Pivot Selection for Dimension Reduction Using Annealing by Increasing Resampling*

Yasunobu Imamura[1], Naoya Higuchi[1], Tetsuji Kuboyama[2],
Kouichi Hirata[1] and Takeshi Shinohara[1]

[1] Kyushu Institute of Technology, Kawazu 680–4, Iizuka 820–8502, Japan
[2] Gakushuin University, Mejiro 1–5–1, Toshima, Tokyo 171–8588, Japan
imamura.kit@gmail.com

Abstract. In order to select an optimal set of pivots for dimension reduction, such as Simple-Map and sketches based on ball partitioning, we propose a method named Annealing by Increasing Resampling (AIR, for short). AIR assumes that every state is evaluated by using a sample set. Starting from an arbitrary initial state, AIR repeats to transit states by hill climbing, with evaluating the resampled sets whose size initially is small and gradually increases. Experiments verify that AIR can find better sets of pivots than the conventional method and in shorter time than simulated annealing.

Keywords: Similarity Search, Dimension Reduction, Pivot Selection, Simulated Annealing, Annealing by Increasing Resampling.

1 Introduction

Similarity search is one of the most important tasks for information retrieval of multi-dimensional data. In this paper, we deal with similarity search in metric spaces, where objects within smaller distance are considered similar. Thus, similarity search is a task to find objects near to a given query object.

When the dimensionality of objects is m, the computational cost to measure distance between two objects is $O(m)$, and when the number of database objects is n, a naïve similarity search by sequential manner needs $O(mn)$ cost, which is unrealistic for larger m and n. In order to weaken the effect of n, hierarchical index structures such as R-Tree [1] and M-Tree [2, 3] have been developed. On the other hand, the dimension reduction is a method to avoid influence of m.

Dimension reductions for Euclidean spaces include K-L transformation (or principal component analysis, PCA) [4] and FastMap [5]. On the other hand, dimension reductions such as H-Map [6] and Simple-Map (S-Map) [7] are applicable to any metric spaces metricized by L_1 distance, Hamming distance, string edit distance and so on [8].

* This work is partially supported by Grant-in-Aid for Scientific Research 17H00762, 16H02870, 16H01743 and 15K12102 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

Sketches [9–13] are representations of objects in multidimensional data by compact bit strings to reduce search spaces. In conventional search with sketches, Hamming distance between sketches is adopted. However, the mapping to sketches on Hamming distance can never imply a dimension reduction. On the other hand, since quantization of S-Map based on L_p distance [14] is regarded as a kind of sketches and provides a method to compute the distance lower bound between queries and sketches, it can provide the sketch mapping implying a dimension reduction.

The dimension reduction not only reduces distance computation cost but also avoids so-called "the curse of dimensionality". For example, it is known that the efficiency of R-Tree is decreasing when the dimension is increasing, but the performance can be improved if R-Tree is constructed on projected objects into lower dimensional space by S-Map.

In the dimension reduction, the object is projected to a low dimensional data or a compact bit string so that the projected distance does not extend with respect to the original distance. Although the projected objects of low dimensions cannot completely maintain the original distance relationship, it is important to reduce the information loss. Because the projection distance does not extend the original distance, it is guaranteed that distant objects in the projection space are far from the original space, so "safely pruning" can be done by searching in the projection space. However, if the shrinkage of the distance is large, the object outside the retrieval range actually becomes closer in the projection space, resulting in deterioration in retrieval efficiency. For PCA, analytically optimal projection can be obtained. On the other hand, for H-Map, S-Map and sketches, it has been known no analytically optimal solution, and therefore, it is necessary to use random selection with evaluation function as a clue or heuristic method such as annealing method.

In S-Map, the reference object is selected as a pivot [15–17], and the distance between each object and the pivot is set as a coordinate value, thereby the number of coordinates is given as the number of pivots. Then, the number of pivots at this time is the dimensionality of the projection space and the distance between objects in the projection space is given as the L_∞ distance. Here, a ball partitioning (BP) is to assign 0 and 1 to the inside and the outside of a ball of radius r centered on the reference object p, respectively. Then, the sketch using BP can be regarded as the quantization of S-Map image to 0 or 1 depending on whether the distance from the pivot p of the S-Map is not less than the radius r [14].

Note that conventional search methods such as random selection, local search and simulated annealing and binary quantization method using distribution characteristics of data [18] have been adopted to search a set of pivots for S-Map and BP sketches. All of them are optimized by evaluating values concerned with samples. In the S-Map, the distance preservation ratio is adopted as an evaluation value to maximize it. In BP, the collision probability is adopted as the evaluation value to minimize it. In this paper, we propose a new method named *annealing by increasing resampling* (AIR) as an optimization method and verify the effectiveness in pivot selection.

The *simulated annealing* (SA) is a search method to transit stochastically according to temperature with evaluating values from the current provisional solution to its neighbor. At the beginning, it starts from a state of high temperature and gradually lowers the temperature. At high temperature, the probability of transition to low evaluation value is high. When it has low temperature, it transits only according to the evaluation value, that is, it behaves as a local search. On the other hand, this paper proposes a method named *Annealing by Increasing Resampling* (AIR, for short), where a hill climbing is carried out by using subsample resampled from the sample used for evaluation, and the resampling number is gradually increased. While the number of resampling is small, the evaluation error for the entire sample is large, so the probability of making a transition to a low evaluation is high. That is, the transition using a small number of samples is similar to the random transition at high temperature in SA. As the number of resampling increases, the error of evaluation gradually decreases and approaches the local search. In this way, the behavior of AIR is very similar to SA.

Empirically, in order to obtain a good solution in a wide area by SA, it is necessary to increase the number of transitions at high temperature, so it takes much time to process at high temperature. On the other hand, in AIR, process to high temperature in SA corresponds to transition using a small number of resamples, and evaluation with a small number of samples is low in cost. Therefore, AIR is possible to realize processing at high temperature in low cost, which needs high cost in conventional SA.

2 Preliminaries

In this section, we briefly introduce dimension reduction, Simple-Map, and ball partitioning (BP) sketch to which the optimization method proposed in this paper is applied.

Let (U, D) and (U', D') be two metric spaces, where D and D' are distance functions satisfying the triangle inequality. The dimensionality of data x is denoted dim(x). A mapping φ:U→U' is called a *dimension reduction*, if the following conditions are satisfied for any $x, y \in U$.

$$\dim(\varphi(x)) \le \dim(x) \tag{1}$$

$$D'(\varphi(x), \varphi(y)) \le D(x, y) \tag{2}$$

Condition (1) means that it reduces the dimensionality, and condition (2) means that D' gives a lower bound of D, respectively.

A Simple-Map (S-Map) is based on the projection φ_p , using a point p called a *pivot*, defined as follows.

$$\varphi_p(x) = D(p, x)$$

From the triangle inequality, the following formula holds for $x, y \in U$.

$$\left|\varphi_p(x) - \varphi_p(y)\right| \le D(x, y)$$

Using a set $P = \{p_1, \ldots, p_m\}$ of pivots, we define an S-Map φ_P and a distance D' as follows.

$$\varphi_P(x) = (\varphi_{p_1}(x), \ldots, \varphi_{p_m}(x))$$

$$D'(\varphi_P(x), \varphi_P(y)) = \max_{i=1}^{m} \left|\varphi_{p_i}(x) - \varphi_{p_i}(y)\right|$$

Thus, when m is smaller than the original dimension, φ_P becomes a dimension reduction.

Projecting objects with S-Map, the distance between them may shrink. This shrinkage, that is, the distance deficiency, is desired to be small for similarity search. Increasing the projective dimension reduces the shrinkage of the distance, but it is strongly influenced by “the curse of dimensionality.” Thus, it is important to minimize the shrinkage of the distance in a lower dimension. The *distance preservation ratio* for a set S of pairs (x_i, y_i) of points is the following ratio of sums of distances.

$$\frac{\sum D'(\varphi(x_i), \varphi(y_i))}{\sum D(x_i, y_i)}$$

Sketches [9–13] are compact bit sequences representing multidimensional data. In this paper, we consider sketches based on *ball partitioning* (BP). A *pivot for BP* is a pair (p, r) of a point p and a radius r. A *BP projection* $\sigma_{(p,r)}$ using a pivot (p, r) is defined as follows.

$$\sigma_{(p,r)}(x) = \begin{cases} 0 & \text{if } D(p,x) \le r, \\ 1 & \text{otherwise.} \end{cases}$$

A *sketch mapping* σ_P of *width* w bits is defined by a set of pivots $P = \{(p_1, r_1), \ldots, (p_w, r_w)\}$ as follows.

$$\sigma_P(x) = \sigma_{(p_1,r_1)}(x) \ldots \sigma_{(p_w,r_w)}(x)$$

For example, let consider 4 points A, B, C and D in a Euclidian plane as in Figure 1. Then, sketches using pivots $P = \{(p_1, r_1), (p_2, r_2)\}$ are $\sigma_P(A) = 01$, $\sigma_P(B) = 00$, $\sigma_P(C) = 10$ and $\sigma_P(A) = 11$.

The conventional similarity search using sketches consists of two stages. First, candidates are selected based on Hamming distances between sketches. Then, the

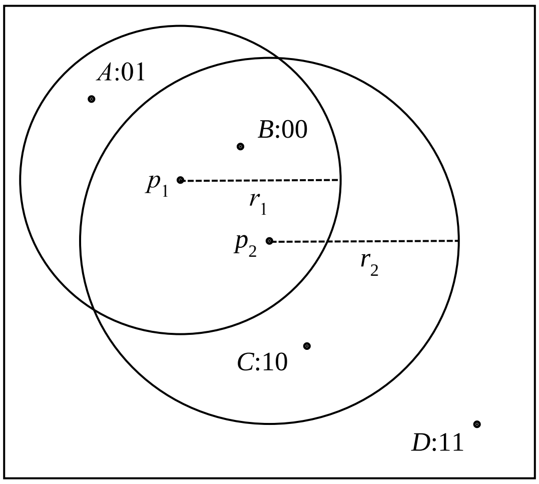

Fig. 1. Sketches using two balls

answer is selected from candidates using actual distance. As long as Hamming distance is used, sketch mapping can never imply a dimension reduction. Ohno *et.al.* [14] proposed a method to compute lower bound of distance using sketches. Therefore, such a sketch mapping can imply a kind of dimension reduction. We use the *collision probability* of a sketch mapping using a set P of pivots as the evaluation function in optimization. We say a *collision* occurs when two distinct points share the same sketch.

3 Annealing by Increasing Resampling

First we give several assumptions for optimization problem. Let Φ be the search space of possible solutions. We call an element of Φ a *state*. A *cost function* f gives the evaluation value of a state x with respect to a sample set S. Roughly speaking, an optimization problem is to find a solution x from Φ whose evaluation value is the smallest. The cost function f is desirable to satisfy the following formula for any sample sets $S_1, S_2 \subseteq S$ and any state $x \in \Phi$.

$$\min\left(f(S_1,x),\, f(S_2,x)\right) \leq f(S_1 \cup S_2, x) \leq \max\left(f(S_1,x),\, f(S_2,x)\right) \qquad (3)$$

For example, if f is defined by the average of evaluation values for individual samples such as the distance preservation ratio of S-Map, f satisfies the formula (3). The collision probability of sketch approximately satisfies the formula (3) except smaller sample sets. Further, we assume that the neighbor $N(x)$ of a state x always satisfies the following statement.

$$\forall x, y \in \Phi, y \in N^*(x)$$

Here, N^* is a reflexive and transitive closure of N. Then, the above statement claims that we can get any state y from any x by finitely many applications of N.

We present the algorithm of Annealing by Increasing Resampling (AIR, for short) in Figure 2. Here, the iteration number i of the loop is considered as time, t: $\mathbb{N} \rightarrow (0, 1]$ is a monotonic increasing function to give the ratio of resampling number with respect to total samples S and T is the total number of state transitions. Note that resampling

```
function AIR($S$:samples):state;
begin
    $x$ := any state;
    for $i$ := 1 to $T$
    begin
        $R$ := randomly selected samples of size $t(i) \times |S|$ from $S$;
        if $f(x, R) > f(y, R)$ for some $y \in N(x)$ then
            $x$ := $y$;
    end
    return $x$;
end
```

Fig. 2. Algorithm AIR

at the *i*-th iteration should be done independently to the preceding resampling. Because it is not appropriate for AIR to make the larger set by adding samples to the previous smaller set in incremental manner.

Note that, when $t(i) = 1$ for any i, AIR always uses total samples for state evaluation, thus, it behaves like so called local search. We do not care about detail of method to select a state from the neighbor $N(x)$. In practice, we may select a state with the best evaluation value within a subset of $N(x)$ in a steepest descent manner.

Since, at the beginning stage, the number of resampled samples R is small, the error of $f(x, R)$ with respect to $f(x, S)$ becomes large with high probability, and therefore, AIR may make state transition to a state with lower evaluation value. Thus, AIR makes random walks as SA at high temperature. Finally when $t(i)$ becomes close to 1, AIR behaves as local search because $f(x, R) \fallingdotseq f(x, S)$.

As for the advantage of AIR, it can make search at the beginning stage faster, because state evaluation using smaller samples can be done in low cost. On the other hand, a conventional SA needs high cost for state transitions in high temperature. There is no significant difference of AIR and SA in convergence speed, because AIR can behave almost same as SA by using resampling sizes corresponding to the annealing schedule.

4 EXPERIMENTS

In this section, we give experimental results on optimization of dimension reductions S-Map and BP sketch by the proposed method. We use two kinds of data, feature data of images (*images*) and SISAP colors database (*colors*). The number of data in *images* is 6.8 million extracted from 1,700 videos and dimensionality n of data in *images* is 64. On the other hand, the number of data in is about 0.1 million and dimensionality n of data in *colors* is 112. For both data of *images* and *colors*, each axis has integer value from 0 to 255 and distances between them are L_1.

4.1 Simple-Map

In this experiment, we adopt $m = 8$ for the dimensionality of S-Map, which shows the best performance in similarity search using R-Tree constructed by S-Map images. We use the average value (Ave.) and the standard derivation (S.D.) for distance preservation ratio (DPR) to evaluate pivot sets using randomly selected 5,000 pairs of features. AIR finds a pivot set with maximum distance preservation ratio. A pivot set $P = \{p_1, \ldots , p_m\}$ consists of mn integers corresponding to m pivots of n dimension. The neighbor $N(P)$ of a pivot set P is defined as the set of pivot sets such that any P' in $N(P)$ is the same as P but at one of mn integers. For data of *images*, features consist of 8-bit integers from 0 to 255. Therefore, $N(P)$ consists of $256mn = 256 \times 8 \times 64$ pivot sets. In our experiments, we implement AIR to randomly choose one of mn integers of P, change it from 0 to 255, and move to the best of 256 neighbors of P. That is, AIR makes a hill climbing using subsets of neighbors.

We compare AIR with conventional simulated annealing (SA), binary quantization (BQ)[18] and local search (LS). BQ is a heuristic method using stochastic property of data which can find relatively good pivot set within a small computation time. Table 1 shows the results for images. We repeat each method at 10 times. The computing times of BQ and LS are about 50 and 100 seconds, respectively. For SA and AIR, we tuned parameters of the number of state transition trials, which is corresponding to *T* in Figure 2, to compare their computing times with BQ and LS. We also run SA and AIR in about 500 seconds.

From Table 1, we can observe that AIR can find better pivot sets than BQ in about 50 seconds and LS in about 100 seconds. On the other hand, pivot sets by SA are almost comparable with BQ and LS. For every case of computing time about 50, 100 and 500 seconds, the number of state transition trials by AIR is about 8 times as large as one by SA. This experimentally shows the AIR's merit to SA pointed out in Section 3.

From Table 2, which shows the results for *colors*, we can observe the similar behavior of AIR to *images* in Table 1.

Table 1. Results for Simple-Map (*images*)

Method	Time (sec)	Trials ($\times 10^3$)	DPR (%)	
			Ave.	S.D.
BQ	47.9	—	56.5	0.329
LS	95.3	—	56.1	0.357
SA	49.2	3	56.5	0.208
	98.0	7	56.9	0.313
	511	40	57.4	0.238
AIR	47.5	24	57.3	0.260
	94.2	56	57.4	0.130
	487	330	57.5	0.154

Table 2. Results for Simple-Map (*colors*)

Method	Time (sec)	Trials ($\times 10^3$)	DPR (%)	
			Ave.	S.D.
BQ	84.6	—	83.2	0.191
LS	196	—	83.6	0.307
SA	85.7	3	83.2	0.233
	167	7	83.4	0.305
	858	40	83.6	0.324
AIR	85.0	24	83.7	0.206
	167	56	83.8	0.210
	880	330	83.9	0.145

4.2 Sketches

In this experiment, we adopt w = 32 bits as the width of sketch. Neighbors of pivot set are similarly defined as for S-Map. Radius of a pivot is selected to equally divide space by the ball. The set S of samples for evaluating pivot sets consists of randomly selected 10,000 points from database. We use collision probability (CP) to evaluate pivot set to be minimized.

We compare AIR with a conventional ball partitioning with random selection (BP), BP using binary quantization (QBP). As for observation of the search performance, we show their *precision*. Nearest neighbor search using sketches consists of two stages. At first stage, we select candidates using Hamming distance, that is, we select the top K nearest data in the meaning of Hamming distance. At the second stage, we select the nearest neighbor from the K candidates. *Search precision* is the probability that top K candidates include the exact nearest neighbor. For both databases images and colors, we adopt K as the 0.1% of the database size, which is reasonable from both viewpoints of speed and precision.

Table 3 and 4 show results for sketches on *images* and *colors*, respectively.

5 Concluding Remarks

In this paper, we have proposed a method of Annealing by Increasing Resampling (AIR, for short) to select an optimal set of pivots for dimension reduction. As shown in Table 1, 2, 3 and 4, AIR can efficiently find better sets of pivots than the conventional method from the viewpoint of evaluation function used for optimization. However, from both Table 3 and 4, from the viewpoint of search precision, the best pivot set is found by the conventional method QBP. However, this is completely the matter of evaluation function. It is a future work to explain the behavior of AIR theoretically. For example, we expect that the solution found by AIR will eventually converge to the optimum one. It is also an important future work for similarity search to inves-

Table 3. Results for Sketches (*images*)

Method	Time	CP ($\times 10^{-6}$)		Precision
	(sec)	Ave.	S.D.	(%)
BP	116	2.6	0.62	95.2
QBP	106	2.2	0.50	96.6
AIR	97.8	1.0	0.46	94.5

Table 4. Results for Sketches (*colors*)

Method	Time	CP ($\times 10^{-5}$)		Precision
	(sec)	Ave.	S.D.	(%)
BP	174	3.4	0.29	74.7
QBP	163	7.4	0.96	86.4
AIR	306	1.3	0.24	67.3

tigate other evaluation functions than distance preservation ratio for S-Map and collision probability for sketch.

References

1. Guttman, A.: R-trees: A dynamic index structure for spatial searching, Proc. SIGMOD'84, pp. 47–57 (1984).
2. Ciaccia P., Patella M., Zezula P.: M-tree: An Efficient Access Method for Similarity Search in Metric Spaces, Proc. VLDB'97, pp. 426–435 (1997).
3. Zezula, P., Savino, P., Amato, G., Rabitti, F.: Approximate similarity retrieval with m-trees, VLDB J. vol. **7**, pp. 275–293 (1998).
4. Fukunaga, K.: Statistical pattern recognition. (Second edition), Academic Press (1990).
5. Faloutsos, C., Lin, K.I.: FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets, Proc. ACM SIGMOD'95 vol. 24, pp. 163–174 (1995).
6. Shinohara, T., Chen, J., Ishizaka, H.: H-Map: A dimension reduction mapping for approximate retrieval of multi-dimensional data, Proc. DS'99, LNAI vol. 1721, pp. 299–305 (1999).
7. Shinohara, T., Ishizaka, H.: On dimension reduction mappings for approximate retrieval of multi-dimensional data, Progress of Discovery Science, LNCS vol. 2281, pp. 89–94 (2002).
8. Deza, M.M., Deza, E.: Encyclopedia of distances (Second edition), Springer (2013).
9. Dong, W., Charikar, M., Li, K.: Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces, Proc. ACM~SIGIR'08, pp. 123–130 (2008).
10. Mic, V., Novak, D., Zezula, P.: Improving sketches for similarity search, Proc. MEMICS'15, pp. 45–57 (2015).
11. Mic, V., Novak, D., Zezula, P.: Speeding up similarity search by sketches, Proc. SISAP'16, pp. 250–258 (2016).
12. Müller, A.J., Shinohara, T.: Efficient similarity search by reducing I/O with compressed sketches, Proc. SISAP'09, pp. 30–38 (2009).
13. Wang, Z., Dong, W., Josephson, W., Lv, Q., Charikar, M., Li, K.: Sizing sketches: A rank-based analysis for similarity search, Proc. ACM SIGMETRICS'07, pp. 157–168 (2007).
14. Onho, S., Murakami, Y., Kawaguchi, H., Kishikawa, N., Shinohara, T.: Similarity search of multi-dimensional database by quantization dimension reduction mapping, Hinokuni Information Processing Symposium 2012, ISPJ Kyushu (in Japanese) (2012).
15. Chavez, E., Navarro, G., Baeza-Yates, R., Marroqu, J.: Searching in metric spaces. ACM Comput. Surv. vol. 33, pp. 273--321 (2001).
16. Mao, R., Miranker, W., Miranker, D. P.: Pivot Selection: dimension reduction for distance-based indexing. J. Discret. Algo. vol. 13, 32--46 (2012).
17. Mao, R., Zhang, P., Li, X., Liu, X., Lu, M.: Pivot selection for metric-space indexing, Internat. J. Mach. Learn. Cybernet. vol. 7, pp. 311--323 (2016).
18. Hau, N., Shinohara, T.: Pivot selection in dimension reduction projection Simple-Map with quantization, Hinokuni Information Processing Symposium 2015, ISPJ Kyushu (in Japanese) (2015).