Dimension Reduction for Distance-Based Indexing

Rui Mao
Shenzhen University
3688 Nanhai Rd., Office Tower #342
Shenzhen,Guangdong, 518060, China
(+86)755 2655 8644
mao@szu.edu.cn

Willard L. Miranker
Yale University
227 Church Street, PH2E
New Haven, CT 06510, USA
(+1)203 432 7226
miranker@cs.yale.edu

Daniel P. Miranker
University of Texas at Austin
1 University station, C0500
Austin, TX 78712, USA
(+1)512 471 9541
miranker@cs.utexas.edu

ABSTRACT

Distance-based indexing exploits only the triangle inequality to answer similarity queries in metric spaces. Lacking of coordinate structure, mathematical tools in R^n can only be applied indirectly, making it difficult for theoretical study in metric space indexing. Toward solving this problem, we formalize a "pivot space model" where data is mapped from metric space to R^n, preserving all the pair wise distances under L^∞. With this model, it can be shown that the indexing problem in metric space can be equivalently studied in R^n. Further, we show the necessity of dimension reduction for R^n and that the only effective form of dimension reduction is to select existing dimensions, i.e. pivot selection. The coordinate structure of R^n makes the application of many mathematical tools possible. In particular, Principle Component Analysis (PCA) is incorporated into a heuristic method for pivot selection and shown to be effective over a large range of workloads. We also show that PCA can be used to reliably measure the intrinsic dimension of a metric-space.

Categories and Subject Descriptors

H.2.2 [**Database management**] Physical Design – *Access methods*; H.3.1 [**Information storage and retrieval**]: Content analysis and indexing — *indexing methods*; H.3.3 [**Information storage and retrieval**]: Information search and retrieval — *clustering*, *search process*

General Terms

Algorithms.

Keywords

Similarity query, metric space, dimension reduction, intrinsic dimension, pivot selection, pivot space model.

SISAP2010, September 18-19, 2010, Istanbul, Turkey.
Copyright 2010 ACM 978-1-4503-0420-7/10/09 …$5.00.

1. INTRODUCTION

Distance-based indexing [7, 11, 20], also known as metric-space indexing, only requires a metric distance function to answer similarity queries. This, so-called, "black-box" model is advantageous for any application where the data cannot be effectively mapped to feature vectors, enabling a uniform programming model for problems that can be recast into metric spaces. The generality of the approach is also its challenge. What makes distance-based indexing difficult is the lack of coordinate structure. The dimension of the data is not explicit. As a result, mathematical tools developed for R^n are not directly applicable to distance-based problems.

A common method (Section 3) is to first map the metric space to low dimensional R^k, and then to answer the similarity query with geometric methods developed for R^k. To summarize and formalize this method, we propose the *pivot space model*. With this model, there are three steps to answer a similarity query in a metric space. In step 1, the data in the metric space is mapped into a subset of R^k, called the *pivot space*. Then, a region in R^k containing all the query results, named the *query cube*, is determined. In step 2, multi-dimensional methods are applied to retrieve all the points in the query cube. In step 3, all the points retrieved in step 2 are compared with the query object to remove the false positives.

It has been shown that any finite metric space of n points can be mapped isometrically into R^n, one dimension for each point, using the L^∞ norm [16]. In pivot space model we call the image of a dataset so constructed the *complete pivot space*. It will follow that the evaluation of a query in the complete pivot space may be accomplished using multidimensional indexing methods but will require a calculation for each of the n dimensions. Thus, dimension reduction is necessary. Next, we prove that any dimension reduction technique that creates new dimensions will still require a calculation for each of the n dimensions. Therefore, the only effective form of dimension reduction for the complete pivot space is one that selects a subset of the existing dimensions, i.e. pivot selection.

To demonstrate how a general dimension reduction technique can be applied to distance-based problems, we design a pivot selection algorithm based on PCA, a popular dimension reduction technique for multi-dimensional spaces (Section 5). The basic idea is to select existing dimensions that would best approximate the eigenvectors computed by PCA. Results show that our pivot selection heuristic outperforms a deterministic corner-selection and a non-deterministic incremental selection heuristic [5].

Several analytic studies of high dimensional query problems have concluded that the indexability of data is sensitive to its dimension and that the performance of tree-based indexing on fixed size vector data sets degrades to a sequential scan as the dimension of the data increases [2, 21, 23]. Based on the pivot space model, we propose to estimate the intrinsic dimension of a metric space dataset based on the eigenvalues computed by PCA on its complete pivot space. Empirical results show that our method gives more accurate estimates of intrinsic dimension (Section 6).

2. RELATED WORK

The development of distance-based indexing algorithms is commonly decomposed into to two sub-problems, pivot selection and partitioning methods. These are well discussed in a pair of surveys and a text [7, 11, 20].

Only a few methods have been investigated for pivot selection [3-5, 14, 15, 25]. Bustos et al. exploit sampling and select pivots that maximize the mean [5]. Further, they argue that good pivots are usually outliers, but that the reverse is not true. The Metric Tree (M-tree) bulkload algorithm selects pivots randomly but does not use any sampling [8]. SA-tree selects the centers of neighboring cells of a Voronoi diagram as pivots [18].

The farthest-first-traversal (FFT) k-center clustering algorithm is usually used to choose pivots. It is a fast and convenient way to identify corners, or outliers. FFT minimizes the maximum cluster diameter and gives a result at most twice the optimal diameter [12]. Its time and space complexities are both O(n). The use of FFT is based on Yianilos' observations made of uniformly distributed points in the unit square [25]. He proposes to select the corners of the data set as pivots for Vantage Point Tree (VPT). Multi-Vantage Point Tree (MVPT) [3] selects multiple corners.

Brin, in GNAT, suggests index trees be constructed with a variable number of pivots such that a larger number of pivots are used for higher populated clusters to maintain balance [4].

What determines the indexability of data is the intrinsic dimension, which is invariant and independent of data representation. Many authors have tried to define and quantify intrinsic dimension [7, 14]. We consider it to be k, where the data can be embedded into R^k with small distortion.

The following are two existing methods.

Method 1: Chavez et al. [7] define the intrinsic dimension of a metric space as $\rho = \mu^2/2\sigma^2$, where μ and σ^2 are the mean and variance of the pairwise distances.

Method 2: Mao et al. [14] measure how the volume of a hyper-ball, or the number of points in it, changes with respect to the radius. Let r be the range query radius, and n be the average number of range query results. Then, slope coefficient, given by linear regression on the logarithm of n and r, is an estimate of the intrinsic dimension.

Method 1 is simple. Ramakrishnan et al. use similar forms to that of Method 1 to determine the stability of workloads [2, 21]. Method 2 is actually a variation of the Box Dimension [9]. It is limited by values of r and the assumption of uniform distribution.

3. PIVOT SPACE MODEL

In this section, we summarize a generally used metric space indexing approach and propose the pivot space model. A theorem is given showing that the same pivot space can be produced from datasets in a metric space, or in R^n.

3.1 General Steps

In this subsection, let R^n denote a general real coordinate space of dimension n. There are three steps.

Step 1: (1) Map the data into R^n. (2) Map the query object into R^n. (3) Determine a region in R^n that completely covers the range query ball.

Pivot selection maps the data into R^n, i.e. to select particular points in the database as pivots and to represent each point by its distances to the pivots. We name the image of data so obtained the *pivot space*. Determining the distances between a range query object and pivots essentially maps the query into the pivot space.

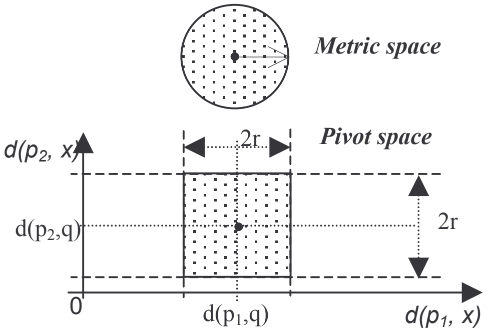


Figure 1. The query ball of a range query in a metric space is covered by a square in the pivot space

The shape of the image of the query ball of a range query (q, r) in a general metric space is not clear in a pivot space. However, it can be proved, from the triangle inequality, that the image of the query ball is completely covered by a hypercube of edge length 2r in the pivot space [7], which is actually a ball of radius r in the new metric space specified by the pivot space and the L^∞ distance. We call the hypercube the *query cube*. Figure 1 shows an example where 2 pivots are selected. All the points in the query ball are mapped into the query square in the 2-d pivot space plane. Points outside of the square can be discarded using the triangle inequality while points within the square cannot be discarded. To answer the query, one first retrieve all the points in the square. Then their distances to q are computed directly to determine whether their pre-images are in the query ball.

Step 2: Exploit multi-dimensional techniques to retrieve all the points in the region determined in Step 1.

The basic idea of Step 2 is divide-and-conquer based on the data coordinates. Many partition methods in multi-dimensional indexing can be applied here. For example M-tree [8, 22] partitions the data in a manner similar to an R-tree [10], while MVPT's partition is similar to a k-d tree [1].

Step 3: For each point retrieved in Step 2, compute its distance to the query object to remove false positives.

In Step 1, the query cube is a superset of the image of the query ball. Therefore, after all the points in the query cube are retrieved, their distances to the query object have to be computed to remove the false positives. Step 1 is the focus in this paper.

3.2 Pivot Space Model

Let (M, d) be a metric space, where M is the space containing the data, and d is a metric distance function. Let $S = \{x_i \mid x_i \in M, i = 1, 2, \ldots, n\}$, be the database, $n \geq 1$. S is a finite indexed subset of M. Duplicates are not allowed.

Let $P = \{p_j \mid j = 1, 2, \ldots, k\}$ be a set of pivots. $P \subseteq S$. Duplicates are not allowed.

Definition 1 Pivot Space, $F_{P,d}(S)$: Given the set of pivots $p_j \in P$, each point in S can be mapped to a non-negative number, which is its distance to p_j. That is, the following mappings can be defined:

$$f_j: M \rightarrow R^+ \bigcup \{0\},\ f_j(x) = d(x, p_j),\ p_j \in P,\ j = 1, \ldots, k.$$

The ordered collection of these k mappings defines a vector valued mapping $F_{P,d}$ on M, which maps a point in M to a point in the non-negative orthant of R^k. The j-th coordinate of the image represents the distance to p_j:

$$F_{P,d}{:}M \rightarrow R^k\!: x_p \equiv F_{P,d}(x) \in F_{P,d}(M),$$

$$x_p=(f_1(x), \ldots, f_k(x))=(d(x, p_1), \ldots, d(x, p_k)),$$

We say x_p, a k-dimensional vector in R^k, is the image of x. The *pivot space* of S is defined as the image set of S under $F_{P,d}$,

$$F_{P,d}(S) = \{x_p \mid x_p = F_{P,d}(x) = (d(x,p_1), \ldots, d(x,p_k)), x \in S\}.$$

Here and in what follows, the superscript p denotes objects in the pivot space. $F_{P,d}(x)$ can be written as F(x, P, d) and $F_{P,d}(S)$ as F(S, P, d) to show that these mappings are specified in term of three parameters. Pivot space shall also refer to R^k where $F_{P,d}(S)$ resides, since confusion will not result.

	p_1^p	p_2^p	…	p_k^p
x_1^p	$d(x_1, p_1)$	$d(x_1, p_2)$	…	$d(x_1, p_k)$
x_2^p	$d(x_2, p_1)$	$d(x_2, p_2)$	…	$d(x_2, p_k)$
…	…	…	…	…
…	…	…	…	…
x_n^p	$d(x_n, p_1)$	$d(x_n, p_2)$	…	$d(x_n, p_k)$

Figure 2. Point-pivot pairwise n×k distance matrix $D_{P,d}(S)$

Definition 2 Point-Pivot Pairwise Distance Matrix $D_{P,d}(S)$: $D_{P,d}(S)$, also written as D(S, P, d), is an n×k matrix (Figure 2), whose (i,j)-th element is the distance from the i-th data point to the j-th pivot. Each row vector (x_i^p) can be regarded as a point in the pivot space specified by all the distances from a database point to each of the pivots, and each column vector (p_j^p) can be regarded as a basis vector in the pivot space specified by the distances from all points to the j-th pivot:

$$D_{P,d}(S) = (x_1^p, x_2^p, \ldots, x_n^p)^T = (p_1^p, p_2^p, \ldots, p_k^p),$$

where the x_i^p s are the row vectors,

$$x_i^p \equiv F_{P,d}(x_i) = (d_{i1}, d_{i2}, \ldots, d_{ik}),$$

and p_j^p is the column vector, $p_j^p \equiv (d_{1j}, d_{2j}, \ldots, d_{nj})^T$. Moreover, $d_{ij} = d(x_i, p_j) \geq 0$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, k$.

Properties of the pivot space and the distance matrix include:

(1) Each point in $F_{P,d}(S)$ is a row in $D_{P,d}(S)$. $F_{P,d}(S)$ and $D_{P,d}(S)$ are representations of one another.

(2) Because different points in S can have the same distance to a pivot, there might be duplicates among the rows of $D_{P,d}(S)$. Therefore, the correspondence between the database points and points in the pivot space $F_{P,d}(S)$, or the rows of $D_{P,d}(S)$, is many-to-one. Similarly, there is a many-to-one correspondence between the pivots and the points in $F_{P,d}(S)$, or the columns of $D_{P,d}(S)$.

The following characterizes the case when all the database points are used as pivots.

Definition 3 Complete Pivot Space F_d^c (S) and Complete Distance Matrix D_d^c (S): F_d^c (S) = $F_{S,d}(S)$ and D_d^c (S) = $D_{S,d}(S)$.

That is, when all points in S are selected as pivots, the complete pivot space of S is the pivot space of S, and the complete distance matrix of S is the distance matrix of S.

The complete pivot space and the complete distance matrix have the following properties.

(1) The dimension of the complete pivot space F_d^c (S) is n.

(2) Because duplicates are not allowed in S, there are no repetitions in the ordered set of distances from one point to all others. Therefore, there are no duplicates in F_d^c (S).

(2.1) Each point in the complete pivot space has exactly one coordinate with value zero, while all other coordinates are positive. That is, all such points reside in the non-negative orthant of R^n.

(2.2) D_d^c (S) is a symmetric n×n matrix, with zeros on the main diagonal and positive entries elsewhere.

(2.3) The correspondence between points in S and points in F_d^c (S) (row vectors of D_d^c (S)) is one-to-one. The correspondence between points in S and the basis vectors in F_d^c (S) (column vectors of D_d^c (S)) is one-to-one.

(3) D_d^c (S) contains all and only the information provided by the distance oracle. Different data sets from various metric spaces can have the same complete pairwise distance matrix.

The following shows the identicalness between pivot spaces.

Theorem 1: Given metric space (M, d), database S and pivot set P, then F(S, P, d) = F(F(S, P, d), F(P, P, d), L^∞), and D(S, P, d) = D(F(S, P, d), F(P, P, d), L^∞).

Proof: It suffices to show: F(S,P,d) = F(F(S,P,d), F(P,P,d), L^∞).

The following 5 relations follow by definition:

$F(S,P,d)=\{x^p \mid x^p=F(x,P,d)=(d(x,p_1), \ldots, d(x,p_k)), x \in S\}$

$F(P,P,d)=\{p^p \mid p^p=F(p,P,d)=(d(p,p_1), \ldots, d(p,p_k)), p \in P\}$

$F(F(S,P,d), F(P,P,d), L^\infty) = \{ F(x^p, F(P,P,d), L^\infty)\}$

$F(x^p, F(P,P,d), L^\infty) = (L^\infty(x^p,p_1^p), \ldots, L^\infty(x^p,p_k^p)), x^p \in F(S,P,d)$

$L^\infty(x^p,p^p) =L^\infty[(d(x,p_1), \ldots,d(x,p_k)), (d(p,p_1),\ldots,d(p,p_k))]$

$\qquad = \max\{ \mid d(x, p_j) - d(p, p_j)\mid, j = 1, 2, \ldots, k \}$

Equality holds in one of the triangle inequalities

$|d(x,p_j) - d(p,p_j)| \leq d(x, p)$, $j = 1, 2, \ldots, k$ since for $p \in P \subseteq S$, there exists a t such that $p = p_t$, $t \in \{1, \ldots, k\}$.

Therefore, we deduce that $L^\infty(x^p, p^p) = d(x, p)$, and

$$F(x^p, F(P, P, d), L^\infty) = (d(x, p_1), \ldots, d(x, p_k)) = x^p.$$

Thus $F(S,P,d)=F(F(S,P,d), F(P,P,d),L^\infty)$, as required. □

In others word, Theorem 1 says that for any pivot space F(S, P, d) generated from a metric space (S, d) and a set P of k pivots, an identical pivot space can be generated from a subset of R^k, i.e. F(S, P, d), with pivots F(P, P, d) and the L^∞ distance. Therefore, when dealing with the pivot space, there is no difference whether the pivot space is created from a metric space dataset S or a real coordinate dataset F(S, P, d). Thus, we can assume that the original data is F(S, P, d), in R^k.

In the complete pivot space, Theorem 1 becomes:

$$F^c_{L^\infty}(F^c_d(S)) = F^c_d(S), \text{ and thus } D^c_{L^\infty}(F^c_d(S)) = D^c_d(S)$$

Note that it states the known fact that any finite metric space (size n) is isometric to a metric space formed by a subset of R^n (R^{n-1} , to be more precise) with the L^∞ distance [16].

In the following, Corollary 1 states that the mapping from the metric space to the complete pivot space can be applied recursively and the result remains invariant. Corollaries 2 and 3 describe the impact of Theorem 1 on distance-based indexing. The proofs are omitted as they are straightforward.

Corollary 1: Let $F^{c\ (1)}_{L^\infty}(F^c_d(S)) = F^c_{L^\infty}(F^c_d(S))$ and

$F^{c\ (n+1)}_{L^\infty}(F^c_d(S)) = F^c_{L^\infty}(F^{c\ (n)}_{L^\infty}(F^c_d(S)))$, n≥1. Then,

$$F^{c\ (n)}_{L^\infty}(F^c_d(S)) = F^c_d(S), \; n\geq 1.$$

Corollary 2: Since the data set S from any general metric space can be mapped isometrically into the complete pivot space with the L^∞ metric without loss of any distance information, instead of indexing S in a black-box metric space, it is equivalent to index $F^c_d(S)$ in the much more palpable vector space R^n.

Corollary 3: The intrinsic dimensions of S and $F^c_d(S)$ equal.

This is because that the intrinsic dimensions are specified by the pairwise distances and the pairwise distance matrices of S and $F^c_d(S)$ are the same.

4. DIMENSION REDUCTION FOR DISTANCE-BASED INDEXING

This section is dedicated to the problem "how to map a metric space to R^k"? We study the question from the perspective of dimension reduction. The discussion starts from the complete pivot space, the most straightforward case and the case with all the information. Issues under discussion include "can we evaluate similarity queries in the complete pivot space directly?", "how to perform dimension reduction for the complete pivot space?", "why is pivot selection important?", and "how to select pivots?"

4.1 Evaluating Similarity Queries in the Complete Pivot Space Directly

Theorem 2 answers the question "can we evaluate similarity queries in the complete pivot space directly?" It is straightforward and helps to understand Theorem 3.

Theorem 2: Evaluation of similarity queries in the complete pivot space degrades the query performance to linear scan.

Proof: Given a similarity query object, the computation of its coordinates in all dimensions of the complete pivot space is already a linear scan of the database. □

Therefore, to answer the similarity query posed in the metric space in R^n, dimension reduction in the complete pivot space is inevitable.

4.2 Pivot Selection: the Only Form of Effective Dimension Reduction

Theorem 2 establishes the necessity of dimension reduction in the complete pivot space. The underlying reason is the lack of uniform coordinate structure in general metric spaces. In the following, we discuss the type of dimension reduction that can be applied to the complete pivot space. A dimension reduction technique is termed "effective" if evaluation of similarity queries in the space generated by this technique does not degrade to a linear scan. Not all the dimension reduction for multi-dimensional indexing is effective for distance-based indexing.

Theorem 3: If a dimension reduction technique creates new dimensions based on all existing dimensions, evaluation of similarity queries in the space generated by this technique degrades to a linear scan.

Proof: Let $S = \{x_i \mid i = 1, 2, \ldots, n\}$, be the database, and let d be the distance oracle. Let y be an arbitrary point in the metric space, and let its coordinates in the complete pivot space $F^c_d(S)$ be $y^{(1)}, y^{(2)}, \ldots, y^{(n)}$. Then, $y^{(i)} = d(y, x_i)$, i = 1, 2, …, n. Let a new coordinate $y^{(n+1)} = G(y^{(1)}, y^{(2)}, \ldots, y^{(n)})$, be specified by some unspecified function G. For example, G might be a linear combination of its variables. Given a query object q, to compute coordinate $q^{(n+1)}$, coordinates $q^{(1)}, q^{(2)}, \ldots, q^{(n)}$ in the complete pivot space $F^c_d(S)$ have to be computed first. This already represents a linear scan of the database. □

Although the power of dimension reduction methods in multi-dimensional indexing is clear, the cost to set up the coordinate structure based on new dimensions is excessive. Therefore, dimension reduction for the complete pivot space should only select existing dimensions.

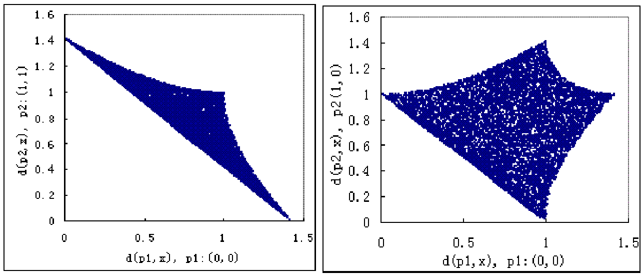
Therefore, pivot selection is the only effective form of dimension reduction for the complete pivot space. It is the only effective form of mapping from a metric space to R^k.

4.3 The Importance of Pivot Selection

To date, the major attention in this area has been on the partition problem in Step 2 (Section 3.1). The importance of pivot selection is not fully recognized, and most methods just try to mimic multi-dimensional indexing in metric spaces.

Pivot selection can be viewed as a process of information loss. The information available to Step 2 is limited by pivot selection. Pivot selection impacts the distribution of data in the pivot space, which is critical to the search performance of Step 2. Moreover, in a pivot space produced from a "better" set of pivots, data are typically more distinguishable. Thus, false positives are reduced and Step 3 is faster.

An example is to select two pivots for points randomly and uniformly sampled from the unit square with the Euclidean norm. The common method is to use the FFT algorithm to select the two farthest opposite corners of the data [12]. For uniform vector data, these two corners are close to the two ends of the data's first principal component (with the largest eigenvalue). The distances from a point to the two farthest opposite points largely correlate with each other. If a point is close to the first pivot, usually it is far from the second pivot. As a result, if two points are not distinguishable by the first pivot (their distances to the first pivot are similar), they are not likely to be particularly distinguishable by the second pivot. Therefore, selecting these two pivots does not provide much more information than selecting just one pivot. An alternative is to select two corners on the 1st and 2nd principal components. Since the principal components are orthogonal to each other, the two pivots in this case define two less correlated dimensions. If two points are not distinguishable by one pivot, they can still be distinguished by the other pivot.



a) Pivots: opposite corners (0,0) and (1,1)

b) Pivots: neighboring corners (0,0), (1,0)

Figure 3 Pivot spaces (2 pivots) of points randomly sampled from the unit square, with different selection of

Figure 3 illustrates the case just discussed. The two pivots are opposite corners (0,0) and (1,1) in a) and are neighboring corners (0,0) and (1,0) for b). We can see that the pivot space in b) spreads out more widely (less density) than that in a). Therefore, data in pivot space in b) are more distinguishable. Although pivots are always corners, different corners can still have much different impact on query performance. This supports Bustos et al.'s observation that outliers might not be good pivots [5].

4.4 Adapt Dimension Reduction to Pivot Selection

We propose a heuristic to adapt general dimension reduction methods that create new dimensions to distance-based problems. The basic idea is to select existing dimensions that best approximate the new dimensions created by dimension reduction.

One way is to select the existing point with the maximum correlation, or the minimum angle, with the new dimensions created by dimension reduction. Bustos et al. [5] suggest that a good choice of pivots should maximize the mean of the pairwise distances in the pivot space. To increase the mean of the pairwise distance also, we consider the covariance instead of the correlation. That is, for each new dimension, select the point with the largest projection on that new dimension in the pivot space.

With this heuristic, a general dimension reduction technique can be adapted to pivot selection in two steps. First, run the dimension reduction on the complete pivot space to create the new dimensions. Second, select the pivots with largest projections on the new dimensions.

We show how to adapt PCA to pivot selection with this heuristic in the next section.

5. PCA IN DISTANCE-BASED INDEXING

In this section, we apply PCA to pivot selection and to the estimation of intrinsic dimension.

5.1 PCA for Pivot Selection

In Step 3 (see in Section 3.1), points of the query cube in the pivot space need to be checked by computing the distance with the query object directly. Therefore, it is of key importance to reduce the number of points in the query cube. We aim to maximize the variance of the data along the dimensions of the pivot space, which means the data points are more distinguishable among each other. Because PCA considers the distribution of the whole data set and maximizes the variance of the data along each principal component, we choose it for pivot selection.

A difficulty with PCA is the computational cost. Even a fast approximation usually takes O(n^2) time, which is too expensive for large databases even if it is computed off-line. As Bustos et al. [5] point out, although good pivots are usually outliers, outliers are not always good pivots. The set of outliers forms a good candidate set for good pivots. PCA can be conducted on the candidate set. In other words, PCA is not performed on the complete pivot space, but on a pivot space with outliers as pivots. Since the size of the set of outliers is much smaller than the database, the computational cost of PCA is decreased.

```
PivotSelection ( (S, d):data set, k: number of pivot, c: constant)
{    //run FFT to create a candidate set of size k*c
1.   candidate = FFT(S, k*c);
     //generate the pivot space with candidate as the pivot set
2.   PS = F(S, candidate, d);
     //run PCA on PS
3.   PCSET = EMPCA(PS, k);
     // for each PC, find the point with the largest project on it
     Pivots = {};
4.   for each PC ∈ PCSET
         Pivots = Pivots U $\mathrm{argmax}_x$( Proj(PC,x) );
     return Pivots;    }
```

Figure 4. Algorithm for pivot selection

Our pivot selection algorithm is shown in Figure 4. In step 1, a number of outliers of the data are found by FFT and taken to form a candidate set of pivots. Empirical results show that a good choice of the constant c is approximately 30. In step 2, the distances between the corners and the database points are computed to form the distance matrix of the candidate outlier pivot space, on which PCA is performed in step 3. The PCA algorithm applied is EMPCA [19], which can compute only a given number of principal components with the largest eigenvalues. Finally in step 4, for each principal component, the data point whose image in the pivot space has the largest projection on that principal component is selected as a pivot. Since k and ck is much smaller than the database size, the time complexity is O(n). The algorithm is compared with FFT and Bustos et al.'s incremental sampling selection method [5]. Results show that the overall performance of our algorithm surpasses the other two.

5.2 Estimating the Intrinsic Dimension

According to previous work [16] and the pivot space model, the complete pivot space together with the L^∞ distance is isometric to the original metric space. Therefore, methods to estimate the intrinsic dimension of R^n [6, 13] are now applicable to a metric space. We introduce a third method to estimate the intrinsic dimension based on the relative change of eigenvalues of PCA in the complete pivot space, equivalently, of the complete distance matrix D_d^c(S).

Method 3: Let Q = {q_1, q_2, …, q_n} be the principal components (PCs) of the data in the complete pivot space. Let $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be the variances (eigenvalues) corresponding to each PC, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$. Note that the λ_i are normalized so that they sum to 1. The intrinsic dimension is estimated as (1) $\hat{d}$ = argmax$_i$ (λ_i / λ_{i+1}), i = 1, …, n-1, and (2) $\sum_{j=1}^{i} \lambda_j > 0.6$, and (3) $0.015 \leq \lambda_{i+1} \leq 0.035$.

Condition (1) indicates that the eigenvalues decreases relatively the most from λ_i to λ_{i+1}. Condition (2) guarantees that at least 60% of the variance of the data is maintained if they are reduced to dimension $\hat{d}$. Condition (3) ensures that principal components with tiny (<1.5%) variance (eigenvalue) will be excluded and principal components with larger (>3.5%) variance will be included.

The three methods are evaluated using a suite of workloads. The results in Section 6 show that all three methods are asymptotically correct, while Method 3 gives quantitatively more accurate estimates for data whose intrinsic dimension are known.

6. EMPIRICAL RESULTS

6.1 Organization of Empirical Study

The test suite consists of synthetic vector data, biological data, real vector data and an image dataset [17]. The synthetic vector consists of data of uniform, exponential and normal distributions. Different dimensions have independent identical distributions. Three types of biological data are considered: (1) the amino-acid sequence fragments of the yeast proteome with weighted-edit distance based on the metric PAM substitution matrix [24], (2) the DNA sequence fragments of the Arabidopsis genomes with Hamming distance, and (3) analytically determined peptide fragmentation spectra of human and E. coli proteins with a pseudo-semi-metric cosine distance. The real vector data consists of the US cartographic boundary data of Texas and Hawaii. The image dataset consists of images represented by 66 dimensional feature vectors with a linear combination of L^1 and L^2 norms. The suite is summarized in Table 1.

The index method used in this study is MVPT [3]. The partition algorithm is clustering partition [14]. The sizes of the databases are all 100k, except for those small workloads for which only limited amounts of data is available. The number of pivots is 2 for Texas and mass-spectra data, 4 for DNA and protein data, and 3 for all others. Based on each pivot, 3 partitions are generated. The maximum number of data points in each index leaf node is 100. Since distance evaluation in a metric space is usually costly, we use the average number of distance calculations, which is implementation independent, to answer 5000 range queries as the performance measure of each index. The queries are chosen sequentially from the beginning of the dataset files of each workload. The radii of range queries are chosen so that approximately 0.01% of the databases are retrieved.

Table 1. Summary of test suite

Workload	Total size	Distance oracle	Domain dimension
Vector (uniform)	1M	L^1, L^2, L^∞ norm	1-20
Vector(exponential)	100k		1-10
Vector(normal)	100k		1-10
Texas	190k		2
Hawaii	9k		2
Mass-spectra	<90k	Fuzzy cosine distance	40,000
protein	100k	Weighted edit distance	6-18
DNA	<256k	Hamming distance	9-18
Image	10221	L-norms	66

In the following, the query performance of PCA-based pivot selection is compared with FFT and incremental pivot selection. Then, we show the results on the intrinsic dimension of the test suite using the 3 methods in Section 6.3.

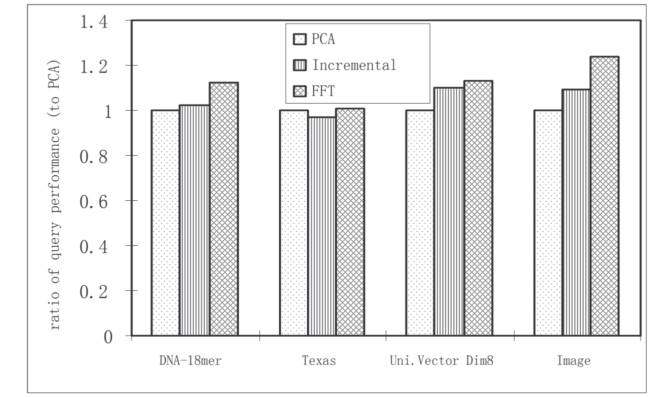


Figure 5. Query cost of three pivot selection methods

6.2 Comparison of pivot selection heuristics

We compare the three pivot selection heuristics, i.e. FFT, Bustos et al.'s incremental selection method [5], and PCA-base method. Indices are built with different heuristics and their query performances are compared.

To be fair, we make sure the index construction costs (measured by number of distance computations to build the indices) of the PCA-based method and the incremental method similar. There are two parameters, A and N, for the incremental method. According to Bustos et al. [5], we try to use a large value for A and a small value for N (Table 2). Moreover, since it is hard to make the two methods have exactly the same construction cost, we always allow more construction cost for incremental method (Table 2).

The query performances of the three methods are also shown in Figure 5 (normalized by the value of the PCA method). Both Figure 5 and Table 2 show that FFT always yields the worst performance. For most of the workloads, the PCA based method yields the best query performance. The only case where the incremental method yields the best performance is for Texas data, where all the three methods do not differ much. We believe this is because Texas data are of low intrinsic dimension and so are easy to index.

Table 2. Comparison of pivot selection heuristics, measured by query cost per the number of distance calculations

Dataset	Radius	Selectivity	FFT	Incremental				PCA	
			query	build	query	A	N	build	query
DNA 18-mer fragments		0.017%	68357.2	60.9M	62417.7	9k	10	57.5M	60968.1
Texas	0.015	0.011%	27.7	66.6M	26.7	30k	10	58.8M	27.5
Uniform Vector dimension 8	0.65	0.015%	4952.5	66.3M	4822.6	30k	10	58.5M	4394.4
image	0.08	0.13%	1371.4	5.4M	1209.9	5k	15	4.6M	1106.1

Table 3. Estimates of intrinsic dimension of three methods

Workload	Domain dimension	Distance oracle	Intrinsic dimension		
			$\mu^2/2\sigma^2$	regression	$\mathrm{argmax}_i(\lambda_i/\lambda_{i+1})$
Vector (uniform)	D=1-20	L^∞	1.72d - 1.81	0.73d + 0.88	d+1 (d≠3,4), 4, 7 (d=3, 4)
		L^1	d	0.75d + 0.84	d+1
		L^2	1.41d - 0.71	0.78d - 0.72	d+1
Vector (exponential)	D = 1-10	L^∞	0.244d + 0.446	0.676d + 0.62	d+1
		L^1	0.499d - 0.0006	0.737d + 0.482	d+1
		L^2	0.427d + 0.113	0.72d + 0.534	d+1
Vector (normal)	D = 1-10	L^∞	0.644d + 0.559	0.858d + 0.325	d+1
		L^1	0.875d + 0.002	0.863d + 0.32	d+1
		L^2	0.989d - 0.145	0.872d + 0.305	d+1
Texas	2	L^∞ / L^1 / L^2	1.29 / 1.42 / 0.87	1.54 / 1.54 / 1.51	3
Hawaii	2	L^∞ / L^1 / L^2	0.31 / 0.26 / 0.36	1.47 / 1.45 / 1.44	2
Protein q-gram	q = 6-18	Weighted edit distance	2.46q + 2.32	-0.08q + 4.16	q+1 (q<18), 17 (q=18)
DNA q-gram	q = 9-18	Hamming distance	1.27q + 0.37	0.14q + 2.52	q+1 (q<18), 21 (q=18)
Mass-spectra	40,000	Fuzzy cosine distance	0.62	1.23	2
Image	66	Linear combination of L-norms	5.26	4.85	5

6.3 Estimate of intrinsic dimension

Finally, we show results of estimated intrinsic dimension on all the workloads with the three methods. Wherever possible, we vary the domain dimension to see how the estimates change. If a linear relationship is observed, we compute the slope and intercept of the relationship by linear regression. For vector data, we use L^1 and L^∞ norms in addition to the L^2 norm. The estimates are shown in Table 3. First, we can see that as domain dimensions increase, estimates given by all the three methods increase linearly (one exception is the regression method and protein data). Therefore, we think all the three methods are asymptotically accurate. Methods 1 and 2 can be adjusted by constant factors to give accurate estimations for particular data and metric. However, they have different values for slope and intercept for different data and metrics. No generally applicable relationship can be drawn from Methods 1 and 2. Method 3, although some constants are determined empirically, always gives d+1 (q+1) as the estimate except a few cases. Thus, Method 3 is more consistent and stable.

Moreover, Method 3 almost always yields d+1 for the vector data of dimension d, no matter what is the probability distribution of the data and the metric. This is consistent with the observation that if the distances from an unknown vector to d + 1 vectors are known, then the coordinates of the unknown vector can be computed accurately. The three methods are consistent with each other for image data.

7. CONCLUSIONS AND FUTURE WORK

The "power" of metric-space indexing is the generality of the abstraction and encapsulation; just provide a distance function. But, we (the community) have been powerless to do anything other than apply what we can make work with respect to R^n. To date, this has been done opportunistically. The theorems in this paper reveal that within certain boundaries we can in fact be methodical about this. We now even understand, for example, how to exploit PCA in this context. No surprise, the heuristic PCA inspired methods outperform heuristics inspired by ad-hoc observations.

We believe the objective function of incremental sampling is worthy but the associated algorithm is computationally expensive. It would be interesting to see if additional iterations of incremental sampling, despite its cost, can achieve query performance similar to our PCA method. Further, incremental sampling might be made more efficient by sampling from a set of corners instead of the whole dataset. There are other dimension reduction methods for a vector space with annealing characteristics. We anticipate more research along this direction.

By virtue of considering the complete pivot space, our work has drawn a direct parallel between distance-based indexing and high-dimensional indexing. Both have to do dimension reduction and remove false positives. One may project a finite metric space to a reduced dimension coordinate system, but no proper subset of the pivots can faithfully recreate the geometry of the space. Thus, pivot selection results in information loss, typical of dimension reduction techniques. Dimension reduction is integral to indexing both finite metric-spaces and high-dimensional data.

We show that the intrinsic dimension of data in a metric space can be estimated by the intrinsic dimension of the complete pivot space. Thus, methods for vector spaces are now applicable to metric spaces. We have demonstrated how a PCA method can be applied to estimate the intrinsic dimension of the metric space. More work on this is expected.

Another interesting piece of future work is to determine the optimal number of pivots, and its relationship with the intrinsic dimension.

8. ACKNOWLEDGMENTS

We sincerely thank Gonzalo Navarro, Glen Nuckolls, and Piotr Indyk for their comments and suggestions.

This research was supported by a grant from the US National Science Foundation, DBI-0640923, a J. Tinsley Oden Faculty Research Fellowship at the Institute for Computational Engineering and Science, the University of Texas at Austin, USA, and the Initiating Fund for Member of the Chinese Academy of Sciences from Shenzhen University, China.

9. REFERENCES

[1] Bentley, J.L., *Multidimensional binary search trees used for associative searching.* Commun.ACM, 1975. **18**(9): p. 509-517.

[2] Beyer, K.S., J. Goldstein, R. Ramakrishnan, and U. Shaft. *When Is "Nearest Neighbor" Meaningful? the 7th International Conference on Database Theory*. 1999: Springer-Verlag.

[3] Bozkaya, T. and M. Ozsoyoglu, *Indexing large metric spaces for similarity search queries.* ACM Trans. Database Syst., 1999. **24**(3): p. 361-404.

[4] Brin, S. *Near Neighbor Search in Large Metric Spaces*. in *the 21th International Conference on Very Large Data Bases (VLDB'95)*. 1995: Morgan Kaufmann Publishers Inc.

[5] Bustos, B., G. Navarro, and E. Chavez, *Pivot selection techniques for proximity searching in metric spaces.* Pattern Recogn. Lett., 2003. **24**(14): p. 2357-2366.

[6] Camastra, F., *Data dimensionality estimation methods: a survey.* Pattern Recognition, 2003. **36**(12): p. 2945-2954.

[7] Chavez, E., G. Navarro, R. Baeza-Yates, and J. Marroqu, *Searching in metric spaces.* ACM Computing Surveys, 2001. **33**(3): p. 273-321.

[8] Ciaccia, P. and M. Patella. *Bulk loading the M-tree*. in *9th Australasian Database Conference (ADO'98)*. 1998.

[9] Clarkson K.L., Nearest-neighbor searching and metric space dimensions, In: Nearest-Neighbor Methods for Learning and Vision: Theory and Practice, MIT Press, 2006, pp. 15--59

[10] Guttman, A., *R-trees: a dynamic index structure for spatial searching*, in *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*. 1984.

[11] Hjaltason, G.R. and H. Samet, *Index-driven similarity search in metric spaces.* ACM Transactions on Database Systems (TODS), 2003. **28**(4): p. 517-580.

[12] Hochbaum, D.S. and D.B. Shmoys, *A best possible heuristic for the k-center problem.* Mathematics of Operational Research, 1985. **10**(2): p. 180-184.

[13] Kegl, B., *Intrinsic dimension estimation using packing numbers.* Advances in Neural Information Processing Systems, 2003. **15**: p. 681-688.

[14] Mao, R., W. Xu, S. Ramakrishnan, G. Nuckolls, and D.P. Miranker. *On Optimizing Distance-Based Similarity Search for Biological Databases*. in *the 2005 IEEE Computational Systems Bioinformatics Conference (CSB 2005)*. 2005.

[15] Mao, R., W. Xu, N. Singh, and D.P. Miranker, *An Assessment of a Metric Space Database Index to Support Sequence Homology.* International Journal on Artificial Intelligence Tools (IJAIT), 2005: p. 867-885.

[16] Matousek, J., *Lectures on Discrete Geometry*. 2002: Springer-Verlag New York, Inc. 497.

[17] Test suite. http://aug.csres.utexas.edu/mobios-workload/.

[18] Navarro, G. *Searching in Metric Spaces by Spatial Approximation*. in *Proceedings of the String Processing and Information Retrieval Symposium & International Workshop on Groupware*. 1999: IEEE Computer Society.

[19] Roweis, S., *EM Algorithms for PCA and SPCA.* Neural Information Processing Systems 10 , 1997: p. 626-632.

[20] Samet, H., *Foundations of Multidimensional and Metric Data Structures*. 2006, Morgan-Kaufmann.

[21] Shaft, U. and R. Ramakrishnan. *When Is Nearest Neighbors Indexable?* in *Tenth International Conference on Database Theory (ICDT 2005)*. 2005: Springer

[22] Uhlmann, J.K., *Satisfying General Proximity/Similarity Queries with Metric Trees.* Information Processing Letter, 1991. **40**(4): p. 175-179.

[23] Weber, R., H.J. Schek, and S. Blott. *A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces*. in *International Conference on Very Large Data Bases*. 1998.

[24] Xu, W. and D.P. Miranker, *A Metric Model of Amino Acid Substitution.* Bioinformatics, 2004. **20**(8): p. 1214-1221.

[25] Yianilos, P.N. *Data structures and algorithms for nearest neighbor search in general metric spaces*. in *the fourth annual ACM-SIAM Symposium on Discrete algorithms*. 1993: Society for Industrial and Applied Mathematics.