ORIGINAL ARTICLE

Pivot selection for metric-space indexing

Rui Mao[1] · Peihan Zhang[1] · Xingliang Li[2] · Xi Liu[1] · Minhua Lu[3]

Received: 17 December 2015 / Accepted: 23 January 2016 / Published online: 3 February 2016

Abstract Metric-space indexing abstracts various data types into universal metric spaces and prunes data only exploiting the triangle inequality of the distance function in metric spaces. Since there is no coordinates in metric space, one usually first pick a number of reference points, pivots, and consider the distances from a data point to the pivots as its coordinates. In this paper, we first survey and discuss the state of the art of pivot selection for metric-space indexing from the perspectives of importance, objective function, number of pivots, and selection algorithm. Further, we propose a new objective function, a new method to determine the number of pivots and an incremental sampling framework for pivot selection. Experimental results show that the new objective function is more consistent with the query performance, the new method to determine the number of pivots is more efficient, and the incremental sampling framework leads to better query performance.

Keywords Metric-space indexing · Pivot selection · Intrinsic dimension · Objective function · Range query

1 Introduction

Among the characteristics or challenges of big data, “variety” is relatively less studied. To conquer “variety”, one can first find a universal abstraction covering various data types, and then build data management and analysis system based on the characteristics of the universal abstraction. Such system works for any particular data type that is a special case of the universal abstraction [1].

Metric space [2] has been proposed as a universal abstraction for big data [1]. Metric space does not make any requirement of the intrinsic structure of data, but only a distance function, with the non-negativity, symmetry and triangle inequality properties, of pairs of data points [2].

Since there is no coordinates in metric space, many mathematical tools cannot be directly applied. As result, a common first step of data management and analysis in metric space is to impose coordinates to data. To do so, one can pick a number of reference points, named pivots [3–6]. For an arbitrary data point, its distances to the pivots can be calculated and these distances form the imposed coordinates of this data point [7].

According to the pivot space model [7], pivot selection in metric space is analogous to dimension reduction in multi-dimensional space. The coordinate information available to data processing steps that follow is determined by pivot selection. As a result, pivot selection is of critical importance in metric-space data management and analysis.

✉ Minhua Lu
luminhua@szu.edu.cn

Rui Mao
mao@szu.edu.cn

Peihan Zhang
554178445@qq.com

Xingliang Li
lxl_ustc@163.com

Xi Liu
liuxiwudi@hotmail.com

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

[2] College of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China

[3] School of Medicine, Shenzhen University, Shenzhen 518060, China

There are at least three issues that need to be settled down in pivot selection. The first issue is the objective function of pivot selection, or the criterion in comparing two set of pivots. The second issue is the number of pivots to be selected. The more are the pivots, the more information they provide, but also the more space occupied. It is worthwhile to achieve a good balance between information available and space occupation. The third issue is the pivot selection algorithm. Given the objective function and the number pivots, the importance of a pivot selection algorithm determining good pivots with reasonable time and space consumption is obvious.

Our contributions are two-folds. First, we briefly survey the state of the art of pivot selection in metric space from the perspectives of objective function, number of pivots and selection algorithm. To the best of our knowledge, this is the first paper surveying pivot selection from the above three perspectives.

Second, we propose new ideas for all the three perspectives above. That is: (1) we propose a new radius-sensitive objective function for pivot selection for metric-space indexing. Experimental results show that the new objective function is more consistent with query performance than existing ones; (2) we propose a new method to determine the number of pivots based on the eigenvalues of the coordinate matrix. Experimental results show that this method is of comparable performance to one of the best existing methods, but is computationally simpler; (3) an incremental sampling pivot selection framework is proposed. Experimental results shows its superiority to existing methods.

The remainder of this paper is organized as follows: The importance of pivot selection is discussed in Sect. 2. We survey existing objective functions and propose a new objective function in Sect. 3. A survey of number of pivots determination methods and a new method to determine the number of pivots are presented in Sect. 4, followed by a survey of existing pivot selection algorithms and an incremental sampling pivot selection framework in Sect. 5. Experimental results are presented in Sect. 6, followed by conclusions and future work in Sect. 7.

2 The importance of pivot selection

Let S = {x_i | i = 1, 2, …, n} be a dataset in metric space, d be the distance function and P = {p_j | j = 1, 2, …, k} be a set of pivots selected from S. The above notation will be followed throughout this paper.

According to the pivot space model [7], pivot selection defines a mapping from metric space to a k-dimensional space, named the pivot space. For an arbitrary point x in S, its image, x^p, in the pivot space derived by P is:

$$x^p = (d(x, p_1), \ldots, d(x, p_k)).$$

When k = n, i.e. all points in S are selected as pivots, the derived pivot space is called the complete pivot space [7]. It has be proved that the mapping to the complete pivot space is isometric [7] with respect to the L∞ distance in the complete pivot space, where

$$L_\infty((a_1, a_2, \ldots, a_n), (b_1, b_2, \ldots, b_n)) = max_i(|a_i - b_i|)$$

When k < n, pivot selection is analogous to dimension reduction in multi-dimensional space. Since pivot selection is usually the first step of data processing in metric space, it determines the coordinate information available to subsequent data processing steps. Therefore, the importance of pivot selection is obvious.

In the following, we present four examples to show the different effect of different pivots.

Example 1 Let's consider a dataset consists of three points A, B and C with values 1, 2 and 3, respectively. Figure 1 shows the pivot spaces with A, B or C as the pivot, respectively. When A or C is the pivot, all three points are distinguishable in the pivot space, while A and C are not distinguishable in the pivot space when B is the pivot. This example gives a heuristic that corners of data might form good pivots. Please note that this is different from a common heuristic in the area of clustering, where centers are usually picked to represent the clusters.

Example 2 Let's consider a dataset consists of three thousand points randomly selected from unit square, and two pivots are to be selected. According to Example 1, corners might be good pivots. Figure 2 show the pivot spaces with opposite corners or neighboring corners as the pivots. Apparently, when neighboring corners are pivots, data distribute more widely, or data is more distinguishable, than the case that opposite corners are pivots. This example gives a heuristic that neighboring corners are better than opposite corners.

Example 3 This example (Fig. 3) is similar to Example 2 except that data was randomly sampled from then unit ball. Similar heuristic can be drawn here.

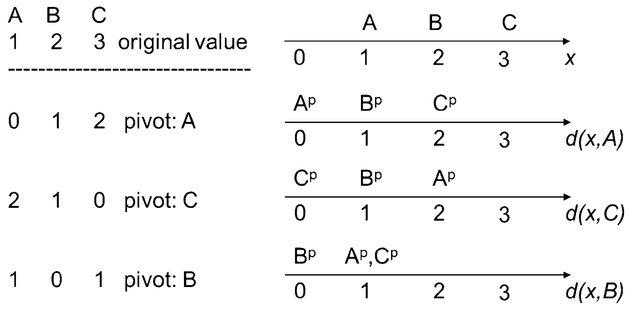


Fig. 1 Pivot spaces of three numbers

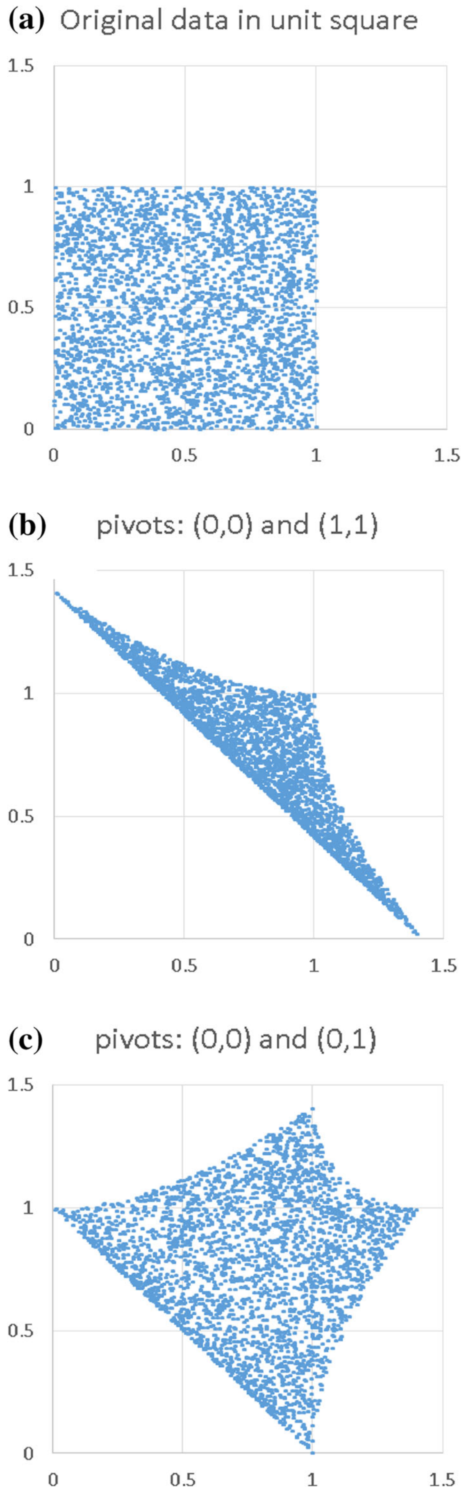


Fig. 2 Pivot spaces of 3000 points randomly selected from the unit square with Euclidean distance: **a** original data, **b** pivots (0,0) and (1,1), **c** pivots (0,0) and (0,1)

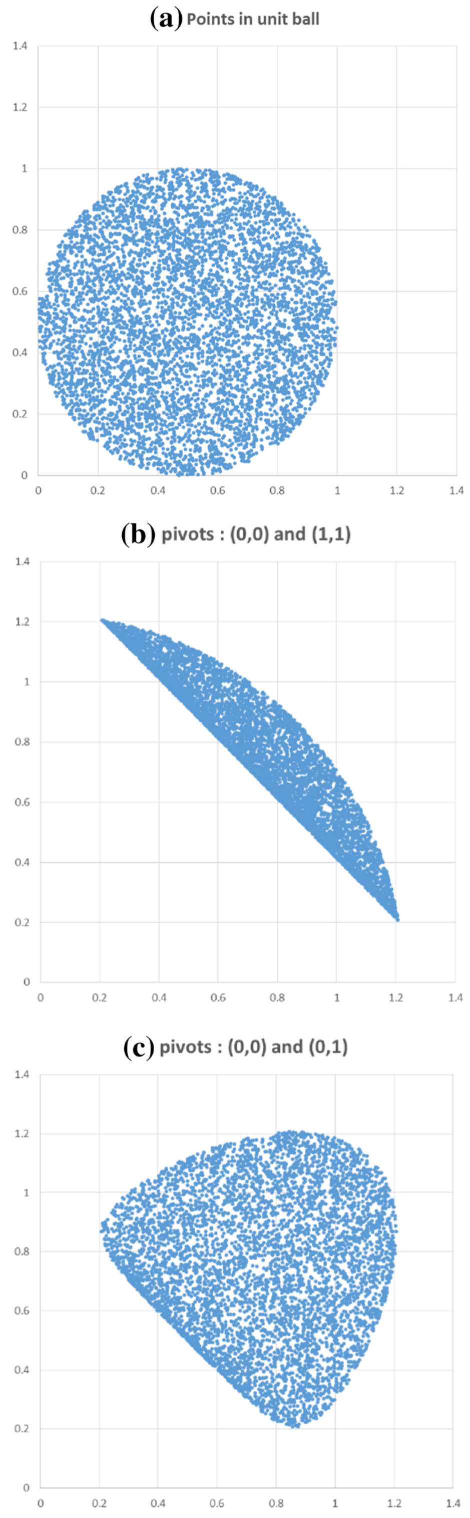


Fig. 3 Pivot spaces of 3000 points randomly selected from the unit ball with Euclidean distance: **a** original data, **b** pivots (0,0) and (1,1), **c** pivots (0,0) and (0,1)

Example 4 This example considers all the 65,536 strings consist of only 0 or 1 of length 16 with Hamming distance. In this example, when opposite corners, i.e. “0000 0000 0000 0000” and “1111 1111 1111 1111”, are pivots (Fig. 4a), data in the pivot space fall on a line. This is because the Hamming distance between a “0/1” string and

"0000 0000 0000 0000" is the number of "1"s in the string, and the Hamming distance between a "0/1" string and "1111 1111 1111 1111" is the number of "0"s in the string. As a result, the sum of the distances is the length of the string, 16. Consequently, the points in Fig. 4a fall on the line $x + y$=16. When neighboring corners, i.e. "0000 0000 0000 0000" and "0000 0000 1111 1111", are pivots (Fig. 4b), data in the pivot space distribute more widely and are more distinguishable. Again, this example indicates that neighboring corners are better pivots than opposite corners.

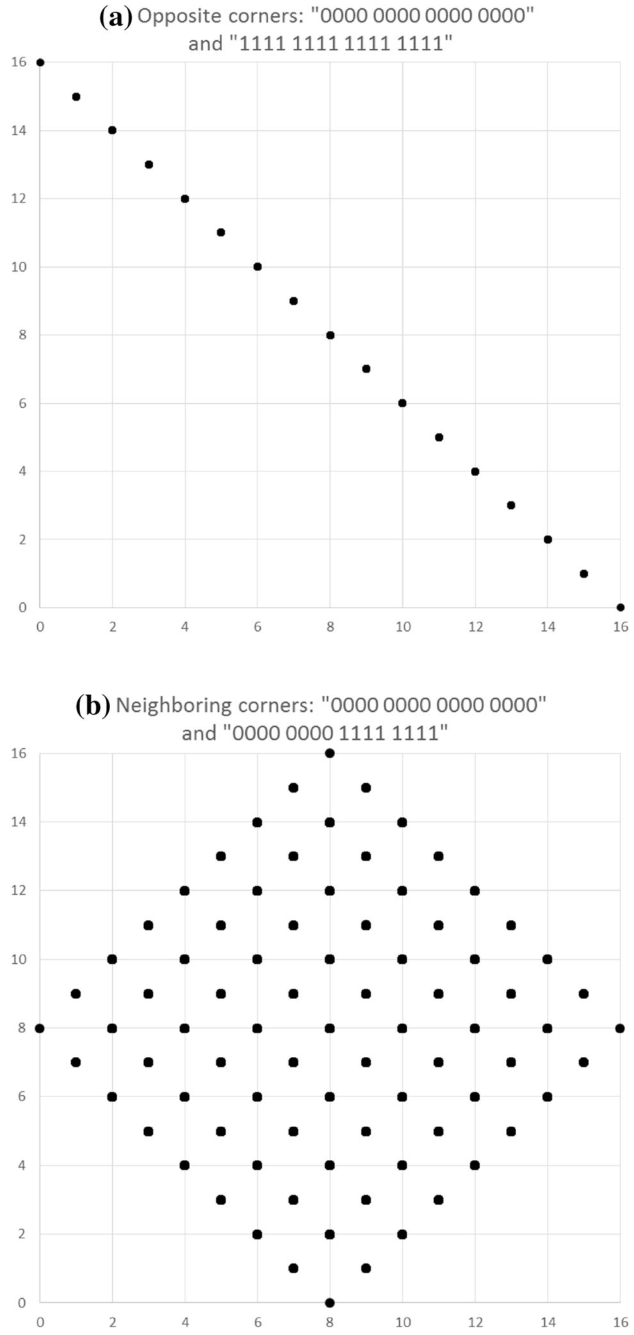


Fig. 4 Pivot spaces of 65536 binary strings of length 16 with Hamming distance

3 Objective function

Most existing work on pivot selection focus on heuristic selection algorithm, and the objective function is less studied. The ultimate goal of pivot selection is the query performance of index. Therefore, the best or the real objective function of pivot selection is the query performance. Unfortunately, query performance is not available during pivot selection. As a result, all existing objective functions try to estimate the query performance. In the following, we first introduce existing objective functions, and then propose a radius-sensitive objective function for metric-space similarity indexing.

3.1 The state of the art

We enumerate popular objective functions in the following.

3.1.1 Maximum variance

Vantage point tree (VPT) [8, 9] is an indexing tree for similarity search in metric space. It only exploits one pivot for each index node. According to the pivot space model [7], with one pivot, data is mapped into a one-dimensional pivot space. VPT's objective function for pivot selection is the variance of data in the one-dimensional pivot space. That is, VPT selects the pivot maximizing the variance of data in the pivot space. Venkateswaran et al. [30] adopt the same idea, i.e.:

$$p = argmax_t(Var(d(x, t)), x \in S, t \in T)$$

This objective function can be used iteratively to select more pivots.

3.1.2 Maximally separated

The LAESA [33] proposes to select pivots that are maximally separated. That is, the sum of the inter-pivots distances should be maximized:

$$P = argmax_T(\sum d(x, y), x, y \in T, T \subseteq S, |T| = k)$$

Traina et al. [38] propose to select pivots near the hull of the dataset. They choose points that have the most similar distances to the previously chosen pivots as new pivot. The underlying objective function is very similar to this Maximally Separated one.

3.1.3 Priority vantage point

The KVP structure [31, 32] proposes the idea of priority vantage point. That is, pivot is better if close or distant to query or data, and the closest is better than the furthest for clustered data. Therefore, as an extension of LAESA, they first select a large set of pivots by some methods, and then, for each data object, only store its distances to a subset (close or distant) of the pivots. During search, the order of pivots being used is changed according to their pruning efficiency.

Venkateswaran et al. also adopt this idea [30].

3.1.4 Maximum mean

Bustos et al. [11] concern the mean and variance of the pair-wise L_∞ distance of data in the k-dimensional pivot space. Among the variations, the best objective function is the mean. That is, the best set of pivots should be the one maximizing the mean of pair-wise L^∞ distance of data in the pivot space.

$$P = argmax_T\left(\sum L_\infty\left(F_{T,d}(x),\ F_{T,d}(y)\right), x, y \in T, T \subseteq S, |T| = k\right)$$

The rationality of Bustos's object function is explained next. According to the pivot space model [7], the pair-wise distance of data does not increase from metric space to pivot space. That is, for two arbitrary points, x and y, in S, d(x, y) $\geqq L_\infty$ (x^p, y^p). As a result, pivot selection, or mapping data to pivot space, always loses distance information. Bustos's objective function actually try to minimize the loss of distance information.

VPT's objective function considers only one pivot. Bustos's objective function considers multiple pivots, and is widely adopted. However, experimental results show that it might not be consistent with the real goal of data processing task. Please see Sect. 7 for details.

3.1.5 Corner selection

Shapiro [36] experimentally show that pivot should be outside the data cluster. Further, when two pivots are selected, neighboring corners are better than opposite corners [36].

3.1.6 Spacing-correlation Based

Veltkamp et al. [34, 35] suggest that pivots should produce large spacing and the correlation between them should be small.

For spacing, they suggest that points in the one-dimensional pivot space created by a pivot should have large spacing, or small variance of the spacing, i.e.:

$$\mathrm{Var}(d(p,x_{i+1}) - d(p,x_i)), d(p,x_1) \leq \ldots \leq d(p,x_n)$$

The correlation between two pivots, p1 and p2, is the correlation between their distance to other points, i.e.:

$$\mathrm{Cor}\left((d(p_i,x_1),\ldots,d(p_i,x_n)),\left(d(p_j,x_1),\ldots,d(p_j,x_n)\right)\right)$$

3.1.7 Sparse spatial selection (SSS)

Brisaboa et al. [41] propose to make pivots far to each other. Let M be the maximal possible distance, and α be a threshold, the distance between any two pivots should be not less than Mα, i.e.:

$$P = argmax_T\left(|T|, T \subseteq S, \forall_{t_1,t_2 \in T}(d(t_1,t_2) \geq M\alpha)\right)$$

3.1.8 Maximum pruning

For a range query R(q,r), the condition that a point x can be pruned by p for q is |d(p,q)-d(p,x)| > r. While some objective functions aim to increase the possibility for which the condition holds, the condition itself can serve as an objective function, probably the ultimate and the most direct one. Assuming the query have the same the distribution as the data base, Berman and Shapiro use this objective function iteratively to select multiple pivots [43].

Venkateswaran et al. [30] consider the number of points can be pruned as the objective function, and name this object function Maximum Pruning. The query set and range query radius are parameters, i.e.:

$$P = argmax_T\left|(q,x)|x \in S, q \in Q, L_\infty\left(F_{T,d}(x), F_{T,d}(q)\right) \geq r\right|, |T| = k, T \subseteq S$$

3.2 Radius-sensitive objective function

Our radius-sensitive object function is designed particularly for pivot selection to support range query in metric space. A range query R(q, r) [3] returns all data points in S that is within distance r to q. It is usually answered by descending an index [3–6].

Typically, each index node is defined by a few pivots and the distances, or ranges of them, from data in the sub-tree to the pivots. The key to speed up the query is to prune as much data as possible. For example, given a pivot p, a data point x in the sub-tree of p, and their distance d(p, x) stored in the index node, if d(p, x) + r < d(p, q), x can be safely pruned [3–6].

The goal of pivot selection in the construction of index tree for range query is the average performance of range queries, where query object q is commonly assumed to distribute the same as the database.

The radius-sensitive object function is an extension of the maximum pruning one for the case of multiple pivots, and assuming the query have the same distribution as the data base:

$$P = argmax_T \left|(x,y)|x,y \in S, L_\infty\left(F_{T,d}(x), F_{T,d}(y)\right) \geq r\right|, |T| = k, T \subseteq S$$

Experimental results show that radius-sensitive objective function is more consistent with query performance than others.

4 Number of pivots

The number of pivots to select is important at least because it is a parameter to pivot selection algorithms. More pivots provide more distance information, but also take up more space.

As discussed before, VPT [8, 9] selects one pivots while MVPT [10] selects multiple. Brin suggests to select more pivots for larger partitions of data to maintain tree balance in GNAT [21].

A hypothesis is that the number of pivots should be close, if not equal, to the intrinsic dimension of data [7, 38].

Intuitionally, the intrinsic dimension is the "real" dimension of data ignoring its representation. If the number of pivots is larger than the intrinsic dimension, data is embedded into spaces with more dimensions than its real number of dimensions. Since the amount of information is not increased, adding more pivots only leads to redundancy.

The following of this section focuses on estimation of intrinsic dimension.

4.1 The state of the art

A number of mathematical definitions and estimation methods have been proposed for intrinsic dimension [12–14]. Unfortunately, these definitions are usually too mathematically strict and the estimations methods are usually to time costly to be applied in metric-space indexing. In the following, we introduce estimation methods proposed for metric-space indexing.

Method 1 Let μ and σ^2 be the mean and variance of the pairwise distances of data in the metric space. Chavez et al. (2001) define the intrinsic dimension of a metric space as $\rho = \mu^2/2\sigma^2$.

Method 2 Let r be the radius, and n be the average number of points within distance r to a given point. This method assumes that n is proportional to r^ρ, analogous to that the volume of a ρ-dimensional hyper-ball is proportional to r^ρ. Values of n and r can be collected by experiments, linear regression can be performed on the logarithm of n and r, and the resulting slope coefficient is an estimate of the intrinsic dimension [15, 16].

Methods 1 and 2 are for metric space. The pivot space model [7] builds a bridge between metric space and multi-dimensional space, especially the complete pivot space. As a result, methods to estimate the intrinsic dimension of a multi-dimensional space can be applied to the complete pivot space to get an estimate of the intrinsic dimension of a metric space [7].

Method 3 Let $Q = \{q_1, q_2, \ldots, q_n\}$ be the principal components of the data in the complete pivot space. Let $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be the eigenvalues, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$. The intrinsic dimension is estimated as

$$\rho = argmax_i(\lambda_i / \lambda_{i+1}),\ i = 1, \ldots, n-1$$

[7]. Method 1 is computationally simple, and experimental results shown that it is asymptotically accurate [7]. Similar forms was used to determine the stability of workloads [17, 18].

Method 2 can be regarded as a variation of the Minkowski fractal dimension [12, 19], and was also shown to be asymptotically accurate. It is limited by the assumption of uniform distribution and is sensitive to the values of radius.

Method 3 was also shown to be quantitatively accurate for data whose intrinsic dimension are known [7].

All the three methods are computationally very costly, with O(n3) original time complexity or similar. Since Method 3 is for multi-dimensional space, some mathematical technique, such as the EM method [20], can be applied to reduce the time complexity.

4.2 A distance matrix eigenvalue method

In Method 3, to perform PCA, one first compute the covariance matrix of the complete pivot space, and then compute the eigenvalues of the covariance matrix. The step to compute the covariance matrix is very costly, with $O(n^3)$ original time complexity. To reduce the computation cost, we propose the distance matrix eigenvalue method, which is only different from Method 3 in that the computation of the covariance matrix is skipped, and eigenvalues are computed directly from the complete pivot space, or the distance matrix. The eigenvalues computed this way might be negative, in which case their absolute values are used instead.

There are three variations of this method:

1. $\rho = \text{argmax}_i\ (|\lambda_i|/|\lambda_{i+1}|)$;
2. $\rho = \text{argmax}_i\ (|\lambda_i| - |\lambda_{i+1}|)$; and

3. $\rho = \text{argmin}_i \left(\sum_{j=1}^{i} |\lambda_j| / \sum_{j=1}^{n} |\lambda_j| \geq 0.7\right)$, i = 1, …, n-1.

Experimental results comparing this method and Method 3 are presented in Sect. 6.2.

5 Pivot selection algorithm

In this section, we first survey existing pivot selection algorithms, and then propose the incremental sampling pivot selection framework.

5.1 The state of the art

The Metric Tree (M-tree) selects pivots randomly Ciaccia [22].

The spatial approximation tree (SA-tree) selects the centers of cells of a Voronoi diagram as pivots [23].

Other popular pivot selection algorithms are introduced next.

5.1.1 *Corners as pivots*

Using corners as pivots is a widely adopted heuristic. Example 1 gives an illustration when data is one-dimensional. Yianilos further analyzed this heuristic in VPT [9]. The data under concern are points uniformly distributed in the unit square. VPT selects on pivot and then draw a circle, centered at the pivot, to divide the data equally into two parts. Natural choices of pivots are center of the square, p_m, middle point of an edge, p_e, or a corner, p_c (Fig. 5 [9]). In the context of range query, it can be deduced that the length of the circle is an indicator of the performance of the pivot, the shorter the better [9]. It can be geometrically proved that p_c has the shortest curve among the three candidates, with the radii and lengths of circles marked in Fig. 5 [9]. As a result, the eligibility of selecting corners as pivots is established. Bustos et al. [11] also concluded that good pivots are usually corners, while the reverse might not true.

5.1.2 *Farthest-first-traversal*

The farthest-first-traversal (FFT) k-center clustering algorithm [24, 25] is usually applied to find corners, because of its linear time and space complexity. FFT runs iteratively. The first pivot is selected at random, and the next pivot is the point whose smallest distance to existing pivots is the maximum:

$c_1 =$ random or other selection

$c_k = argmax_x(min_{i=1}^{k-1} d(x, c_i)),\ x \in S,\ k \geq 2$

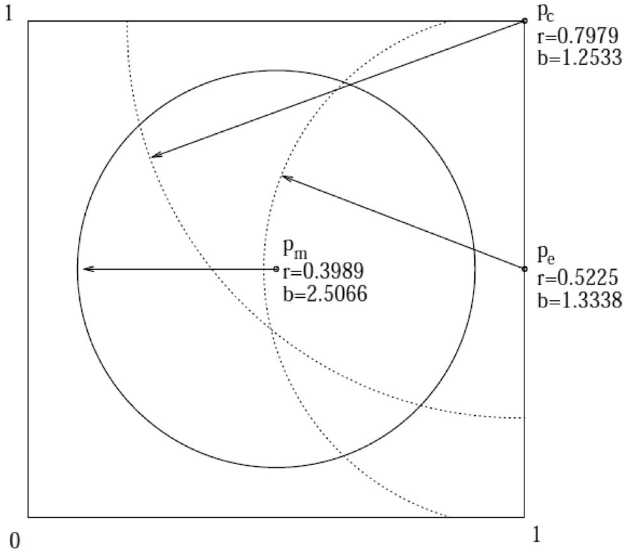


Fig. 5 Choices of pivot to divide the unit square [9]

FFT minimizes the maximum cluster diameter and gives a result at most twice the optimal diameter [25].

Vleugels and Veltkamp [34], in an AESA-alike structure, uses FFT, which is named MaxMin by Hennig and Latecki [40]. Mao et al. [39] use FFT in an extension of M-tree. Berman and Shapiro also use FFT, which is named Cluster [43], while the first two pivots is the pair furthest apart.

5.1.3 *Sparse spatial selection (SSS)*

Brisaboa et al. [41] propose to make pivots far to each other. Let M be the maximal possible distance, and α be a threshold, the distance between any two pivots should be not less than Mα. The selection algorithm is an iterative greedy one. A point is selected as pivot if its minimal distance to previously selected pivots is not less than Mα. They experimentally show that 0.35 ≤ α≤0.4 is better, but use 0.5 in their own experiments. This algorithm does not need a number of pivots as input, and works for dynamic data set.

Bustos et al. [42] combine SSS with their previous work on incremental sampling method [11]. In dynamic case, it adds every new point selected by SSS as pivot. If the pivot's contribution is larger than an existing pivot, remove the least contribution pivot. The contribution of a pivot is summed over all evaluation pairs. If it is not the best pivot for a pair, the contribution for that pair is 0. If it is, let the second best pivot be p2, then the contribution is: $|d(p,x)-d(p,y)|- |d(p_2,x)-d(p_2,y)|$

5.1.4 *Maximum pruning*

Venkateswaran et al. [30] consider the number of points can be pruned as the objective function. The query set and range query radius are parameters. They use sampling based on statistical gain to speed up the greedy algorithm.

Berman and Shapiro [43] propose the Greedy Separation algorithm which selects pivot one at a time. When count the number of pairs that can be pruned by a candidate pivot, the pairs already pruned by previous pivots are discounted.

5.1.5 Maximum variance

Venkateswaran et al. [30] consider the variance of data in one-dimensional pivot space as the objective function. The pivots are selected iteratively. Each time, the point with the largest variance is picked, and those points that can be pruned by this pivot is removed from the candidate set.

Berman and Shapiro [43], in their heuristics Variance, just simply pick the pivots of the largest variances on a sample without considering the correlation among pivots.

5.1.6 Maximally separated and hull of foci

The LAESA [33] proposes to select pivots that are maximally separated. That is, the sum of the inter-pivots distances should be maximized. Their selection algorithm is an iterative one. The first pivot is randomly selected. The next pivot is the one whose sum of distances to previously selected pivots is the largest.

$$p_1 = random\, selection$$

$$p_i = argmax_x\left(\sum_{j=1}^{i-1} dx, p_j\right), x \in S - P), i = 2, .., k$$

Obviously, the time cost of this algorithm is O(kn).

Traina et al. [38] propose to select pivots near the hull of the dataset. Their HF (Hull of Foci) algorithm chooses points that have the most similar distances to the previously chosen pivots as new pivots:

$p_1 = \mathrm{argmax}_y \mathrm{d}(x,y)$, x: a random point
$p_2 = \mathrm{argmax}_y \mathrm{d}(p_1,y)$, edge $= \mathrm{d}(p_1, p_2)$

$$p_i = argmin_y\left(\sum_{j=1}^{i-1}\left|edge - d\left(p_j, y\right)\right|, y \in S - P\right), i > 2$$

Since edge is normally large, these two selection algorithm are almost equivalent.

5.1.7 Spacing-correlation based

Veltkamp et al. [34, 35] suggest that pivots should produce large spacing and the correlation between them should be small. The selection is a greedy one, which takes a random set of pivots and two thresholds for variance and correlation as inputs. It scans the data and replaces a pivot with a new random one if its variance or correlation go beyond the thresholds.

5.1.8 Principal component analysis

Ramasubramanian and Paliwal [37] use PCA in the original multi-dimensional data space for pivot selection, i.e. one at the original, then one for each PC.

The pivot space model makes it possible to study the pivot selection problem in a multi-dimensional space, the complete pivot space [7]. Mao et al. [7] propose to apply dimension reduction techniques in multi-dimensional space for pivot selection. That is, first run dimension reduction in the complete pivot space to generate new dimensions, and then find points close to the new dimensions as pivots. As a demonstration, they apply PCA and experimentally show that this method outperforms others [7].

5.1.9 Incremental sampling

Bustos et al. [11] exploit sampling and seek to choose a set of pivots maximize the objective function discussed in Sect. 4, which is the mean of the pair-wise L∞ distances of data in the pivot space.

Let SetA be a set of A pairs of points to calculate objective function, the evaluation set. Let SetN be a set of N pivots, the candidate set. SetN is randomly selected for the selection of each pivot.

$$p_1 = argmax_t(Var(d(t,x)), x \in S), t \in S$$

$$p_i = argmax_t\left(\sum_{(x,y)\in SetA} L_\infty\left(F_{P\cup\{t\}}(x), F_{P\cup\{t\}}(y)\right)\right),$$

$$t \in SetN, i = 2, \ldots, k$$

The authors suggest to use larger value of A and smaller value of N, e.g. large evaluation set and small candidate set.

Experimental results on synthetic vector data sets, with dimension 8 and 14, show that Maximally Separated pivots slightly outperforms Bustos's incremental sampling. However, on real world data set, NASA and color images, Bustos's pivots are better, especially when the number of pivots is small.

The incremental sampling pivot selection framework we propose next is based on Bustos et al.'s work.

5.2 An incremental sampling pivot selection framework

The basic idea of the incremental sampling framework for pivot selection is:

1. Incrementally select pivots one at a time;
2. In each iteration, each point of a pre-determined candidate set is combined with pivots selected by previous iterations, and then a value of a pre-defined objective functions over a pre-determined evaluation set of pairs of points is computed. Finally the point with the best value of the objective function is selected as the next pivot.
3. The algorithms to determine the objective function, the candidate set, and the evaluation set are configurable.

The steps of the incremental sampling pivot selection algorithm is illustrated in Fig. 6.

Two objective functions are tested in our implementation, i.e. Bustos's objective function and the radius-sensitive objective function proposed in Sect. 3.

Two heuristics to determine the candidate set are tested:

1. Random sampling;
2. Corners found by FFT.

Please note that the idea to use corners as candidate set is supported by Bustos's observation that good pivots are usually corners [11]. The same idea is exploited by Mao et al. [7] to speed up the computation of PCA for pivot selection.

Two heuristics to determine the evaluation set are tested:

1. Random sampling;
2. For each point, randomly sample a number of points to form pairs with it.

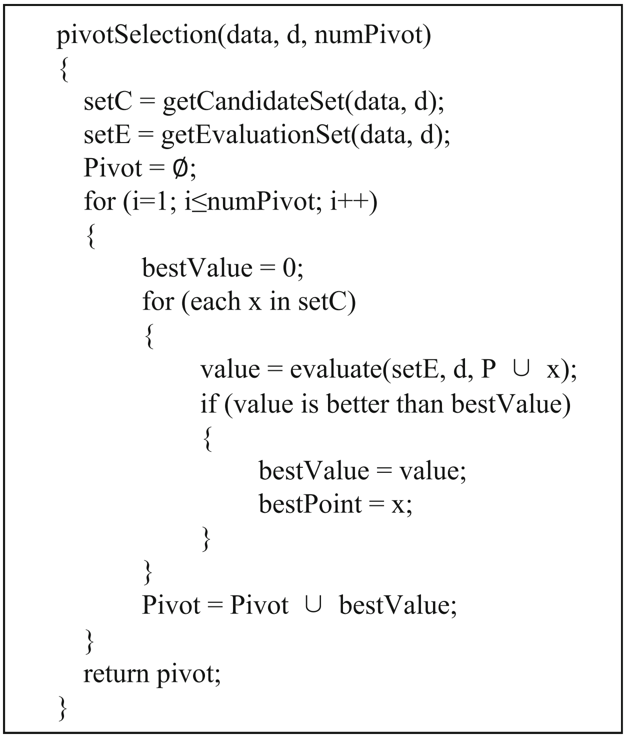
Please note that Bustos et al.'s incremental pivot selection algorithm is a special case under this framework with objective function (1), candidate set (1) and evaluation set (1).

Experimental results show that for objective function (2), candidate set (2) and evaluation set (2) form the best combination among the four possible ones. It steadily outperforms Bustos et al.'s algorithm.

6 Experiment results

In this section, we present experimental results on objective function, estimation of intrinsic dimension, and pivot selection algorithm.

The test datasets consist of uniformly distributed vectors of up to 20 dimensions, US cartographic boundary data (2-dimensional coordinates) of Texas and Hawaii, and protein sequence fragments of length 6, all from the test suite of the UMAD (Universal Management and Analysis of Data) project [26]. For uniform vector and boundary data, the distance function is the Euclidean distance. The protein sequence is represented as a string, and the distance function is global alignment [27], a form of weighted edit distance, with mPAM [28] as the substitution matrix. These datasets are summarized in Table 1.



```
pivotSelection(data, d, numPivot)
{
  setC = getCandidateSet(data, d);
  setE = getEvaluationSet(data, d);
  Pivot = ∅;
  for (i=1; i≤numPivot; i++)
  {
      bestValue = 0;
      for (each x in setC)
      {
          value = evaluate(setE, d, P ∪ x);
          if (value is better than bestValue)
          {
              bestValue = value;
              bestPoint = x;
          }
      }
      Pivot = Pivot ∪ bestValue;
  }
  return pivot;
}
```

Fig. 6 The incremental sampling pivot selection algorithm

6.1 Objective function

The objective functions are compared by range query performance on vector, DNA and protein data. That is, for each data set, indexes are built for the two objective functions, a number of range queries with various radii are executed, and the average number of distance calculations, which is independent to implementation and hardware/software environment, and computed as the performance metric.

To focus on the performance of objective functions of pivot selection, and avoid the influence of data partitioning, the index structure is Pivot Table [29], for which partitioning is not involved and data is sequentially scanned and pruned by their distance to pivots. Further, to avoid heuristic pivot selection algorithms, brutal force method is employed to select pivots based their value of objective functions. Due to the high computational cost, the

Table 1 Summary of test suite

Workload	Total size	Distance function	Domain dimension
Uniform vector	1 M	Euclidean distance	1-20
Hawaii	9 k		2
Texas	190 k		2
Protein	100 k	Global alignment	6

Table 2 The performance difference of pivot selection objective functions

Radius	2d vector	3d vector	Radius	DNA	Protein
0.02	0.038	0.242	1	4.269	0
0.04	0.094	0.556	2	3.801	0
0.06	0.026	0.918	3	0	0
0.08	0.04	1.594	4	0	0
0.1	0.046	1.696	5	0	0
0.12	0.09	1.424	6	0.501	0
0.14	0.028	1.522	7	2.302	0
0.16	0.004	1.102	8	5.475	0.216
0.18	0.078	0.64	9	5.865	0.666
0.2	0.02	0.206	10	3.809	0.874

Bustos's—radius-sensitive

experiments are based on data sets of size 1000. To make the experiments accurate, all the data points in the data sets are used as query objects.

The experimental results show that the performances of the two objective functions are very close. To show the difference, in Table 2 we show the difference obtained by subtracting the average number of distance calculations to answer range queries with the radius-sensitive objective function from that of Bustos's. Although the differences are tiny, there are all non-negative and mostly positive. That is, the radius-sensitive objective function outperforms Bustos's for most of the cases, and is never outperformed by it.

6.2 Estimation of intrinsic dimension

The distance matrix eigenvalue method proposed in Sect. 4.2 with its three variations is compared with Method 3 surveyed in Sect. 4.1, which has been shown to outperform Methods 1 and 2.

For uniform vector data, dimensions from 1 to 10 are used. For every dataset, the estimates of intrinsic dimension of the 3 criteria of our method and Method 3 are listed in Table 3. It is obvious that all the 3 variations of the proposed method yield comparable accuracy to Method 3. Since this distance matrix eigenvalue method skips the computation of the covariance matrix, its superiority to Method 3 is conspicuous

6.3 Pivot selection algorithm

Under the incremental pivot selection framework, there are three components to configure, i.e. the objective function, the choice of the candidate set, and the choice of evaluation set. For each of the three components, we have presented two choices, thus there are 8 combinations. Preliminary experimental results show that the combination with the radius-sensitive objective function, option (2) for candidate set, and option (2) for evaluation set (see Sect. 5.2) yields out the best performance. Therefore, we compare this combination with Bustos et al.'s incremental sampling algorithm next, in terms of similarity query performance.

For each dataset, 100 thousand points from the beginning of the data file are picked on which an index tree is built for our best combination and Bustos's algorithm, respectively. For each index tree, 5000 points from the beginning of the data file are picked as range query objects, with a series of radii. To focus on the algorithmic aspect, the average number of distance calculations to answer a range query, which is independent of implementation and hardware/software environment, is used as the performance metric. Results on all the datasets show similar trend, thus only results on 20-dimensional vectors and protein sequences are plotted in Fig. 7. It shows clearly that our best combination always outperforms Bustos et al.'s algorithm.

7 Conclusion, discussion and future work

Metric space can serve as a universal abstraction of a wide range of data types, and building system for metric space is an effective approach to conquer the "variety" challenge of big data. Pivot selection imposes coordinates to data in metric space and is usually the first step of metric-space data management and analysis.

In this paper, we first show the importance of pivot selection, then survey the state of the art of pivot selection from the perspectives of objective function, number of pivots, and selection algorithm, and finally make new proposals for pivot selection from each of the perspectives above.

Bustos et al.'s objective function generally performs well. The radius-sensitive object function is particularly for

Table 3 Estimate of intrinsic dimension

Data	Domain dim.	Intrinsic dimension			
		Distance matrix eigenvalue method			Method 3
		(1)	(2)	(3)	
Uniform vector	1	2	2	2	2
	2	3	3	3	3
	3	4	4	4	4
	4	5	5	5	5
	5	6	6	6	6
	6	7	7	7	7
	7	8	8	8	8
	8	9	9	9	9
	9	10	10	10	10
	10	11	11	11	11
Protein	6	7	7	7	7
Hawaii	2	3	2	3	3
Texas	2	3	2	4	2

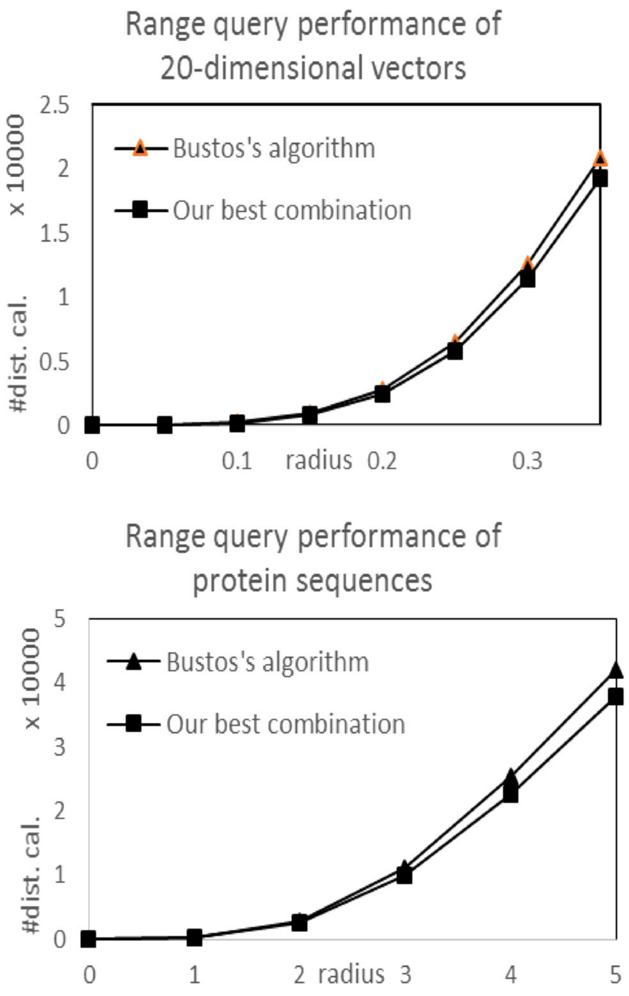


Fig. 7 Comparison of pivot selection algorithms

range query indexing. For other data management of analysis tasks, such as clustering, classification, more specific objective functions can be studied.

The intrinsic dimension of data is commonly accepted as a choice of the number of pivots. We propose a new estimation method that is faster than one of the best existing methods with comparable accuracy.

The pivot space model makes it possible to estimate intrinsic dimension a metric space with dimension reduction method for multi-dimensional space. More work is expected alone this direction. Intrinsic dimensions estimated this way is the dimension of a multi-dimensional space into which a metric space is embedded. Other properties, such as degree of freedom and manifold, of a metric space also deserve investigation.

The incremental sampling pivot selection framework is generalized from Bustos et al.'s algorithm. We proposed new heuristics for candidate set and evaluation set, which are experimentally shown to outperform Bustos et al.'s algorithm. More objective function, candidate set heuristic and evaluation set heuristic can be defined and easily plugged into this framework for the study of pivot selection algorithms.

The pivot space model also makes it possible to exploit dimension reduction for multi-dimensional space for pivot selection for metric space. More work is expected alone this direction, especially non-linear methods.

Further, it is of interesting to see how good the pivot selection algorithms work. That is, how far the objective function values of pivots selected by those algorithms are from that of the brutal force method.

In conclusion, this paper surveys the state of the art of pivot selection, and makes new proposals. More work is expected for pivot selection, an important component to conquer the "variety" challenge of big data problems.

Acknowledgments This work is partially supported by the following Grants: China 863: 2015AA015305; NSF-China: 61170076, U1301252, 61471243; Guangdong Key Laboratory Project: 2012A061400024; NSF-Shenzhen: JCYJ20140418095735561, JCYJ20150731160834611; Shenzhen-Hong Kong Innovation circle Project: SGLH20131010163759789. Dr. Minhua Lu is the corresponding author.

References

1. Mao R, Honglong X, Wenbo W, Li J, Li Y, Minhua L (2015) Overcoming the challenge of variety: big data abstraction, the next evolution of data management for AAL communication systems. IEEE Commun Mag 53(1):42–47
2. Roman S (1992) Advanced linear. Algebra graduate texts in mathematics, vol 135. Springer, Berlin
3. Chavez E, Navarro G, Baeza-Yates R, Marroqu J (2001) Searching in metric spaces. ACM Comput Surv 33(3):273–321
4. Zezula P, Amato G, Dohnal V, Batko M (2006) Similarity search: the metric space approach. Springer, Heidelberg
5. Samet H (2006) Foundations of multidimensional and metric data structures. Morgan-Kaufmann, San Francisco
6. Hjaltason G, Samet H (2003) Index-driven similarity search in metric spaces. ACM Trans Database Syst (TODS) 28(4):517–580
7. Mao R, Miranker W, Miranker DP (2012) Pivot Selection: dimension reduction for distance-based indexing. J Discret Algorithm Elsevier 13:32–46
8. Uhlmann JK (1991) Satisfying general proximity/similarity queries with metric trees. Inf Proc Lett 40(4):175–179
9. Yianilos PN (1993) Data structures and algorithms for nearest neighbor search in general metric spaces. In the fourth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics
10. Bozkaya T, Ozsoyoglu M (1999) Indexing large metric spaces for similarity search queries. ACM Trans Database Syst 24(3):361–404
11. Bustos B, Navarro G, Chavez E (2003) Pivot selection techniques for proximity searching in metric spaces. Pattern Recogn Lett 24(14):2357–2366
12. Clarkson KL (2006) Nearest-neighbor searching and metric space dimensions, In: Nearest-neighbor methods for learning and vision: theory and practice, MIT Press, pp. 15–59
13. Kegl B (2003) Intrinsic dimension estimation using packing numbers. Adv Neural Inf Proc Syst 15:681–688
14. Camastra F (2003) Data dimensionality estimation methods: a survey. Pattern Recogn 36(12):2945–2954
15. Mao R, Xu W, Ramakrishnan S, Nuckolls G, Miranker DP (2005) On optimizing distance-based similarity search for biological databases. In the 2005 IEEE computational systems bioinformatics conference (CSB 2005)
16. Traina C, Jr, Traina A, Faloutsos C (1999) Distance exponent: a new concept for selectivity estimation in metric trees, Technical Report CMU-CS-99-110, Computer Science Department, Carnegie Mellon University
17. Beyer KS, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful? The 7th international conference on database theory. Springer, Berlin
18. Shaft U, Ramakrishnan R (2005) When is nearest neighbors indexable? In the tenth international conference on database theory (ICDT 2005). Springer, Berlin
19. Grassberger P, Procaccia I (1983) Measuring the strangeness of strange attractors. Physica 9D(1–2):189–208
20. Roweis S (1997) EM Algorithms for PCA and SPCA. Neural Inf Proc Syst 10:626–632
21. Brin S (1995) Near neighbor search in large metric spaces. In the 21th international conference on very large data bases (VLDB'95). 1995. Zurich, Switzerland, Morgan Kaufmann Publishers Inc
22. Ciaccia P, Patella M (1997) Bulk loading the M-tree. In 9th Australasian database conference (ADO'98)
23. Navarro G (1999) Searching in metric spaces by spatial approximation. In: Proceedings of the string processing and information retrieval symposium and international workshop on groupware. IEEE Computer Society
24. Gonzalez TF (1985) Clustering to minimize the maximum intercluster distance. Theoret Comput Sci 38:293–306
25. Hochbaum DS, David B (1985) Shmoys, A best possible heuristic for the k-center problem. Math Op Res 10(2):180–184
26. The UMAD project: https://github.com/ruimao/UMAD
27. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453
28. Xu W, Miranker DP (2004) A metric model of amino acid substitution. Bioinformatics 20(8):1214–1221
29. Navarro G (2009) Analyzing metric space indexes: what for? In the proceedings of the second international conference on similarity search and applications (SISAP2009), pp. 3–10
30. Venkateswaran J, Kahveci T, Jermaine CM, Lachwani D (2008) Reference-based indexing for metric spaces with costly distance measures. VLDB J 17(5):1231–1251 **Springer**
31. Celik C (2002) Priority vantage points structures for similarity queries in metric spaces. In: Proceedings of EurAsia-ICT 2002: information and communication technology, ser. LNCS(2510). pp. 256–263. Springer
32. Celik C (2008) Effective use of space for pivot-based metric indexing structures. In: Proceedings of international workshop on similarity search and applications (SISAP'08). IEEE Press, pp. 402–409
33. Micó ML, Oncina J, Vidal E (1994) A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. Pattern Recognition Letters 5(1):9–17
34. Vleugels J, Veltkamp RC (2002) Efficient image retrieval through vantage objects. Pattern Recogn. 35(1):69–80 **Elsevier**
35. Van Leuken RH, Veltkamp RC (2011) Selecting vantage objects for similarity indexing. ACM Trans Multim Comput Commun Appl 7(3):1–18
36. Shapiro M (1977) The choice of reference points in best-match file searching. Commun ACM 20(5):339–343
37. Ramasubramanian V, Paliwal KK (1992) An efficient approximation-elimination algorithm for fast nearest-neighbor search based on a spherical distance coordinate formulation. Pattern Recogn Lett 13(7):471–480
38. Traina C Jr, Filho RF, Traina AJ, Vieira MR, Faloutsos C (2007) The Omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. VLDB J 16(4):483–505
39. Mao R, Xu W, Singh N, Miranker DP (2005) An assessment of a metric space database index to support sequence homology. Int J Artif Intell Tools 14(5):867–885
40. Hennig C, Latecki LJ (2003) The choice of vantage objects for image retrieval. Pattern Recognit 36(9):2187–2196

41. Brisaboa NR, Farina A, Pedreira O, Reyes N (2006) Similarity search using sparse pivots for efficient multimedia information retrieval. In Proceedings of the 8th IEEE international symposium on multimedia (ISM'06). IEEE Press, pp. 881–888
42. Bustos B, Pedreira O, Brisaboa NR (2008) A dynamic pivot selection technique for similarity search in metric spaces. In Proceedings of 1st international workshop on similarity search and applications (SISAP'08). IEEE Press, pp. 105–112
43. Berman A, Shapiro LG (1998) Selecting good keys for triangle-inequality-based pruning algorithms. In: Proceedings of the 1998 international workshop on content-based access of image and video databases (CAIVD '98), pp. 12–19,1998, Bombay, India