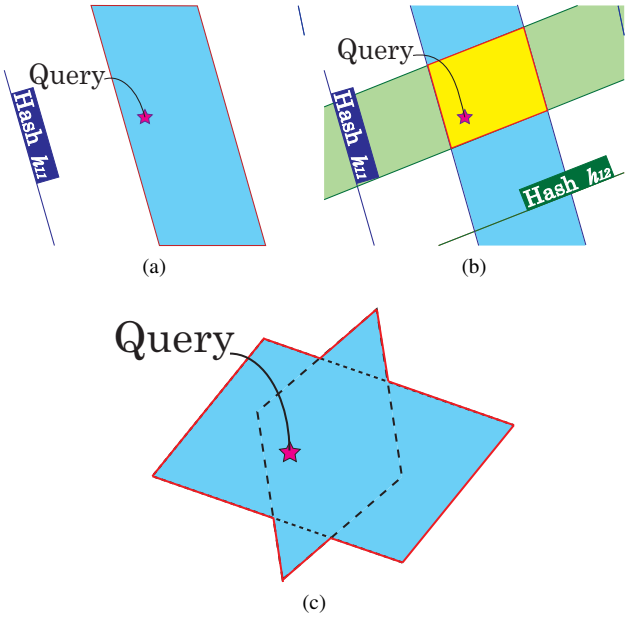


Fig. 1: The regions that approximate near neighbors exist, defined by locality sensitive hash functions. (a) The region of one hash function. (b) The region of a bucket (yellow region) consisting of two hash functions. (c) The region of two buckets.

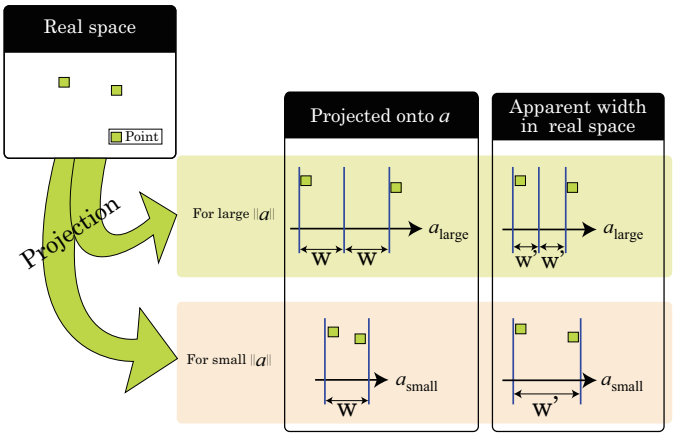


Fig. 2: The *apparent widths* in the feature space change according to the norm of a vector $\|\boldsymbol{a}\|$. While the width in the projected space (left column) is the same, the corresponding width in the original feature space (right column) are different.

nearest neighbor is likely to exist in a relatively close region, the retrieval accuracy depends on the locations of the buckets. To the contrary, computational cost of distance calculation is determined by the number of candidates for distance calculation. Thus the cost is independent of the locations of the buckets, as long as the distribution of the points is assumed to be uniform.

Although the retrieval accuracy and computational cost of

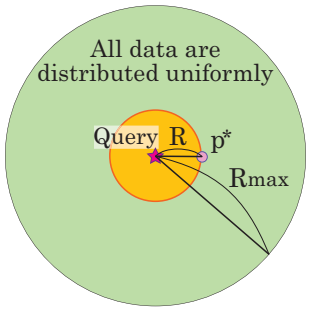


Fig. 3: The model used for derivation. All data are uniformly distributed in a very large query-centered hypersphere with radius R_{max}. A point p^* is placed on the surface of a hypersphere with radius R.

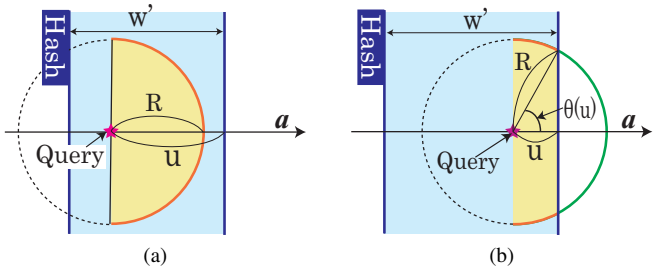


Fig. 4: The relationships between the half-hypersphere and the query's bin. (a) Whole the half-hypersphere is contained in the bin. (b) Part of the half-hypersphere is contained in the bin. As presented in Step 1, $\mathrm{P}_h(u, w', R)$ is given by (the length of orange curves) / (the length of orange and green curves).

distance calculation have a slight difference in their tendencies, the theoretical formulae of them can be similarly derived. For the sake of simplicity, we create a simple model; as shown in Fig. 3, all data are uniformly distributed in a very large query-centered hypersphere with radius R_{max}, and a point p^* is placed on the surface of a hypersphere with radius R. Then, the probability that p^* exists in the query's bin is calculated. Since the model is mirror symmetric, we consider only the right half-hypersphere. In the following steps, we derive the two formulae using the model

[Step 1] In this step, we obtain $\mathrm{P}_h(u, w', R)$, which is the probability that p^* falls into the query's bin in a single hash function. The probability is calculated as the ratio of the following two volumes: (i) the surface area of the half-hypersphere (the length of orange and green curves in Fig. 4), and (ii) the overlapping surface area between the hypersphere and the query's bin (the length of orange curves in Fig. 4). The surface area (i) is easily obtained as

$$\mathrm{s}^d(R) = \frac{2\pi^{d/2}}{\Gamma(d/2)} R^{d-1} \tag{2}$$

is the surface area of a d-dimensional hypersphere with radius R. The surface area (ii) is a bit complicated because, as shown in Fig. 4, the surface area changes due to the relationship

among R, w', and the relative position u of the query in the bin. We present a way of calculation of the surface area (ii), which is given by

$$\mathrm{s}_0^d(R) = \int_{\theta(u)}^{\frac{\pi}{2}} \mathrm{s}^{d-1}(R\sin\phi)R \; d\phi, \tag{3}$$

where

$$\theta(u) = \begin{cases} 0, & \text{for (A)} \\ \cos^{-1}\frac{u}{R}, & \text{for (B)} \; . \end{cases} \tag{4}$$

Finally, we obtain the ratio of two surface areas as

$$\mathrm{P}_h(u, w', R) = \frac{\mathrm{s}_0^d(R)}{\mathrm{s}^d(R)/2}. \tag{5}$$

In the following steps, we take care of changes of the parameters u, w' and R to the probability of Eq. (5), respectively.

[Step 2]

In this step, we take care of the change of the relative position u of the query. Let $\mathrm{P}_h(w', R)$ be the expectation of Eq. (5) for u. Since b of Eq. (1) follows the continuous uniform distribution $U(0, w)$ and the positions of bins are decided according to b, the distribution of u is also $U(0, w)$. Therefore, $\mathrm{P}_h(w', R)$ is given by

$$\mathrm{P}_u(w', R) = \frac{1}{w'} \int_0^{w'} \mathrm{P}_h(u, w', R) \; du. \tag{6}$$

[Step 3]

In this step, we take care of change of the apparent width

$$w' = \frac{w}{\|\boldsymbol{a}\|}. \tag{7}$$

Let $A = \|\boldsymbol{a}\|$. Since each element of the vector $\boldsymbol{a}$ follows Gaussian distribution, A follows χ^2-distribution. Thus the expectation of

$$\mathrm{P}_u(w', R) = \mathrm{P}_h(\frac{w}{A}, R) \tag{8}$$

for A is obtained as

$$\mathrm{P}_h(w, R) = \int_0^{\infty} p_{\chi^2}(A)\mathrm{P}_h\left(\frac{w}{A}, R\right) \; dA, \tag{9}$$

where $p_{\chi^2}(A)$ is the probability density function of χ^2-distribution.

[Step 4]

In this step, we take care of change of the radius R. As in previous steps, the expectation of Eq. (9) for R is given by

$$\mathrm{P}_h(w) = \int_0^{R_{\max}} p(R)\mathrm{P}_h(w, R) \; dR, \tag{10}$$

where $p(R)$ is a probability density function of R. As mentioned above, since $p(R)$ for the retrieval accuracy and computational cost are different, we use different functions as $p(R)$. Therefore, Eq. (10) is denoted by $\mathrm{P}_h^{\mathrm{a}}(w)$ and $\mathrm{P}_h^{\mathrm{b}}(w)$ for the retrieval accuracy and computational cost, respectively.

For the computational cost,

$$p(R) = \frac{\mathrm{s}^d(R)}{\mathrm{V}^d(R_{\max})} \tag{11}$$

is used, where

$$\mathrm{V}^d(R_{\max}) = \frac{\pi^{d/2}R_{\max}^d}{\Gamma[(d/2)+1]} \tag{12}$$

is the volume of the d-dimensional hypersphere with radius $R_{\max}$. For the retrieval accuracy, the probability density of the exact nearest neighbor, which is directly derived from Eq. (2.1.6) in p.10 of [6], given by

$$p(R) = N\left[1 - \frac{\mathrm{V}^d(R)}{\mathrm{V}^d(R_{\max})}\right]^{N-1} \frac{\mathrm{s}^d(R)}{\mathrm{V}^d(R_{\max})} \tag{13}$$

is used, where N is the number of the data exist in the very large hypersphere with radius $R_{\max}$.

[Step 5]

The probabilities calculated in the previous steps are with respect to a single hash function. Since LSH constructs buckets using multiple hash functions, the probabilities with respect to buckets are calculated in this step. Let $\mathrm{P}^{\mathrm{a}}(w)$ be the probability that the query and the exact nearest neighbor exist in at least one bucket. Similarly, let $\mathrm{P}^{\mathrm{b}}(w)$ be the probability that the query and any point exist in at least one bucket. Finally, as written in [1], $\mathrm{P}^{\mathrm{a}}(w)$ and $\mathrm{P}^{\mathrm{b}}(w)$ are obtained by

$$\mathrm{P}^{\mathrm{a}}(w) = 1 - [1 - \{\mathrm{P}_h^{\mathrm{a}}(w)\}^k]^L \tag{14}$$

$$\mathrm{P}^{\mathrm{b}}(w) = 1 - [1 - \{\mathrm{P}_h^{\mathrm{b}}(w)\}^k]^L, \tag{15}$$

respectively.

[Step 6]

This step is only for computational cost. As the indicator of it, the number of candidates for distance calculation is used. It is given by $N\mathrm{P}^{\mathrm{b}}(w)$.

IV. Evaluations

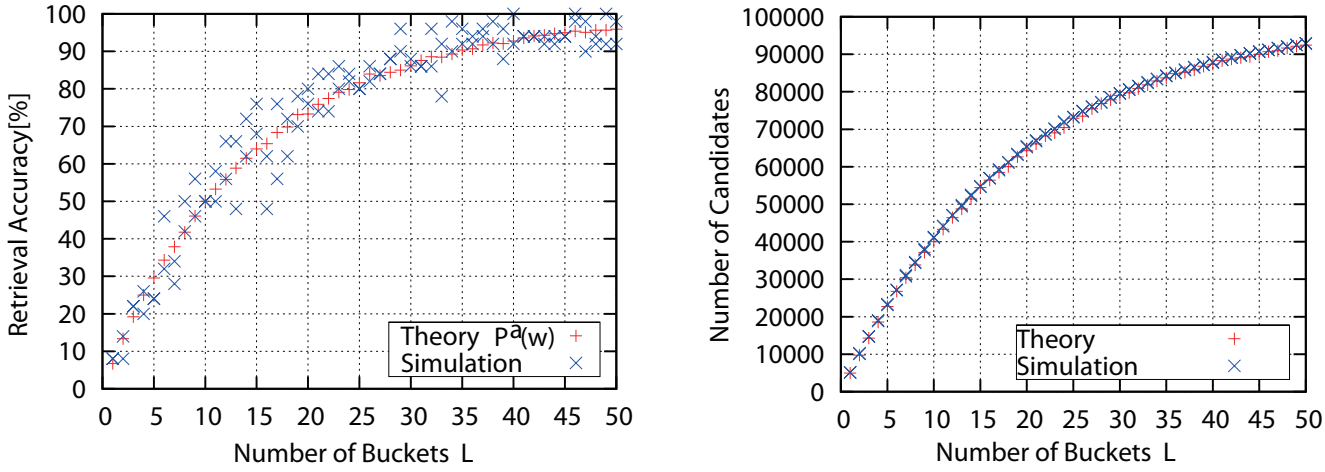
In this section, we present two results; one is comparison between simulation results and the theoretical values in order to evaluate correctness of the theoretical values derived in Sect. III, and the other is the pruning effect defined by retrieval accuracy per the numbers of candidates. As the theoretical values, $\mathrm{P}^{\mathrm{a}}(w)$ and $N\mathrm{P}^{\mathrm{b}}(w)$ were calculated by Monte Carlo method because the equations contain integrations which cannot be obtained analytically.

Firstly, we performed comparison between theoretical values and simulation results. In the simulations, we prepared the same situation as the derivation of the formulae. That is, artificial data distributed uniformly in the very large hypersphere with radius $R_{\max}$ were used. As simulation results, averaged values of 8,000 trials (combinations of 160 data and 50 sets of hash functions) were used. As the parameters, the dimensionality $d = 100$, the number of data $N = 100,000$, the radius of the very large hypersphere $R_{\max} = 5,000$, and the width of bins $w = 5,000$ were used. The graphs in Figs. 5(a) and 5(b) show that

(a) the relationship between the number of buckets L and the retrieval accuracy,

and

(b) the relationship between L and the number of candidates, respectively.



(a) (b)

Fig. 5: (a) retrieval accuracy and (b) the number of candidates in change of the number of buckets L in the range $[1, 50]$. $d = 100$, $k = 3$, $w = 5000$ were used.

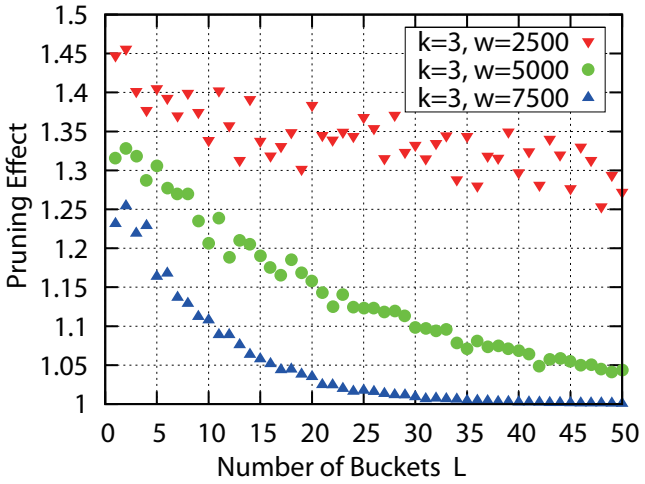


Fig. 6: The relationship between L and the pruning effect $\mathrm{P}^{\mathrm{a}}(w)/\mathrm{P}^{\mathrm{b}}(w)$.

In both figures, the theoretical values were consistent with the simulation results. We obtained similar results to the graphs, on the examinations using various combinations of parameters k, L and w.

Secondly, we evaluated the pruning effect. Since the theoretical values were consistent with the simulation results, we used $\mathrm{P}^{\mathrm{a}}(w)/\mathrm{P}^{\mathrm{b}}(w)$ as the indicator of the pruning effect. Fig. 6 shows the relationship between L and the pruning effect. In the figure, all results were not below 1. Since the expected value of the pruning effect for random selection of data is 1, LSH achieved better results. In addition to that, the pruning effect equaled 1 when $\mathrm{P}^{b}(w) = 1$, because all of the data were selected as the candidates of the nearest neighbor point. While small L and w tend to increase the pruning effect, they tend to decrease the retrieval accuracy. Thus, achieving high retrieval accuracy and high pruning effect simultaneously is difficult. In addition, an attempt to increase the pruning effect by using too small w may be failed because too small w likely to construct too narrow bins containing no data.

V. Conclusion

In this paper, in order to evaluate the effectiveness of pruning nearest neighbor candidates by LSH, we analytically derived theoretical formulae of retrieval accuracy and computational cost of distance calculation. According to the evaluations, we found that our theoretical values were consistent with the simulation results, and confirmed that pruning by LSH is suitable to applications which allow certain amount of error for reducing processing time, but has some limitations. Future work includes applying our results to real data.

Acknowledgment

This research was supported by SCAT Research Grant and KAKENHI 19300062.

References

[1] P. Indyk and R. Motowani, “Approximate nearest neighbors: Towards removing the curse of dimensionality,” In Proceedings of the 30th ACM Symposium on Theory of Computing(STOC’98), pp.604–613, May 1999.

[2] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable,” InProceedings of the 20th Annual Symposium on Computational Geometry(SCG2004), pp.253–262, June 2004.

[3] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06), pp.459–468, Oct. 2006.

[4] A. Frome and J. Malik, “Object recognition using locality sensitive hashing of shape contexts,” chapter 10, pp.221–247, MIT Press, 2006.

[5] B. Matei, Y. Shan, H.S. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert, “Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation,” IEEE Trans. PAMI, vol.28, pp.1111–1126, 2006.

[6] H.A. David and H.N. Nagaraja, Order statistics, Wiley-Imterscience, Aug. 2003.