MetricMap: An Embedding Technique for Processing Distance-Based Queries in Metric Spaces

Jason T. L. Wang, *Member, IEEE*, Xiong Wang, Dennis Shasha, and Kaizhong Zhang

***Abstract*—In this paper, we present an embedding technique, called *MetricMap*, which is capable of estimating distances in a pseudometric space. Given a database of objects and a distance function for the objects, which is a pseudometric, we map the objects to vectors in a pseudo-Euclidean space with a reasonably low dimension while preserving the distance between two objects approximately. Such an embedding technique can be used as an approximate oracle to process a broad class of distance-based queries. It is also adaptable to data mining applications such as data clustering and classification. We present the theory underlying *MetricMap* and conduct experiments to compare *MetricMap* with other methods including MVP-tree and M-tree in processing the distance-based queries. Experimental results on both protein and RNA data show the good performance and the superiority of *MetricMap* over the other methods.**

***Index Terms*—Bioinformatics, data mining, embedding method, metric space, nearest neighbors, similarity search.**

I. Introduction

ONE common operation in information retrieval (IR), data mining (DM), and pattern recognition (PR) is similarity search [1], [2]. Given a database of objects $\mathcal{D}$ and a query object Q, the problem of similarity search is to find the objects in $\mathcal{D}$ that are similar to Q. The "similarity" here is measured by a distance function d. In the past, two types of similarity search (or distance-based queries) have been studied: 1) the nearest-neighbor query, which is to locate the objects in $\mathcal{D}$ that are most similar (or closest) to Q and 2) the ϵ-range query, which is to locate the objects in $\mathcal{D}$ whose distances to Q are less than or equal to a user-determined number, ϵ. When the distance function d is simply a metric, several methods, including MVP-tree [3] and M-tree [4], have been proposed to accelerate the searching. In this paper, we collectively refer to these methods as *distance-based data structures*.

The two types of distance-based queries described above are useful in many applications. For example, one widely studied DM application is data clustering. In the agglomerative, hierarchical clustering method, one treats each object as a cluster, and merges those clusters that are close to each other to form larger clusters [2]. In measuring the distance between two clusters, one considers and weighs the distances among the component objects in the clusters. Calculating the distance between two component objects (e.g., RNA secondary structures) is time consuming and often quadratic in the size of the objects in new-generation database systems designed for multimedia and scientific domains [5], rendering online distance calculations prohibitive. The distance-based data structures including MVP-tree and M-tree can be used as a tool to speed up the clustering in these applications.

A. The *FastMap* Algorithm

Another approach for speeding up the clustering is to embed objects in a high-dimensional space (Euclidean or pseudo-Euclidean), R^k, into a low-dimensional target space, $R^n, n \leq k$, in which distance calculations are cheap and then cluster the objects in that low-dimensional space. Such an embedding technique with applications to similarity search and data mining was first proposed by Faloutsos and Lin [6] and their technique was called *FastMap*. The basic idea of *FastMap* is to project objects on a line (O_a, O_b) in R^k, where the line is formed by two *pivot objects* O_a, O_b, which are chosen as follows. First, arbitrarily choose one object and let it be the second pivot object O_b. Let O_a be the object that is farthest apart from O_b. Then, update O_b to be the object that is farthest apart from O_a. The two resulting objects O_a, O_b are pivots.

Consider an object O_i and the triangle formed by O_i, O_a, and O_b (Fig. 1). From the cosine law, one obtains

$$d_{b,i}^2 = d_{a,i}^2 + d_{a,b}^2 - 2x_i d_{a,b}. \tag{1}$$

Thus, the first coordinate x_i of object O_i with respect to the line (O_a, O_b) is

$$x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{2d_{a,b}}. \tag{2}$$

FastMap extends the above projection method to embed objects (data points) of R^k into the target space R^n as follows. Consider a $(k-1)$-dimensional hyperplane $\mathcal{H}$ that is perpendicular to the line (O_a, O_b), where O_a and O_b are two pivot objects. The *FastMap* algorithm then projects all objects in a given database onto this hyperplane. Let O_i, O_j be two objects and let O_i', O_j' be their projections on the hyperplane $\mathcal{H}$. It can be shown [6] that the dissimilarity d' between $O_i', O_j', i, j = 0, \ldots, N-1$, is

$$(d'(O_i', O_j'))^2 = (d(O_i, O_j))^2 - (x_i - x_j)^2 \tag{3}$$

Manuscript received January 16, 2004; revised November 20, 2004. This work was supported in part by the National Science Foundation (NSF) under Grants IIS-9988345, IIS-9988636, MCB-0209754, NIH Grant GM32877, and by the Natural Sciences and Engineering Research Council of Canada under Grant OGP0046373. Approved by Associate Editor W. Pedrycz.

J. T. L. Wang is with the Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102 USA (e-mail: wangj@njit.edu).

X. Wang is with the Department of Computer Science, California State University at Fullerton, Fullerton, CA 92834 USA.

D. Shasha is with the Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 USA.

K. Zhang is with the Department of Computer Science, University of Western Ontario, London, ON N6A 5B7, Canada.

Digital Object Identifier 10.1109/TSMCB.2005.848489

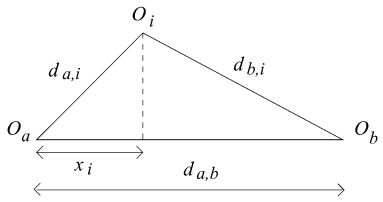

Fig. 1. Illustration of the projection method used in *FastMap*.

where N is the number of objects in the database. Being able to compute d' allows one to project on a second line, lying on the hyperplane $\mathcal{H}$, and therefore orthogonal to the first line (O_a, O_b). The *FastMap* algorithm repeats the above steps recursively, $k-n$ times, thus mapping all objects in R^k to points in R^n.

Previously we introduced *MetricMap* [7] and compared the performance of *FastMap* and *MetricMap* in data mining and clustering applications [8], [9]. The major difference between *FastMap* and *MetricMap* lies in the target space they choose—the former uses Euclidean space while the latter uses pseudo-Euclidean space. Although both *FastMap* and *MetricMap* employ the cosine law, this major difference leads to a new embedding method adopted by *MetricMap*, which is totally different from that used in *FastMap*.

In this paper, we extend the work in [7] by providing detailed proofs, examples and illustrations to explain the theorems given in [7]. Furthermore, we present new results by applying *MetricMap* to processing the nearest-neighbor query and ϵ-range query, and comparing *MetricMap* with related distance-based data structures including MVP-tree [3] and M-Tree [4]. To accelerate query processing, we generalize a previously proposed VA-file technique [10] to the pseudo-Euclidean space for pruning the search space. These new results provide insights to the behavior and applications of *MetricMap*, which were not reported in [7].

II. Related Work

A. Distance-Based Data Structures

Distance-based data structures have been studied by several researchers [1], [3], [4], [11]. In [12], Yianilos proposed the vantage-point tree (VP-tree), which partitions the search space according to the relative distances between the data objects and a specific object, called the vantage point. By considering the relative distances as opposed to the absolute coordinates of the objects, VP-tree avoids the dimensional curse problem [10]. Each node in a VP-tree is connected to a vantage point and a distance value. In the binary tree case, after picking up a vantage point v, the median distance r is calculated, so that the number of data objects within distance r of v is the same as the number of data objects outside distance r of v. This way, the data objects are partitioned into two halves. A node n is created, which is connected to v and r. Two vantage points v_1 and v_2 are then chosen in the two halves, and the median distances r_1 and r_2 are calculated, respectively. The nodes connected to v_1, r_1 and v_2, r_2, respectively, become the children of n. This process is applied recursively until a certain number of vantage points are obtained. To process the ϵ-range query, starting at the root, one calculates the distances between the query object Q and the vantage points and prunes the search space by using the triangle inequality. The search descends the branches of the VP-tree until no further pruning is possible. One then calculates the distances between Q and the remaining data objects to find those that are within distance ϵ of Q.

In a subsequent paper, Chiueh [11] applied VP-tree to content based image retrieval in multimedia databases. Bozkaya and Ozsoyoglu [3] extended the approach to include multiple vantage points, ending up with MVP-tree. An MVP-tree has two vantage points in each node and utilizes the pre-computed distances between the vantage points and data objects in processing the ϵ-range query. These distances are calculated during the construction of the MVP-tree.

Ciaccia *et al.* [4] introduced another closely related data structure, called M-tree, which stores subsets of the data objects into fixed-size leaf nodes. Each internal node of an M-tree has a routing object O_r, a covering radius $r(N_r)$ for every child node N_r, and a pointer to that child node $\mathrm{ptr}(N_r)$. The basic property of the covering radius is that for every object O_i in the subtree rooted at $N_r, d(O_r, O_i) \leq r(N_r)$. Thus, the M-tree algorithm basically partitions the data objects into a set of possibly overlapping "balls". Going up the tree, the balls become larger and larger until the root, whose subtrees cover the whole database. In processing the ϵ-range query, the M-tree algorithm prunes the search space using the triangle inequality in a way similar to the VP-tree algorithm.

In [13], Hjaltason and Samet described a general, incremental nearest-neighbor algorithm that is applicable to a large class of hierarchical spatial data structures. The authors proved informally that, at any step in the execution, the incremental nearest-neighbor algorithm is optimal with respect to the spatial data structure employed. Though not directly related to our techniques, the authors presented a useful search framework with applications in spatial and geographic information systems.

B. Embedding Methods

Another line of works related to our work are embedding techniques. Roweis and Saul [14] introduced an embedding approach, called locally linear embedding, to reduce the dimensionality of high dimensional data. In [15], Donoho and Grimes extended the idea of Roweis and Saul to a significantly broader class of applications. Another algorithm that used weights and regression mapping was presented in [16], which attempts to reduce the squared error introduced by embeddings. Belkin and Niyogi [17] proposed an approach that utilizes the Laplacian operator in an attempt to capture the intrinsic geometric structure of the space under consideration. Their algorithm solves a sparse eigenvalue problem. It is efficient due to its simplicity. A problem associated with using eigenvalues in non-Euclidean vector spaces is that some eigenvalues may be negative. Roth *et al.* [18] introduced a framework in which they adjust the pairwise distances of the vectors in consideration so that the negative eigenvalues can be avoided and the resulting vectors approximate the original vectors satisfactorily.

In [19], Agrafiotis proposed a stochastic process to fine tune an embedding. Courrieu [20] presented a method that uses multidimensional scaling to embed a metric or nonmetric

topological space into a Euclidean space. The algorithm constructs a monotonic embedding. Another approach that extends multidimensional scaling was proposed in [21]. This algorithm differs from the aforementioned locally linear embedding in that the discrepancy in distributional information is used to guide embedding. In [22], Athitsos *et al.* employed a machine learning approach for embedding. Their approach first constructs one-dimensional (1-D) classifiers and uses them to compose a multidimensional classifier. The 1-D classifiers are built in a similar way as *FastMap* calculates the first coordinate of objects described in Section I.A. A training algorithm is then used to choose classifiers that better complement each other to build the multidimensional classifier. In [23], Dubnov and co-authors proposed an algorithm that employs a two-step transformation on a proximity matrix to build a hierarchical cluster. The first step of the transformation represents each data point by its relation to all other data points. The second step re-estimates the pairwise distances between the data points using a statistically motivated proximity measure on these representations. It is worth noting that none of the embedding techniques considers pseudo-Euclidean spaces, as adopted in *MetricMap*. Furthermore, while many of these embedding techniques are useful for data mining applications, they are not designed for processing distance-based queries as addressed in this paper.

The rest of the paper is organized as follows. Section III presents the theory underlying *MetricMap*. Like the distance-based data structures surveyed in Section II-A, the *MetricMap* algorithm proceeds in two phases. In the first phase, which is the embedding phase, the algorithm maps database objects to vectors in a target space. As shown in [9], this phase requires $O(Nm)$ time where N is the number of objects in the database and m is the dimensionality of the target space. In the second phase, which is the on-line search phase, the query object Q is given and the algorithm finds the near(est) neighbors of Q from the database. Section IV describes our techniques for finding the near(est) neighbors of Q using *MetricMap* and a modified VA-file technique [10]. Section V compares the cost of *MetricMap*, MVP-tree, and M-tree occurring in the on-line search phase. Section VI concludes the paper.

III. The Theory Underlying *MetricMap*

In this section, we present the theory underlying *MetricMap*. This theory is important in understanding: 1) how database objects are mapped to a target space; 2) how the dimensionality of the target space is reduced; and 3) how to deal with embeddable and unembeddable objects in both the embedding phase and the on-line search phase. This theory also helps to understand how the subsequent query processing algorithms described in Section IV work in a low dimensional target space.

Section III-A presents notation and some basic definitions. In Section III-B, we consider a database $\mathcal{D}$ of N objects, a distance function d, which is a pseudometric, where $d(O_i, O_j)$, or simply $d_{i,j}$, represents the distance between O_i and O_j, for all $0 \leq i, j \leq N-1$. Thus, $(\mathcal{D}, d)$ is a pseudometric space [24]. We choose a sample $\mathcal{A}$ of $k+1, k < N$, objects from the database and embed the $k+1$ objects into a k-dimensional pseudo-Euclidean space, R^k. Section III-C establishes an orthogonal basis for R^k. Section III-D considers a lower dimensional space $R^n, n \leq k$, by ignoring those dimensions $\dim_j$ where after embedding all the objects of $\mathcal{A}$ into R^k, the differences among the jth components of the corresponding vectors are small. To further reduce the dimensionality, Section III-E considers an orthonormal basis and Section III-F establishes an m-dimensional pseudo-Euclidean space $R^m, m \ll k$. The objects corresponding to the dimensions of R^m are chosen as *reference objects*.

Once the target space R^m is established, the *MetricMap* algorithm maps each object O_* in the database to a point (vector) p_* in the target space by comparing the object with the reference objects. (We refer to the point p_* as the *image* of the object O_*.) The coordinate of p_* is calculated through matrix multiplication. An object may or may not be embeddable in the target space. Section III-G deals with the projection of an embeddable object onto the target space, and Section III-H handles the projection of an unembeddable object. In the beginning of the on-line search phase, the query object Q will also be compared with the reference objects, so that the calculated distances can be used for projecting Q onto the target space R^m.

A. Notation and Basic Definitions

Our notation is mainly based on [25] and [26]. Let $a_i, 1 \leq i \leq k$ be a base vector in the pseudo-Euclidean space R^k. We use $\{a_i\}_{1 \leq i \leq k}$, or simply $\{a_i\}$ when the context is clear, to represent $\{a_1, \ldots, a_k\}$. Use $x_{<a>} = (x^i_{<a>})_{1 \leq i \leq k}$ to represent the coordinate of x with regard to the basis $\{a_i\}_{1 \leq i \leq k}$. Let c_i's be real numbers. The matrix containing these numbers is

$$(c_i)_{1 \leq i \leq k} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix}.$$

Let $c_{i,j}$ be real numbers. The matrix containing these numbers is

$$(c_{i,j})_{1 \leq i \leq k, 1 \leq j \leq k} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,k} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,k} \\ \vdots & & & \vdots \\ c_{k,1} & c_{k,2} & \cdots & c_{k,k} \end{pmatrix}.$$

A diagonal matrix is represented as

$$\mathrm{diag}(c_i)_{1 \leq i \leq k} = \begin{pmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & c_k \end{pmatrix}.$$

B. Pseudo-Euclidean Space R^k

We define a mapping α from the sample $\mathcal{A}$ of the database $\mathcal{D}$ mentioned in the beginning of this section to R^k as follows:

$$\alpha \colon \mathcal{A} \to R^k$$

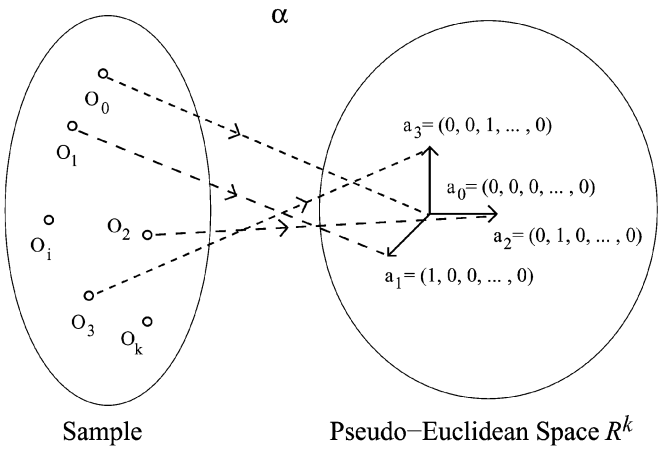

Fig. 2. The mapping α.

such that

$$\alpha(O_0) = a_0 = (0, \ldots, 0)$$
$$\alpha(O_i) = a_i = (0, \ldots, 1_{(i)}, \ldots, 0), \quad 1 \le i \le k.$$

Intuitively, we map O_0 to the origin and map the other sampling objects to vectors (points) $\{a_i\}_{1\le i\le k}$ in R^k so that each of the sampling objects corresponds to a base vector in R^k (see Fig. 2).

Let

$$M(\psi_{\langle a\rangle}) = (m_{i,j})_{1\le i,j\le k}$$

where

$$m_{i,j} = \left(d_{i,0}^2 + d_{j,0}^2 - d_{i,j}^2\right)/2, \quad 1 \le i, \quad j \le k. \tag{4}$$

We define another mapping ψ as follows:

$$\psi\colon R^k \times R^k \to R$$

such that

$$\psi(x, y) = x^T M(\psi_{\langle a\rangle}) y$$

where x^T is the transpose of vector x. Notice that $\psi(a_i, a_j) = m_{i,j}$. ψ is a *symmetric bilinear form* of R^k. $M(\psi_{<a>})$ is the matrix of ψ with regard to the basis $\{a_i\}_{1\le i\le k}$. The vector space R^k equipped with the symmetric bilinear form ψ is called a *pseudo-Euclidean space* [27]. For any two vectors, $x, y \in R^k, \psi(x, y)$ is called the *inner product* of x and y, and $\|x - y\|^2 = \psi(x - y, x - y)$ is called the *squared distance* between x and y.

Note that the inner product that we define here is an extension of the inner product of two vectors in a Euclidean space. Referring to Fig. 3, according to the cosine law, we have

$$d_{i,j}^2 = d_{i,0}^2 + d_{j,0}^2 - 2d_{i,0}d_{j,0}\cos\theta$$

Thus

$$d_{i,0}d_{j,0}\cos\theta = \left(d_{i,0}^2 + d_{j,0}^2 - d_{i,j}^2\right)/2.$$

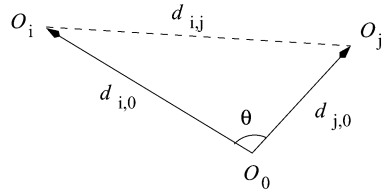

Fig. 3. Illustration of the inner product of two vectors.

By definition, the left-hand side is the inner product of $O_0\vec{O}_i$ and $O_0\vec{O}_j$. The right-hand side is exactly the $m_{i,j}$ as given in (4).

A pseudo-Euclidean space differs from a Euclidean space in that the inner product $\psi(x, y)$ is indefinite and the number $\psi(x, x)$ can be positive, negative or zero, depending on the vector x. A vector $x \neq 0$ is called *space-like*, if $\psi(x, x) > 0$; *time-like*, if $\psi(x, x) < 0$; a *light-vector*, if $\psi(x, x) = 0$. The *light-cone* is the set of all light-vectors.

Notice that we map any two database objects O_i, O_j to two orthonormal unit vectors a_i, a_j without considering the distance $d_{i,j}$. This mapping may cause severe deformations of the target space. In Section III-C, we will introduce a series of transformations to straighten the target space so that the target space gets closer to a Euclidean space as much as possible.

C. ψ-Orthogonal Basis $\{e_i\}$

Since the matrix $M(\psi_{\langle a\rangle})$ is real symmetric, there is an orthogonal matrix $Q = (q_{i,j})_{1\le i,j\le k}$ and a diagonal matrix $D = \mathrm{diag}(\lambda_i)_{1\le i\le k}$ such that

$$Q^T M\left(\psi_{\langle a\rangle}\right) Q = D \tag{5}$$

where Q^T is the transpose of Q, $\lambda_i's$ are eigenvalues of $M(\psi_{\langle a\rangle})$ arranged in some order, and columns of Q are the corresponding eigenvectors [25]. Let $(e_1, \ldots, e_k) = (a_1, \ldots, a_k)Q$ or equivalently

$$(a_1, \ldots, a_k) = (e_1, \ldots, e_k)Q^T. \tag{6}$$

Then $\{e_i\}_{1\le i\le k}$ is another basis of R^k; cf. Fig. 4. Note that the coordinate of e_j with regard to $\{a_i\}_{1\le i\le k}$ is the jth column of matrix Q, and the coordinate of a_j with regard to $\{e_i\}_{1\le i\le k}$ is the jth row of Q.

Example 1: Suppose the sample $\mathcal{A}$ we choose has seven objects $O_0, O_1, \ldots, O_6$ and the distance matrix $(d_{i,j})_{0\le i\le 6, 0\le j\le 6}$ for this sample is

$$\begin{pmatrix} 0.0 & 27.8 & 48.0 & 72.8 & 47.6 & 76.7 & 53.6 \\ 27.8 & 0.0 & 35.7 & 81.5 & 69.6 & 73.2 & 47.2 \\ 48.0 & 35.7 & 0.0 & 98.5 & 85.6 & 105.0 & 76.5 \\ 72.8 & 81.5 & 98.5 & 0.0 & 43.5 & 60.9 & 51.9 \\ 47.6 & 69.6 & 85.6 & 43.5 & 0.0 & 76.7 & 62.7 \\ 76.7 & 73.2 & 105.0 & 60.9 & 76.7 & 0.0 & 29.2 \\ 53.6 & 47.2 & 76.5 & 51.9 & 62.7 & 29.2 & 0.0 \end{pmatrix}.$$

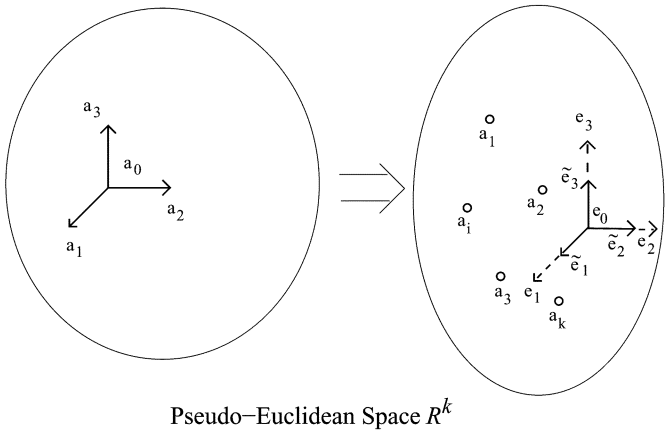

Fig. 4. Basis transformation in R^k.

Then $M(\psi_{\langle a\rangle})$ is as shown in the equation at the bottom of the page. The orthogonal matrix Q for the above $M(\psi_{\langle a\rangle})$ is

$$\begin{pmatrix} -0.624 & 0.404 & -0.004 & -0.124 & 0.488 & -0.440 \\ 0.535 & 0.155 & -0.254 & 0.584 & 0.316 & -0.429 \\ 0.366 & -0.267 & 0.479 & -0.499 & 0.062 & -0.559 \\ -0.141 & -0.740 & -0.534 & -0.130 & 0.360 & -0.040 \\ 0.413 & 0.399 & -0.297 & -0.554 & 0.380 & 0.361 \\ -0.011 & 0.186 & -0.576 & -0.265 & -0.620 & -0.422 \end{pmatrix}$$

and the diagonal matrix D for the above $M(\psi_{\langle a\rangle})$ is

$$\begin{pmatrix} 12\,973.1 & & & & & \\ & 4280.8 & & & & \\ & & 2151.6 & & & \\ & & & -0.6 & & \\ & & & & 0.1 & \\ & & & & & 0.06 \end{pmatrix}.$$

The coordinate of a_1 with regard to the orthogonal basis $\{e_i\}_{1\le i\le 6}$ is $(-0.624, 0.404, -0.004, -0.124, 0.488, -0.440)$, the coordinate of a_2 is $(0.535, 0.155, -0.254, 0.584, 0.316, -0.429)$, and so on. On the other hand, the coordinate of e_1 with regard to $\{a_i\}_{1\le i\le 6}$ is $(-0.624, 0.535, 0.366, -0.141, 0.413, -0.011)$, the coordinate of e_2 is $(0.404, 0.155, -0.267, -0.740, 0.399, 0.186)$, and so forth. □

Since there will often be three different bases of a space in our discussion, we introduce a new notation, which is not common, but convenient. Let $x = (x^1, \ldots, x^k)$ be a vector and $\{a_i\}_{1\le i\le k}$ be a basis of R^k. The coordinate of x with regard to $\{a_i\}_{1\le i\le k}$ is denoted by $x_{\langle a\rangle} = (x^i_{\langle a\rangle})_{1\le i\le k}$. Using this notation, the relation between $\{a_j\}$ and $\{e_j\}$ may be written as

$$e_{j\langle a\rangle} = (q_{1,j}, \ldots, q_{k,j}), \quad 1 \le j \le k$$

and

$$a_{j\langle e\rangle} = (q_{j,1}, \ldots, q_{j,k}), \quad 1 \le j \le k$$

where $e_{j\langle a\rangle}$ is the coordinate of e_j with regard to $\{a_i\}_{1\le i\le k}$, and $a_{j\langle e\rangle}$ is the coordinate of a_j with regard to $\{e_i\}_{1\le i\le k}$. Let x be a vector in R^k. Then

$$\left(x^1_{\langle a\rangle}, \ldots, x^k_{\langle a\rangle}\right) = \left(x^1_{\langle e\rangle}, \ldots, x^k_{\langle e\rangle}\right) Q^T. \tag{7}$$

Therefore, the matrix of the bilinear form ψ with regard to $\{e_i\}_{1\le i\le k}$ is

$$M\left(\psi_{\langle e\rangle}\right) = Q^T M(\psi_{\langle a\rangle}) Q = D.$$

That is, the basis $\{e_i\}_{1\le i\le k}$ is ψ-*orthogonal*. Let x, y be two vectors in R^k. Then

$$\begin{aligned} \psi(x,y) &= x^T_{\langle a\rangle} M\left(\psi_{\langle a\rangle}\right) y_{\langle a\rangle} \\ &= x^T_{\langle e\rangle} Q^T M\left(\psi_{\langle a\rangle}\right) Q y_{\langle e\rangle} \\ &= x^T_{\langle e\rangle} D y_{\langle e\rangle} \\ &= \Sigma_{i=1}^k \lambda_i x^i_{\langle e\rangle} y^i_{\langle e\rangle} \\ \|x-y\|^2 &= \Sigma_{i=1}^k \lambda_i \left(x^i_{\langle e\rangle} - y^i_{\langle e\rangle}\right)^2. \end{aligned} \tag{8}$$

In particular, we have

$$\psi(a_i, a_j) = \Sigma_{l=1}^k \lambda_l q_{i,l} q_{j,l}.$$

Remark 1: If the matrix $M(\psi_{\langle a\rangle})$ has negative eigenvalues, the squared distance between two vectors in the pseudo-Euclidean space R^k may be negative. That is why we never say the "distance" between vectors in a pseudo-Euclidean space. Furthermore, the fact that the squared distance between two vectors vanishes does not imply that these two vectors are the same. These situations cannot happen in a Euclidean space.

Notice that, based on the definition of the squared distance in (8), if an eigenvalue λ_l is zero or very small, the difference between coordinates along this dimension does not contribute much to the squared distance between two vectors. In Sections III-D, III-E, and III-F, we will reduce the dimensionality of the target space by excluding those dimensions with zero or very small eigenvalue values, so as to reduce the total number of distance calculations during the on-line search phase. This idea is similar to principal component analysis [28], though the techniques employed are different.

$$\begin{pmatrix} 774.0 & 903.0 & -285.1 & -903.9 & 647.9 & 709.1 \\ 903.0 & 2305.9 & -1046.5 & -1377.9 & -1418.7 & -336.9 \\ -285.1 & -1046.5 & 5301.3 & 2838.2 & 3741.0 & 2742.4 \\ -903.9 & -1377.9 & 2838.2 & 2263.9 & 1131.9 & 605.8 \\ 647.9 & -1418.7 & 3741.0 & 1131.9 & 5886.0 & 3953.1 \\ 709.1 & -336.9 & 2742.4 & 605.8 & 3953.1 & 2874.0 \end{pmatrix}$$

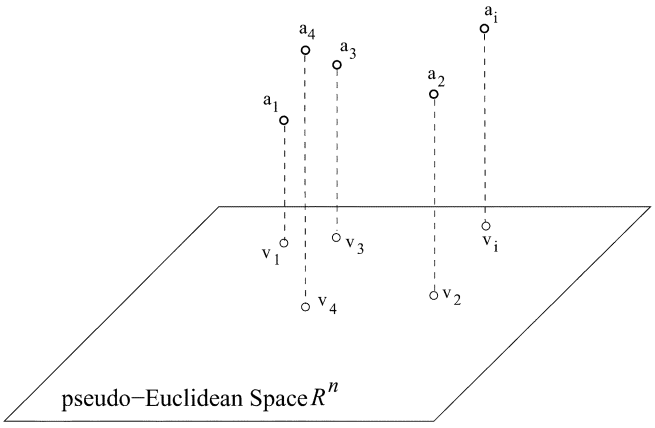

Fig. 5. Projecting R^k onto R^n.

D. Pseudo-Euclidean Space $R^n, n \leq k$

Assume that the eigenvalues of the matrix $M(\psi_{\langle a\rangle})$ are ordered as follows: first n^+ positive eigenvalues, then n^- negative ones and finally zeroes; $n = n^+ + n^-$. Then

$$R^k = V \oplus R^0$$

where $\oplus$ denotes the direct sum of two subspaces: $V = R^{(n^+ + n^-)}$ is the subspace generated by $\{e_i\}_{1\leq i\leq n}$, and R^0 is the subspace generated by $\{e_i\}_{n+1\leq i\leq k}$ [26]. Let $\phi = \psi|_{V\times V}$. Then ϕ is a nondegenerate bilinear form over $V \times V$. The set of vectors $\{e_i\}_{1\leq i\leq n}$ is a ϕ-orthogonal basis of subspace V.

Let x be a vector in R^k. We define the ψ-orthogonal projection

$$\Pi\colon R^k \to R^n$$

such that

$$\Pi\left(x^1_{\langle e\rangle}, \ldots, x^n_{\langle e\rangle}, x^{n+1}_{\langle e\rangle}, \ldots, x^k_{\langle e\rangle}\right) = \left(x^1_{\langle e\rangle}, \ldots, x^n_{\langle e\rangle}\right).$$

Let v_j denote $\Pi(a_j)$; cf. Fig. 5. Let $Q_{[kn]}$ be the $k \times n$ matrix consisting of the first n columns of the orthogonal matrix Q, namely $Q_{[kn]} = (q_{i,j})_{1\leq i\leq k, 1\leq j\leq n}$. Then, from the definition of Π and (6), we have

$$(v_1, \ldots, v_k) = (e_1, \ldots, e_n)Q^T_{[kn]} \tag{9}$$

i.e., the coordinate of v_j with regard to $\{e_i\}_{1\leq i\leq n}$ includes the first n elements of the jth row of the matrix Q, namely $v_{j\langle e\rangle} = (q_{j,1}, \ldots, q_{j,n})$.

All the discussions about the inner product can now be summarized as follows:

$$\begin{aligned}\phi(v_i, v_j) &= \Sigma^n_{l=1}\lambda_l q_{i,l} q_{j,l}\\ &= \Sigma^k_{l=1}\lambda_l q_{i,l} q_{j,l}\\ &= \psi(a_i, a_j)\\ &= \left(d^2_{i,0} + d^2_{j,0} - d^2_{i,j}\right)/2.\end{aligned}$$

Thus, the vector representation of the pseudometric space $(\mathcal{A}, d)$ is the mapping

$$\beta\colon \mathcal{A} \to R^{(n^+ + n^-)}$$

satisfying

$$\beta(O_0) = \Pi(\alpha(O_0)) = \Pi(a_0) = (0, \ldots, 0)_n$$

and

$$\beta(O_j) = \Pi(\alpha(O_j)) = \Pi(a_j) = v_j, \quad 1 \leq j \leq k.$$

Definition 1: A vector representation ρ of the pseudometric space $(\mathcal{A}, d)$ is an *isometric representation* if for any $O_i, O_j \in \mathcal{A}, \|\rho(O_i) - \rho(O_j)\|^2 = d^2_{i,j}$.

From the above discussions, we have the following.

Theorem 1: The mapping β is an isometric representation of the pseudometric space $(\mathcal{A}, d)$ in the pseudo-Euclidean space $R^{(n^+ + n^-)}$. That is, for any pair of indices $i, j, 0 \leq i, j \leq k, \|v_i - v_j\|^2 = \phi(v_i - v_j, v_i - v_j) = d^2_{i,j}$.

Proof: The result follows immediately by observing that

$$\begin{aligned}&\|v_i - v_j\|^2\\ &\quad= \phi(v_i - v_j, v_i - v_j)\\ &\quad= \psi(a_i - a_j, a_i - a_j)\\ &\quad= m_{i,i} - m_{i,j} - m_{j,i} + m_{j,j}\\ &\quad= \left(d^2_{i,0} + d^2_{i,0} - d^2_{i,i}\right)/2 - \left(d^2_{i,0} + d^2_{j,0} - d^2_{i,j}\right)/2\\ &\qquad- \left(d^2_{j,0} + d^2_{i,0} - d^2_{j,i}\right)/2 + \left(d^2_{j,0} + d^2_{j,0} - d^2_{j,j}\right)/2\\ &\quad= d^2_{i,j}.\end{aligned}$$

□

Theorem 1 describes the relation between the distance d in the pseudometric space and the squared distance in the corresponding pseudo-Euclidean space, stating the fact that the mapping β preserves d.

E. ϕ-Orthonormal Basis $\{\tilde{e}_i\}$

Define $\operatorname{sign}(\lambda_i)$ to be

$$\operatorname{sign}(\lambda_i) = \begin{cases} 1, & \text{if } \lambda_i > 0\\ 0, & \text{if } \lambda_i = 0\\ -1, & \text{if } \lambda_i < 0.\end{cases}$$

Let $J = \operatorname{diag}(\operatorname{sign}(\lambda_i))_{1\leq i\leq k}$ and $\tilde{D} = \operatorname{diag}(d_i)_{1\leq i\leq k}$, where

$$d_i = \begin{cases} |\lambda_i|, & \text{if } \lambda_i \neq 0\\ 1, & \text{otherwise}\end{cases}$$

Let $\tilde{Q} = Q \times \tilde{D}^{-1/2}$. Then

$$\begin{aligned}\tilde{Q}^T M(\psi_{\langle a\rangle})\tilde{Q} &= \tilde{D}^{-1/2} Q^T M(\psi_{\langle a\rangle}) Q \tilde{D}^{-1/2}\\ &= \tilde{D}^{-1/2}\operatorname{diag}(\lambda_i)\tilde{D}^{-1/2}\\ &= J.\end{aligned}$$

This means that the first n columns of the matrix $\tilde{Q}$ are ψ-orthonormal vectors. Let

$$\tilde{e}_i = \frac{e_i}{\sqrt{d_i}}, \quad 1 \le i \le k$$

or equivalently

$$(\tilde{e}_1, \ldots, \tilde{e}_k) = (e_1, \ldots, e_k)\tilde{D}^{-1/2}. \tag{10}$$

Then, the set of vectors $\{\tilde{e}_i\}_{1 \le i \le n}$ is a *ϕ-orthonormal basis* of R^n.

From (6) and (10), we have

$$(a_1, \ldots, a_k) = (\tilde{e}_1, \ldots, \tilde{e}_k)\tilde{D}^{1/2}Q^T. \tag{11}$$

From (9) and (10), we have

$$(v_1, \ldots, v_k) = (\tilde{e}_1, \ldots, \tilde{e}_n)\tilde{D}^{1/2}_{[n]}Q^T_{[kn]} \tag{12}$$

where $\tilde{D}_{[n]}$ is the nth leading principal submatrix of the matrix $\tilde{D}$, i.e., $\tilde{D}_{[n]} = \mathrm{diag}(|\lambda_i|)_{1 \le i \le n}$. The coordinate of v_j with regard to the basis $\{\tilde{e}_i\}_{1 \le i \le n}$ includes the first n elements of the jth row of the matrix $\tilde{T} = Q \times \tilde{D}^{1/2}$, i.e., $v_{j\langle\tilde{e}\rangle} = (\sqrt{|\lambda_1|}q_{j,1}, \ldots, \sqrt{|\lambda_n|}q_{j,n})$. Let x, y be two vectors in R^n. Then

$$\psi(x, y) = \Sigma_{i=1}^{n}\mathrm{sign}(\lambda_i)x^i_{\langle\tilde{e}\rangle}y^i_{\langle\tilde{e}\rangle}$$

and

$$||x - y||^2 = \Sigma_{i=1}^{n}\mathrm{sign}(\lambda_i)(x^i_{\langle\tilde{e}\rangle} - y^i_{\langle\tilde{e}\rangle})^2.$$

Example 2: Refer to Example 1. Since all eigenvalues are nonzero numbers, we have $n = k = 6$, and $R^n = R^k = R^6$ in our case. Thus, $v_j = a_j$, where v_j is a_j projected onto R^n. The coordinate of v_1 with regard to $\{\tilde{e}_i\}_{1 \le i \le n}$ is $(-0.624 \times \sqrt{|12\,973.1|}, 0.404 \times \sqrt{|4280.8|}, -0.004 \times \sqrt{|2151.6|}, -0.124 \times \sqrt{|0.6|}, 0.488 \times \sqrt{|0.1|}, -0.440 \times \sqrt{|0.06|}) = (-71.073, 26.433, -0.186, -0.096, 0.154, 0.108)$. Similarly, the coordinate of v_2 with regard to $\{\tilde{e}_i\}_{1 \le i \le n}$ is $(60.936, 10.141, -11.782, 0.452, 0.1, -0.105)$, and so on. □

F. Pseudo-Euclidean Space $R^m, m < n$

In practice, the number of objects in the sample $\mathcal{A}$, i.e., $k+1$, may be large. Consequently, the dimension of R^n could still be large. From (8), we know that the eigenvalues represent the extensions of variances of the objects in $\mathcal{A}$ in the corresponding dimensions. To avoid dealing with a space of very high dimensionality, we ignore the dimensions along which the eigenvalues are small. Specifically, suppose the eigenvalues are sorted in descending order by their absolute values. Let $\{\lambda_i\}_{1 \le i \le m}$ be the first m eigenvalues, $m < n, m = m^- + m^+, m^- \le n^-$, and $m^+ \le n^+$. The mapping

$$\gamma\colon \mathcal{A} \to R^{(m^+ + m^-)}$$

is the projection of the isometric vector representation β onto the subspace spanned by the first m vectors in the ϕ-orthonormal basis. The first m elements of the ith row of the corresponding $\tilde{T}$ would give the coordinates of $\gamma(O_i)$ for the reduced vector representation, i.e.,

$$\gamma(O_i) = (\sqrt{|\lambda_1|}q_{i,1}, \ldots, \sqrt{|\lambda_m|}q_{i,m}.)$$

Let x, y be two vectors in R^n. Then

$$\varphi(x, y) = \Sigma_{i=1}^{m}\mathrm{sign}(\lambda_i)x^i_{\langle\tilde{e}\rangle}y^i_{\langle\tilde{e}\rangle}$$

is the approximate representation of $\psi(x, y)$ for the corresponding vectors in R^m, and

$$\varphi(x - y, x - y) = \Sigma_{i=1}^{m}\mathrm{sign}(\lambda_i)\left(x^i_{\langle\tilde{e}\rangle} - y^i_{\langle\tilde{e}\rangle}\right)^2 \tag{13}$$

is the approximate representation of $||x - y||^2$. Note that

$$||x - y||^2 - \varphi(x - y, x - y) = \Sigma_{i=m+1}^{n}\lambda_i\left(x^i_{\langle e\rangle} - y^i_{\langle e\rangle}\right)^2.$$

Let w_i be v_i projected onto R^m, i.e., $w_i = (\sqrt{|\lambda_1|}q_{i,1}, \ldots, \sqrt{|\lambda_m|}q_{i,m})$. We have

$$||w_i - w_j||^2 - \varphi(w_i - w_j, w_i - w_j) = \Sigma_{l=m+1}^{n}\lambda_l(q_{i,l} - q_{j,l})^2.$$

This equation provides a way to estimate the error incurred by the approximation introduced above.

Theorem 2: Let $\triangle_{i,j} = ||w_i - w_j||^2 - \varphi(w_i - w_j, w_i - w_j)$. Then $|\triangle_{i,j}| \le 4|\lambda_{m+1}|$.

Proof: The result follows by observing that

$$\begin{aligned}|\triangle_{i,j}| &\le \Sigma_{l=m+1}^{n}|\lambda_l|(q_{i,l} - q_{j,l})^2 \\ &\le |\lambda_{m+1}|\Sigma_{l=m+1}^{n}(q_{i,l} - q_{j,l})^2 \\ &\le |\lambda_{m+1}|\Sigma_{l=m+1}^{n}2(|q_{i,l}|^2 + |q_{j,l}|^2) \\ &\le 2|\lambda_{m+1}|\Sigma_{l=1}^{n}(|q_{i,l}|^2 + |q_{j,l}|^2) \\ &= 4|\lambda_{m+1}|.\end{aligned}$$

□

Example 3: Refer to Example 2. After the eigenvalues are sorted according to their absolute values, we have

$$\{\lambda_i\}_{1 \le i \le 6} = \{12\,973.1, 4280.8, 2151.6, -0.6, 0.1, 0.06\}.$$

Since λ_4, λ_5, and λ_6 are very small in comparison with λ_1, λ_2, and λ_3, we ignore those three dimensions corresponding to λ_4, λ_5, and λ_6. Thus, $R^m = R^3$ in our case. According to Theorem 2, we have $\triangle_{i,j} \le 4|\lambda_4| = 2.4$. □

Notice that the first mapping α maps one of the objects in the sampling set to the origin and each of the other sampling objects to a basis vector of R^k; each sampling object thus corresponds to one dimension of R^k. After the first transformation through which we found an orthogonal basis, we removed $k-m$ dimensions whose eigenvalues are small. We also removed the objects that correspond to those $k-m$ dimensions. The remaining $m+1$ objects are reference objects, denoted $\mathrm{ref}_j, 0 \le j \le m$.

In general, the database $\mathcal{D}$ is sizable and the number of the reference objects is relatively small. After the target space R^m is established, all the database objects need to be mapped to vectors in the target space. When a query object is submitted, which may not exist in the database, the query object also needs to be

mapped to a vector in the target space. These objects may or may not be embeddable in the target space. In the following two subsections, we discuss how to embed an object into the target space in cases where the object is embeddable and unembeddable, respectively.

G. Projection of Embeddable Objects

We can map each object O_* in the database to a point (vector) p_* in the target space R^m based on the distances between O_* and the reference objects $\mathrm{ref}_j, 0 \le j \le m$. To begin with, add O_* into $\mathcal{A}$. Let the distances between O_* and O_j be given as:

$$d_{*,j} = d(O_*, O_j), \quad 0 \le j \le k$$

Assume that the object O_* is isometrically represented by a vector $u_* \in R^k$, i.e.,

$$\|u_* - v_j\|^2 = d_{*,j}^2, \quad 0 \le j \le k$$

or equivalently

$$\phi(u_*, v_j) = \left(d_{*,0}^2 + d_{j,0}^2 - d_{*,j}^2\right)/2, \quad 1 \le j \le k \tag{14}$$

$\phi(u_*, u_*) = d_{*,0}$. Let $Q_{[km]}$ be the matrix consisting of the first m columns of the matrix Q, namely $Q_{[km]} = (q_{i,j})_{1\le i\le k, 1\le j\le m}$. Let $\tilde{D}_{[m]}$ be the mth leading principal submatrix of the matrix $\tilde{D}$, i.e., $\tilde{D}_{[m]} = \mathrm{diag}(|\lambda_i|)_{1\le i\le m}$. Then from (9) and (12),

$$\begin{aligned}(w_1, \ldots, w_k) &= (e_1, \ldots, e_m) Q_{[km]}^T \\ &= (\tilde{e}_1, \ldots, \tilde{e}_m) \tilde{D}_{[m]}^{1/2} Q_{[km]}^T. \end{aligned} \tag{15}$$

Let r_* be the ϕ-orthogonal projection of u_* onto R^m. r_* can be represented as a linear combination of the set of vectors $\{w_i\}_{1\le i\le m}, r_* = \Sigma_{i=1}^m r_*^i w_i$. Taking the inner product of r_* and $w_j, 1 \le j \le m$, we obtain $\phi(r_*, w_j) = \Sigma_{i=1}^m r_*^i \phi(w_i, w_j)$. Owing to the ϕ-orthogonality, $\phi(r_*, w_j) = \phi(u_*, w_j), 1 \le j \le m$. Hence

$$\Sigma_{i=1}^m r_*^i \phi(w_i, w_j) = \phi(u_*, w_j), \quad 1 \le j \le m. \tag{16}$$

Let the Gram matrix $G(w_1, \ldots, w_m) = (\phi(w_i, w_j))_{1\le i,j\le m}$, $b = (\phi(u_*, w_j))_{1\le j\le m}$. Then (16) can be re-written as $G(w_1, \ldots, w_m) r_* = b$. Since $G(w_1, \ldots, w_m)$ is nonsingular, i.e., its determinant is not zero, $r_* = [G(w_1, \ldots, w_m)]^{-1} b$.

Note that this equation gives the coordinate of r_* with regard to the basis $\{w_i\}_{1\le i\le m}$. To obtain the coordinate with regard to $\{e_i\}$ or $\{\tilde{e}_i\}$, we need the matrices of coordinate transformation. Let $Q_{[mm]}$ be the mth leading principal submatrix of the orthogonal matrix Q. Then from (15)

$$\begin{aligned}(w_1, \ldots, w_m) &= (e_1, \ldots, e_m) Q_{[mm]}^T \\ &= (\tilde{e}_1, \ldots, \tilde{e}_m) \tilde{D}_{[m]}^{1/2} Q_{[mm]}^T. \end{aligned} \tag{17}$$

So

$$r_{*\langle e\rangle} = Q_{[mm]}^T [G(w_1, w_2, \ldots, w_m)]^{-1} b$$

and

$$r_{*\langle \tilde{e}\rangle} = \tilde{D}_{[m]}^{1/2} Q_{[mm]}^T [G(w_1, w_2, \ldots, w_m)]^{-1} b \tag{18}$$

where $b = (\phi(u_*, w_j))_{1\le j\le m}$.

These equations can be simplified. From (17), we know that the coordinate of w_i with regard to $\{e_j\}_{1\le j\le m}$ is the ith row of $Q_{[mm]}$, i.e., $w_{i\langle e\rangle} = (q_{i,1}, \ldots, q_{i,m})$. According to the formula for the inner product, $\phi(w_i, w_j) = \Sigma_{l=1}^m \lambda_l q_{i,l} q_{j,l}, 1 \le i, j \le m$. Therefore, $G(w_1, \ldots, w_m) = (\phi(w_i, w_j))_{1\le i,j\le m} = Q_{[mm]} D_{[m]} Q_{[mm]}^T$. Substituting this into (18)

$$\begin{aligned} r_{*\langle \tilde{e}\rangle} &= \tilde{D}_{[m]}^{1/2} Q_{[mm]}^T \left(Q_{[mm]}^T\right)^{-1} D_{[m]}^{-1} Q_{[mm]}^{-1} b \\ &= \tilde{D}_{[m]}^{1/2} D_{[m]}^{-1} Q_{[mm]}^{-1} b \\ &= J_{[m]} \tilde{D}_{[m]}^{-1/2} Q_{[mm]}^{-1} b. \end{aligned}$$

Thus

$$r_{*\langle \tilde{e}\rangle} = J_{[m]} \tilde{D}_{[m]}^{-1/2} Q_{[mm]}^{-1} b. \tag{19}$$

Note that, after computing m eigenvalues and eigenvectors, one obtains the matrices $Q_{[mm]}$ and $\tilde{D}_{[m]}$. However, in general we do not know how large $\phi(u_*, w_j)$ is. What we know is $\phi(u_*, v_j) = (d_{*,0}^2 + d_{j,0}^2 - d_{*,j}^2)/2, 1 \le j \le k$. Thus, we have to use $\phi(u_*, v_j)$ as an approximate value for $\phi(u_*, w_j)$ to compute r_*. In other words, the formula we use in practice are

$$\bar{r}_{*\langle e\rangle} = Q_{[mm]}^T [G(w_1, w_2, \ldots, w_m)]^{-1} \bar{b}$$

and

$$\bar{r}_{*\langle \tilde{e}\rangle} = \tilde{D}_{[m]}^{1/2} Q_{[mm]}^T [G(w_1, \ldots, w_m)]^{-1} \bar{b} \tag{20}$$

where $\bar{b} = (\phi(u_*, v_j))_{1\le j\le m}$. Following the way to simplify $r_{*\langle \tilde{e}\rangle}$, (20) can be rewritten as

$$\bar{r}_{*\langle \tilde{e}\rangle} = J_{[m]} \tilde{D}_{[m]}^{-1/2} Q_{[mm]}^{-1} \bar{b}. \tag{21}$$

Notice that, Q is the matrix of the linear transformation that gives us the orthogonal basis $\{e_i\}$. With respect to this basis, the coordinates contribute differently to the overall distance between two objects. The differences between the coordinates are weighted by the eigenvalues. $\tilde{D}$ is the matrix of the linear transformation that finds the orthonormal basis $\{\tilde{e}_i\}$. The motivation of performing this transformation is to show the connection between pseudo-Euclidean space and Euclidean space. After this transformation, the squared distance becomes the sum of the squares of the differences between the coordinates, which is similar to the distance in a Euclidean space. The only difference now lies in the matrix J. In the case of a Euclidean space, J is a matrix of all 1's along the diagonal line. In the case of a pseudo-Euclidean space, however, J may have some -1's along the diagonal line.

Example 4: Refer to Example 3. After transforming $R^k = R^6$ to $R^m = R^3$, $J_{[m]}\tilde{D}_{[m]}^{-1/2}Q_{[mm]}^{-1}$ is

$$\begin{pmatrix} 0.001\,583 & -0.007\,018 & -0.007\,158 \\ 0.015\,499 & -0.011\,288 & 0.015\,132 \\ -0.016\,379 & -0.012\,907 & 0.007\,090 \end{pmatrix}.$$

Suppose we have an object O_*, whose distances to the reference objects O_0, O_1, O_2, and O_3 are 38.44, 44.12, 37.49, and 31.80, respectively. Then $\bar{b} = (918.48, 2686.71, 3176.18)$ and $\bar{r}_{*\langle\tilde{e}\rangle} = (-40.14, 31.97, -27.20)$. □

One may ask how well this works. The following three theorems estimate the error between $r_{*\langle\tilde{e}\rangle}$ and $\bar{r}_{*\langle\tilde{e}\rangle}$ when $\phi(u_*, v_i)$ is used in place of $\phi(u_*, w_i)$. All these theorems are based on the assumption that there is an object O_h in the database $\mathcal{D}$ that is very close to O_*. That is, if $\triangle_j = d_{*,j} - d_{h,j}, 0 \le j \le k$, then there exists a small positive real number ϵ such that

$$|\triangle_j| \le \epsilon, \quad 0 \le j \le k. \tag{22}$$

Theorem 3: Let $||X||_2$ denote the Euclidean norm of a vector or matrix X [25]. Let u_*, a_h be the vector representations (or images) of O_* and O_h, respectively, and let $\lambda_{\min}$ be the nonzero eigenvalue with the smallest absolute value in $\{\lambda_i\}_{1\le i\le n}$. Then $||u_* - a_h||_2 \le \bar{\epsilon}$, where

$$\bar{\epsilon} = \frac{\epsilon}{|\lambda_{\min}|}\left[\Sigma_{j=1}^{k}(d_{h,0} + d_{h,j})^2\right]^{1/2}.$$

Proof: Since p_* can be isometrically embedded into $R^k, ||u_* - a_j||^2 = d_{*,j}^2, 0 \le j \le k$, or equivalently

$$\psi(u_*, a_j) = \left(d_{*,0}^2 + d_{j,0}^2 - d_{*,j}^2\right)/2, \quad 1 \le j \le k. \tag{23}$$

Let $b_* = (b_*^j)_{1\le j\le k}$ where $b_*^j = (d_{*,0}^2 + d_{j,0}^2 - d_{*,j}^2)/2, 1 \le j \le k$. Then (23) can be written as $M(\psi_{\langle a\rangle})u_{*\langle a\rangle} = b_*$. Thus, $(Q^T M(\psi_{\langle a\rangle})Q)(Q^T u_{*\langle a\rangle}) = Q^T b_*$. From (5) and (7), $Du_{*\langle e\rangle} = Q^T b_*$. Since $\lambda_j = 0, n+1 \le j \le k, u_{*\langle e\rangle}^j, n+1 \le j \le k$, can take any values. Let $\bar{D} = \mathrm{diag}(\bar{\lambda_j})_{1\le j\le k}$, where

$$\bar{\lambda_j} = \begin{cases} \lambda_j, & \text{if } j < n \\ \lambda_n, & \text{if } n \le j \le k. \end{cases}$$

We choose those $u^j, 1 \le j \le k$, that satisfy $\bar{D}u_{*\langle e\rangle} = Q^T b_*$. Thus, $u_{*\langle e\rangle} = \bar{D}^{-1}Q^T b_*$. Similarly, $a_{h\langle e\rangle} = \bar{D}^{-1}Q^T b_h$. Thus, $||u_* - a_h||_2 \le ||\bar{D}^{-1}||_2||Q^T||_2||b_* - b_h||_2$. Evaluating these norms, we get $||\bar{D}^{-1}||_2 \le (1/|\lambda_{\min}|), ||Q^T||_2 = 1$,

$$\begin{aligned} &||b_* - b_h||_2 \\ &\quad = \left[\Sigma_{j=1}^{k}\left(b_*^j - b_h^j\right)^2\right]^{1/2} \\ &\quad = \left[\Sigma_{j=1}^{k}\frac{1}{4}\left(d_{*,0}^2 - d_{*,j}^2 - d_{h,0}^2 + d_{h,j}^2\right)^2\right]^{1/2} \\ &\quad = \left[\Sigma_{j=1}^{k}\left(d_{h,0}\triangle_0 - d_{h,j}\triangle_j + \frac{1}{2}\triangle_0^2 - \frac{1}{2}\triangle_j^2\right)^2\right]^{1/2}. \end{aligned}$$

Omitting the infinitesimal of higher order and substituting inequality (22), we get $||b_* - b_h||_2 = \epsilon[\Sigma_{j=1}^{k}(d_{h,0} + d_{h,j})^2]^{1/2}$. Hence, $||u_* - a_h||_2 \le (\epsilon)/(|\lambda_{\min}|)[\Sigma_{j=1}^{k}(d_{h,0} + d_{h,j})^2]^{1/2}$. □

Theorem 4: For each $i, 1 \le i \le k, |\phi(u_*, v_i) - \phi(u_*, w_i)| \le |\lambda_{m+1}|(1 + \bar{\epsilon})$.

Proof: Let $u_{*\langle e\rangle} = (u^j)_{1\le j\le n}$. Since $v_{i\langle e\rangle} = (q_{i,1}, \ldots, q_{i,n})$ [cf. (9)] and w_i is the projection of v_i onto R^m, we obtain

$$\begin{aligned} &|\phi(u_*, v_i) - \phi(u_*, w_i)| \\ &\quad = \left|\Sigma_{j=m+1}^{n}\lambda_j u^j q_{i,j}\right| \\ &\quad = \left|\Sigma_{j=m+1}^{n}\lambda_j(q_{h,j} + \triangle v_j)q_{i,j}\right| \\ &\quad \le \left|\Sigma_{j=m+1}^{n}\lambda_j q_{h,j}q_{i,j}\right| + \left|\Sigma_{j=m+1}^{n}\lambda_j\triangle v_j q_{i,j}\right| \end{aligned}$$

where $\triangle v_j = u^j - q_{h,j}, 1 \le j \le n$.

The first term on the right-hand side is easy to estimate. Since Q is orthogonal, $\Sigma_{j=1}^{k}q_{i,j}^2 = 1, 1 \le i \le k$. Thus

$$\begin{aligned} &\left|\Sigma_{j=m+1}^{n}\lambda_j q_{h,j}q_{i,j}\right| \\ &\quad \le \Sigma_{j=m+1}^{n}|\lambda_j||q_{h,j}||q_{i,j}| \\ &\quad \le |\lambda_{m+1}|\left[\Sigma_{j=m+1}^{n}q_{h,j}^2\right]^{1/2}\left[\Sigma_{j=m+1}^{n}q_{i,j}^2\right]^{1/2} \\ &\quad \le |\lambda_{m+1}|. \end{aligned}$$

Similarly

$$\begin{aligned} &\left|\Sigma_{j=m+1}^{n}\lambda_j\triangle v_j q_{i,j}\right| \\ &\quad \le \Sigma_{j=m+1}^{n}|\lambda_j||\triangle v_j||q_{i,j}| \\ &\quad \le |\lambda_{m+1}|\left[\Sigma_{j=m+1}^{n}(\triangle v_j)^2\right]^{1/2}\left[\Sigma_{j=m+1}^{n}q_{i,j}^2\right]^{1/2} \\ &\quad \le |\lambda_{m+1}|\left[\Sigma_{j=m+1}^{n}(\triangle v_j)^2\right]^{1/2}. \end{aligned}$$

By Theorem 3

$$\begin{aligned} &\left[\Sigma_{j=m+1}^{n}(\triangle v_j)^2\right]^{1/2} \\ &\quad \le \left[\Sigma_{j=1}^{k}(\triangle v_j)^2\right]^{1/2} \\ &\quad = ||u_* - a_h||_2 \\ &\quad \le \bar{\epsilon}. \end{aligned}$$

Hence, $|\phi(u_*, v_i) - \phi(u_*, w_i)| \le |\lambda_{m+1}|(1 + \bar{\epsilon})$. □

Theorem 5:

$$\begin{aligned} ||\triangle r_*||_2 &= ||\bar{r}_{*\langle\tilde{e}\rangle} - r_{*\langle\tilde{e}\rangle}||_2 \\ &\le \sqrt{\frac{m}{|\lambda_m|}}|\lambda_{m+1}|(1 + \bar{\epsilon})\left\|Q_{[mm]}^{-1}\right\|_2 \\ &\le \sqrt{m|\lambda_{m+1}|}(1 + \bar{\epsilon})\left\|Q_{[mm]}^{-1}\right\|_2 \end{aligned}$$

Proof: Subtracting (19) from (21), $\triangle r_* = J_{[m]}\tilde{D}_{[m]}^{-1/2}Q_{[mm]}^{-1}(\bar{b} - b)$. Hence

$$||\triangle r_*||_2 \le \left\|J_{[m]}\right\|_2\left\|\tilde{D}_{[m]}^{-1/2}\right\|_2\left\|Q_{[mm]}^{-1}\right\|_2||\bar{b} - b||_2. \tag{24}$$

Evaluating these norms, we get $||J_{[m]}||_2 = 1, ||\tilde{D}_{[m]}^{-1/2}||_2 = (1)/(\sqrt{|\lambda_m|})$. By Theorem 4

$$\begin{aligned}||\bar{b} - b||_2 &= \left[\Sigma_{i=1}^m (\phi(u_*, v_i) - \phi(u_*, w_i))^2\right]^{1/2} \\ &\le \left[\Sigma_{i=1}^m (|\lambda_{m+1}|(1+\bar{\epsilon}))^2\right]^{1/2} \\ &= \sqrt{m}|\lambda_{m+1}|(1+\bar{\epsilon}).\end{aligned}$$

Substituting these into inequality (24), we get $||\triangle r_*||_2 \le \sqrt{(m)/(|\lambda_m|)}|\lambda_{m+1}|(1+\bar{\epsilon})||Q_{[mm]}^{-1}||_2 \le \sqrt{m|\lambda_{m+1}|}(1+\bar{\epsilon})||Q_{[mm]}^{-1}||_2$. □

From these theorems, it can be seen that the error between $r_{*\langle\tilde{e}\rangle}$ and $\bar{r}_{*\langle\tilde{e}\rangle}$ is negligible whenever m is not large and λ_{m+1} is small enough. The error estimation in these theorems assumes that O_* is an arbitrary object. If O_* is one of the reference objects O_i, the bound would be tighter as described in Theorem 6 below.

Theorem 6: If $\{\bar{w}_i\}_{1\le i\le k}$ are the coordinates calculated based on the formula (20), then $||\triangle w_i||_2 = ||\bar{w}_i - w_i||_2 \le \sqrt{m|\lambda_{m+1}|}||Q_{[mm]}^{-1}||_2$.

Proof: By replacing u_* with $v_i, 1 \le i \le k$, in (19) and (21), we get $w_{i\langle\tilde{e}\rangle} = J_{[m]}\tilde{D}_{[m]}^{-1/2}Q_{[mm]}^{-1}b_i, \bar{w}_{i\langle\tilde{e}\rangle} = J_{[m]}\tilde{D}_{[m]}^{-1/2}Q_{[mm]}^{-1}\bar{b}_i$, where $b_i = (\phi(v_i, w_j))_{1\le j\le m}$ and $\bar{b}_i = (\phi(v_i, v_j))_{1\le j\le m}$. Observe that

$$\begin{aligned}||\bar{b}_i - b_i||_2 &= \left[\Sigma_{j=1}^m (\phi(v_i, v_j) - \phi(v_i, w_j))^2\right]^{1/2} \\ &= \left[\Sigma_{j=1}^m \left(\Sigma_{l=m+1}^n \lambda_l q_{i,l} q_{j,l}\right)^2\right]^{1/2} \\ &\le \left[\Sigma_{j=1}^m \left(|\lambda_{m+1}|\Sigma_{l=m+1}^n |q_{i,l}q_{j,l}|\right)^2\right]^{1/2} \\ &\le \sqrt{m}|\lambda_{m+1}|.\end{aligned}$$

Thus, $||\bar{w}_i - w_i||_2 = ||J_{[m]}||_2||\tilde{D}_{[m]}^{-1/2}||_2||Q_{[mm]}^{-1}||_2||\bar{b}_i - b_i||_2 \le \sqrt{m|\lambda_{m+1}|}||Q_{[mm]}^{-1}||_2$. □

Note that the upper bound described in Theorem 6 is just one term of the upper bound for $\triangle r_*$ in Theorem 5, which is reasonable, since $\epsilon = \bar{\epsilon} = 0$ in this case.

H. Projection of Unembeddable Objects

In the previous subsection, we gave the projection formula for objects that are embeddable to the target space. Such objects include those that were used in the sampling set. In many cases, however, an object may not be isometrically embedded into R^k. For these objects, we still can derive a projection formula that is basically the same as (20). The problem with an unembeddable object is that (14) in Section III-G does not hold. As a consequence, the projection formula of (18) can not be established. To address this problem, we construct a $(k+1)$-dimensional space with the object O_* as the $(k+1)$th dimension. Then we project all the $(k+2)$ objects (i.e., the $k+1$ sampling objects in $\mathcal{D}$, plus the object O_*) onto R^m. The projection of the $(k+2)$th object establishes the formula for the object O_*. We then introduce a new mapping η to connect R^{k+1} with R^k, thus resulting in a formula very similar to the previous one for an embeddable object.

To begin with, let us first establish a $(k+1)$-dimensional space. Let $\mathcal{A}_* = \mathcal{A} \cup \{O_*\}$ and $\alpha_* : \mathcal{A}_* \to R^{k+1}$, such that: 1) $\alpha_*(O_0) = a_{*0} = (0, \ldots, 0, 0)$; 2) $\alpha_*(O_j) = a_{*j} = (0, \ldots, 1_{(j)}, \ldots, 0, 0), 1 \le j \le k$; and 3) $\alpha_*(O_*) = a_{*(k+1)} = (0, \ldots, 0, 1)$. Next, we define a symmetric bilinear form ψ_* over $R^{k+1} \times R^{k+1}$, such that (i) $\psi_*(a_{*i}, a_{*j}) = (d_{i,0}^2 + d_{j,0}^2 - d_{i,j}^2)/2, 1 \le i, j \le k$, and (ii) $\psi_*(a_{*(k+1)}, a_{*j}) = (d_{*,0}^2 + d_{j,0}^2 - d_{*,j}^2)/2, 1 \le j \le k$. Then define the matrix of the bilinear form with regard to $\{a_{*j}\}_{1\le j\le k+1}$: $M_*(\psi_{*<a*>}) = (\psi_*(a_{*i}, a_{*j}))_{1\le i,j\le k+1}$. Comparing the definition of ψ_* with that of ψ in Section III-B, one can see that for each pair of subscripts $i, j, 1 \le i, j \le k, \psi_*(a_{*i}, a_{*j}) = \psi(a_i, a_j)$. Moreover, the matrix $M(\psi_{\langle a\rangle})$ is simply the kth leading principal submatrix of $M_*(\psi_{*<a*>})$.

Analogously to how we dealt with ψ in Sections III-C through III-F, we can compute the eigenvectors of the matrix $M_*(\psi_{*<a*>})$ to obtain a ψ_*-orthonormal basis, say $\{\tilde{e}_{*i}\}_{1\le i\le k+1}$, of R^{k+1}. To derive a formula similar to (18) for an embeddable object, we need another ψ_*-orthonormal basis in R^{k+1}. We define a mapping $\eta : R^k \to R^{k+1}$ such that $\eta(x^1, \ldots, x^k) = (x^1, \ldots, x^k, 0)$, where $(x^1, \ldots, x^k)$ is the coordinate of a vector in R^k with respect to some basis of it.

Consider the subspace R^n of R^k defined in Section III-D. The mapping η associates R^n with a subspace, say R_*^n, in R^{k+1}. The space R^{k+1} can be represented as the direct sum of R_*^n and its ψ_*-orthogonal complement [26]. It follows that the union of a ψ_*-orthogonal basis of R_*^n and a ψ_*-orthogonal basis of its ψ_*-orthogonal complement will become a ψ_*-orthogonal basis of R^{k+1}. The subspace R^n is spanned by $\{\tilde{e}_i\}_{1\le i\le n}$. Theorem 7 specifies that the set of vectors $\{\eta(\tilde{e}_i)\}_{1\le i\le n}$ spans R_*^n. Theorem 8 guarantees that if $\{\tilde{e}_i\}_{1\le i\le n}$ is ψ-orthonormal then $\{\eta(\tilde{e}_i)\}_{1\le i\le n}$ is ψ_*-orthonormal.

Theorem 7: Let $x_1, \ldots, x_l$ be vectors in R^k and let $c_1, \ldots, c_l$ be real numbers. If $\Sigma_{i=1}^l c_i x_i = 0$, then $\Sigma_{i=1}^l c_i \eta(x_i) = 0$.

Proof: Let $x_i = (x_i^j)_{1\le j\le k}, 1 \le i \le l$. We have $\Sigma_{i=1}^l c_i x_i^j = 0, 1 \le j \le k$. Let $y_i = (y_i^j)_{1\le j\le k+1}, 1 \le i \le l$ and $\eta(x_i) = y_i, 1 \le i \le l$. By the definition of $\eta, y_i^j = x_i^j, 1 \le j \le k, 1 \le i \le l$, and $y_i^{k+1} = 0, 1 \le i \le l$. Thus, $\Sigma_{i=1}^l c_i y_i^j = 0, 1 \le j \le k+1$. Namely, $\Sigma_{i=1}^l c_i y_i = 0$. □

Theorem 8: Let x and y be two vectors in R^k. Then $\psi_*(\eta(x), \eta(y)) = \psi(x, y)$.

Proof: By the definition of η

$$\begin{aligned}\eta(x) &= \begin{pmatrix} x \\ 0 \end{pmatrix} \\ \eta(y) &= \begin{pmatrix} y \\ 0 \end{pmatrix} \\ M_*(\psi_{*\langle a*\rangle}) &= \begin{pmatrix} M\left(\psi_{\langle a\rangle}\right) & M_1 \\ M_1^T & \psi_*\left(a_{*(k+1)}, a_{*(k+1)}\right) \end{pmatrix}\end{aligned}$$

where $M_1 = (\psi_*(a_{*(k+1)}, a_{*j}))_{1\le j\le k}$. Thus

$$\begin{aligned}&\psi_*(\eta(x), \eta(y)) \\ &\quad = (x^T \quad 0) \\ &\qquad \times \begin{pmatrix} M(\psi_{\langle a\rangle}) & M_1 \\ M_1^T & \psi_*\left(a_{*(k+1)}, a_{*(k+1)}\right) \end{pmatrix} \begin{pmatrix} y \\ 0 \end{pmatrix} \\ &\quad = x^T M\left(\psi_{\langle a\rangle}\right) y \\ &\quad = \psi(x, y).\end{aligned}$$

□

Therefore, there is a ψ_*-orthonormal basis of R^{k+1} which includes $\{\eta(\tilde{e}_i)\}_{1\le i\le n}$ as a subset. The coordinate of a vector in R^{k+1} with respect to the basis mentioned above may be obtained from its coordinate with regard to $\{\tilde{e}_{*i}\}_{1\le i\le k+1}$, through multiplying the latter one by a certain nonsingular matrix (i.e., through coordinate transformation).

Note that $a_{*j}=\eta(a_j), 1\le j\le k$. According to Theorem 8 and (11)

$$(a_{*1},\ldots,a_{*k})=(\eta(\tilde{e}_1),\ldots,\eta(\tilde{e}_k))\tilde{D}^{1/2}Q^T.$$

Therefore the coordinate of the projection of a_{*j} with regard to $\{\eta(\tilde{e}_i)\}_{1\le i\le k}$ is simply the coordinate of a_j with regard to $\{\tilde{e}_i\}_{1\le i\le k}$. In parallel with the introduction of the subspace R^n_*, we can introduce a subspace R^m_* of R^{k+1} from R^m in R^k, and then consider the projection of the object O_* onto R^m_*. Let w_j be the ψ-orthogonal projection of a_j onto R^m. Then $w_{*j}=\eta(w_j)$ is the ψ_*-orthogonal projection of a_{*j} onto R^m_*. Since the set of projections $\{w_j\}_{1\le j\le m}$ spans R^m, according to Theorem 7, the set of projections $\{w_{*j}\}_{1\le j\le m}$ spans R^m_*. Furthermore, from (17)

$$(w_{*1},\ldots,w_{*m})=(\eta(\tilde{e}_1),\ldots,\eta(\tilde{e}_m))\tilde{D}^{1/2}_{[m]}Q^T_{[mm]}.$$

According to Theorem 6, the Gram matrix of $\{w_{*j}\}$ is simply the Gram matrix of $\{w_j\}$. Summarizing these results, we know that the coordinate of projecting a_{*j} onto R^m_* with regard to $\{\eta(\tilde{e}_i)\}_{1\le i\le m}$ can be computed using the equation

$$r_{*\langle\eta(\tilde{e})\rangle}=\tilde{D}^{1/2}_{[m]}Q^T_{[mm]}[G(w_1,\ldots,w_m)]^{-1}b_*$$

where the matrices $\tilde{D}_{[m]}, Q_{[mm]}$, and $G(w_1,\ldots,w_m)$ are the same as those in (20), and $b_*=(\psi_*(a_{*(k+1)},w_{*j}))_{1\le j\le m}$. Again, we do not know how large $\psi_*(a_{*(k+1)},w_{*j})$ is. What we can do is to replace it by $\psi_*(a_{*(k+1)},a_{*j})$, thus obtaining

$$\overline{r}_{*\langle\eta(\tilde{e})\rangle}=\tilde{D}^{1/2}_{[m]}Q^T_{[mm]}[G(w_1,\ldots,w_m)]^{-1}\overline{b}_* \qquad (25)$$

where $\overline{b}_*=(\psi_*(a_{*(k+1)},a_{*j}))_{1\le j\le m}$. By comparing (20) with (25), we conclude that no matter whether or not an object is embeddable in R^k, one can always use the same formula to calculate the projection of the object, though the resulting coordinates are with respect to the same basis represented in different dimensional spaces (more precisely, with respect to $\{\tilde{e}_i\}_{1\le i\le m}$ and $\{\eta(\tilde{e}_i)\}_{1\le i\le m}$, respectively).

IV. Query Processing Algorithms

After all the database objects are embedded in the pseudo-Euclidean space R^m, we can conduct a search in that space. Following Weber's approach [10], we allocate b_j bits to encode the jth dimension of R^m. Thus, the jth dimension is divided into 2^{b_j} partitions. The borders of these partitions are marked by $\{z_j[0], z_j[1],\ldots,z_j[2^{b_j}-1]\}$, where $z_j[0]$ is the minimum value and $z_j[2^{b_j}-1]$ is the maximum value along the jth dimension. Let O_i be an object in the database $\mathcal{D}$ and let $p_i\in R^m$ be the vector representation (image) of O_i. Let the coordinate of p_i be

$$l_{i,j}=\begin{cases} q_j-z_j[t_{i,j}+1] & \text{if } (t_{i,j}<t_{q,j}) \text{ and } (\lambda_j>0)\\ q_j-z_j[t_{i,j}] & \text{if } (t_{i,j}<t_{q,j}) \text{ and } (\lambda_j<0)\\ 0 & \text{if } (t_{i,j}=t_{q,j}) \text{ and } (\lambda_j>0)\\ \max(q_j-z_j[t_{i,j}], z_j[t_{i,j}+1]-q_j) & \text{if } (t_{i,j}=t_{q,j}) \text{ and } (\lambda_j<0)\\ z_j[t_{i,j}]-q_j & \text{if } (t_{i,j}>t_{q,j}) \text{ and } (\lambda_j>0)\\ z_j[t_{i,j}+1]-q_j & \text{if } (t_{i,j}>t_{q,j}) \text{ and } (\lambda_j<0)\end{cases}$$

$$u_{i,j}=\begin{cases} q_j-z_j[t_{i,j}] & \text{if } (t_{i,j}<t_{q,j}) \text{ and } (\lambda_j>0)\\ q_j-z_j[t_{i,j}+1] & \text{if } (t_{i,j}<t_{q,j}) \text{ and } (\lambda_j<0)\\ \max(q_j-z_j[t_{i,j}], z_j[t_{i,j}+1]-q_j) & \text{if } (t_{i,j}=t_{q,j}) \text{ and } (\lambda_j>0)\\ 0 & \text{if } (t_{i,j}=t_{q,j}) \text{ and } (\lambda_j<0)\\ z_j[t_{i,j}+1]-q_j & \text{if } (t_{i,j}>t_{q,j}) \text{ and } (\lambda_j>0)\\ z_j[t_{i,j}]-q_j & \text{if } (t_{i,j}>t_{q,j}) \text{ and } (\lambda_j<0)\end{cases}$$

Fig. 6. The value of $l_{i,j}$ and $u_{i,j}$, respectively.

$p_i=(p_{i,j})_{1\le j\le m}$ and let the partition into which $p_{i,j}$ falls be numbered $t_{i,j}$, i.e., $z_j[t_{i,j}]\le p_{i,j}<z_j[t_{i,j}+1]$. Then p_i can be encoded as a bit string $t_i=t_{i,1}t_{i,2}\cdots t_{i,m}$, where $t_{i,j}$ has b_j bits.

Now, in the on-line search phase, given the query object Q, we calculate the distances between Q and the reference objects $ref_j, 0\le j\le m$, and then embed Q into R^m based on these distances. Let $q\in R^m$ be the vector representation (image) of Q and let the coordinate of q be $q=(q_j)_{1\le j\le m}$. Let the partition into which q_j falls be numbered $t_{q,j}$, i.e., $z_j[t_{q,j}]\le q_j<z_j[t_{q,j}+1]$. Thus, q can also be encoded as a bit string $t_q=t_{q,1}t_{q,2}\ldots t_{q,m}$, where $t_{q,j}$ has b_j bits. We can derive a lower bound l_i and an upper bound u_i for the squared distance between q and p_i as follows:

$$l_i=\Sigma_{j=1}^m\lambda_j l_{i,j}^2 \qquad (26)$$

$$u_i=\Sigma_{j=1}^m\lambda_j u_{i,j}^2 \qquad (27)$$

where $l_{i,j}$ and $u_{i,j}$ are defined in Fig. 6.

Theorem 9 (Theorem 10, respectively) shows the l_i (u_i, respectively) described above is indeed a lower (upper, respectively) bound of the squared distance between q and p_i.

Theorem 9: Let l_i be as in (26). Then $\|q-p_i\|^2\ge l_i$.

Proof: Since $\|q-p_i\|^2=\Sigma_{j=1}^m\lambda_j(q_j-p_{i,j})^2$, it suffices to prove that for each $j, \lambda_j(q_j-p_{i,j})^2\ge\lambda_j l_{i,j}^2$. There are six cases to examine.

i) $t_{i,j}<t_{q,j}$ and $\lambda_j>0$. Thus, $q_j>p_{i,j}$. Since $p_{i,j}<z_j[t_{i,j}+1], q_j-p_{i,j}>q_j-z_j[t_{i,j}+1]=l_{i,j}$; see Fig. 7. Therefore, $\lambda_j(q_j-p_{i,j})^2>\lambda_j l_{i,j}^2$.

ii) $t_{i,j}<t_{q,j}$ and $\lambda_j<0$. In this case, we still have $q_j>p_{i,j}$. Since $p_{i,j}\ge z_j[t_{i,j}], q_j-p_{i,j}\le q_j-z_j[t_{i,j}]=l_{i,j}$. But now $\lambda_j<0$, and therefore $\lambda_j(q_j-p_{i,j})^2\ge\lambda_j l_{i,j}^2$.

iii) $t_{i,j}=t_{q,j}$ and $\lambda_j>0$. Thus, $l_{i,j}=0$. Obviously, $\lambda_j(q_j-p_{i,j})^2\ge 0$.

iv) $t_{i,j}=t_{q,j}$ and $\lambda_j<0$. Since $z_j[t_{i,j}]\le p_{i,j}, q_j<z_j[t_{i,j}+1], |q_j-p_{i,j}|\le\max(q_j-z_j[t_{i,j}], z_j[t_{i,j}+1]-q_j)$. Therefore, $(q_j-p_{i,j})^2\le l_{i,j}^2$ and $\lambda_j(q_j-p_{i,j})^2\ge\lambda_j l_{i,j}^2$.

v) $t_{i,j}>t_{q,j}$ and $\lambda_j>0$. Thus, $p_{i,j}>q_j$. Since $p_{i,j}\ge z_j[t_{i,j}], p_{i,j}-q_j\ge z_j[t_{i,j}]-q_j=l_{i,j}$. Therefore, $\lambda_j(q_j-p_{i,j})^2\ge\lambda_j l_{i,j}^2$.

vi) $t_{i,j}>t_{q,j}$ and $\lambda_j<0$. In this case, we still have $p_{i,j}>q_j$. Since $p_{i,j}<z_j[t_{i,j}+1], p_{i,j}-q_j<z_j[t_{i,j}+1]-q_j=l_{i,j}$. Therefore, $\lambda_j(q_j-p_{i,j})^2>\lambda_j l_{i,j}^2$. □

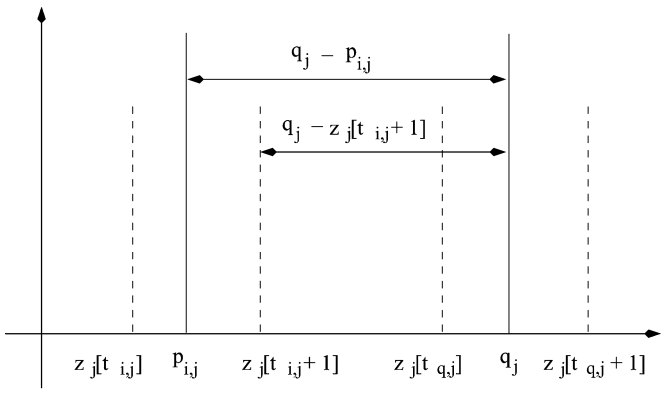

Fig. 7. Illustration of the case in which $t_{i,j} < t_{q,j}$ and $\lambda_j > 0$. Dashed lines represent the boundaries of partitions and solid lines represent the projecting lines of $p_{i,j}$ and q_j, respectively.

Theorem 10: Let u_i be as in (27). Then $\|q - p_i\|^2 \leq u_i$.

Proof: Since $\|q - p_i\|^2 = \Sigma_{j=1}^{m} \lambda_j (q_j - p_{i,j})^2$, it suffices to prove that for each $j, \lambda_j (q_j - p_{i,j})^2 \leq \lambda_j u_{i,j}^2$. Again, there are six cases to examine.

i) $t_{i,j} < t_{q,j}$ and $\lambda_j > 0$. Thus, $q_j > p_{i,j}$. Since $p_{i,j} \geq z_j[t_{i,j}], q_j - p_{i,j} \leq q_j - z_j[t_{i,j}] = u_{i,j}$. Thus, $\lambda_j (q_j - p_{i,j})^2 \leq \lambda_j u_{i,j}^2$.

ii) $t_{i,j} < t_{q,j}$ and $\lambda_j < 0$. In this case, we still have $q_j > p_{i,j}$. Since $p_{i,j} < z_j[t_{i,j} + 1], q_j - p_{i,j} > q_j - z_j[t_{i,j} + 1] = u_{i,j}$. But now $\lambda_j < 0$, and therefore $\lambda_j (q_j - p_{i,j})^2 < \lambda_j u_{i,j}^2$.

iii) $t_{i,j} = t_{q,j}$ and $\lambda_j > 0$. Since $z_j[t_{i,j}] \leq p_{i,j}$ and $q_j < z_j[t_{i,j} + 1], |q_j - p_{i,j}| \leq \max(q_j - z_j[t_{i,j}], z_j[t_{i,j} + 1] - q_j)$. Thus, $(q_j - p_{i,j})^2 \leq u_{i,j}^2$ and therefore $\lambda_j (q_j - p_{i,j})^2 \leq \lambda_j u_{i,j}^2$.

iv) $t_{i,j} = t_{q,j}$ and $\lambda_j < 0$. Thus, $u_{i,j} = 0$. Obviously, $\lambda_j (q_j - p_{i,j})^2 \leq 0$.

v) $t_{i,j} > t_{q,j}$ and $\lambda_j > 0$. Thus, $p_{i,j} > q_j$. Since $p_{i,j} < z_j[t_{i,j} + 1], p_{i,j} - q_j < z_j[t_{i,j} + 1] - q_j = u_{i,j}$. Thus, $\lambda_j (q_j - p_{i,j})^2 < \lambda_j u_{i,j}^2$.

vi) $t_{i,j} > t_{q,j}$ and $\lambda_j < 0$. In this case, we still have $p_{i,j} > q_j$. Since $p_{i,j} \geq z_j[t_{i,j}], p_{i,j} - q_j \geq z_j[t_{i,j}] - q_j = u_{i,j}$. Thus, $\lambda_j (q_j - p_{i,j})^2 \leq \lambda_j u_{i,j}^2$. □

Since q and p_i are the images of Q and O_i, respectively, in the vector space $R^m, \|q - p_i\|$ approximates the real distance $d(Q, O_i)$, or simply $d_{q,i}$, between Q and O_i. Assuming $\|q - p_i\| = d_{q,i}$, we develop algorithms to process the distance-based queries mentioned in Section I. Specifically, for the ϵ-range query [6], whose goal is to find those objects that are within distance ϵ of the query object Q, our algorithm works as follows. We prune those objects O_i's in all the partitions that satisfy $l_i > \epsilon^2$ because these objects are farther away from the query object Q. To see this, notice that $d_{q,i}^2 \geq \|q - p_i\|^2 \geq l_i > \epsilon^2$. For the remaining objects O's, we verify them by testing whether $d(Q, O) \leq \epsilon$.[1] On the other hand, for the nearest-neighbor query (also called the best-match query) [29], we want to find those objects that are closest to Q. Fig. 8 presents the algorithm, whose correctness is shown in Theorem 11.

[1]Based on the assumption $\|q - p_i\| = d_{q,i}$, our algorithm would achieve a recall of 100%. In practice, due to the accumulating errors arising in distance estimation and object embedding (cf. Theorems 2–6), the recall is actually slightly less than 100%, as our experimental results show later. This holds for the nearest-neighbor search as well.

Procedure Find_Nearest_Neighbors
Input: a database of objects $\mathcal{D}$ and their vector representations (images) in the target space R^m, a set of partitions, $PART$, of R^m that contain the images, and a query object Q and its image q in R^m.
Output: the nearest neighbors of Q in $\mathcal{D}$.

```
1. FOUND := ∅;
2. dist := −∞;
3. while PART is not empty do
4.   begin
5.    locate the partition PT in PART that is
      closest to the image q of Q;
6.    remove PT from PART;
7.    while PT is not empty do
8.     begin
9.      pick an object O whose image p is in PT;
10.     remove p from PT;
11.     /* d(Q,O) is the real distance between Q and O.*/
12.     if d(Q,O) = dist then
13.      add O to FOUND;
14.     if d(Q,O) < dist then
15.      begin
16.       dist := d(Q,O);
17.       replace the objects in FOUND by O;
18.       remove partitions PT_i from PART where l_i > dist;
19.      end;
20.    end;
21.  end;
```

Fig. 8. Algorithm for finding the nearest neighbors of the query object Q.

Theorem 11: Based on the assumption $\|q - p_i\| = d_{q,i}$, algorithm Find_Nearest_Neighbors correctly finds all the nearest neighbors of the query object Q.

Proof: Notice that the algorithm maintains a set *FOUND* containing currently found nearest neighbors with the same distance dist to the query object Q. Whenever a closer object O of Q is found, the set is updated to contain only the object O and dist is reset as the real distance between Q and $O, d(Q, O)$. In the meantime, the remaining partitions in *PART* are pruned based on the lower bound l_i. According to Theorem 9, every image (object) p_i in a pruned partition PT_i has the same l_i, and $\|q - p_i\|^2 \geq l_i >$ dist. Thus, only those objects that are farther away from the current nearest neighbors are pruned. This completes the proof. □

V. Performance Evaluation

A. Data and Parameters

We have conducted a series of experiments to compare the proposed query processing algorithms with MVP-tree and M-tree. The algorithms and data structures were implemented in the C programming language under Unix running on a Sun Sparc 20. The data tested included 230 protein sequences and 200 RNA secondary structures. The lengths of the protein sequences ranged from 21 to 2594 amino acids. The distance metric used for the protein sequences was the edit distance for strings [5]. The RNA secondary structures were created by first choosing two phylogenetically related mRNA sequences, rhino 14 and cox5, from GenBank pertaining to the human rhinovirus and coxsackievirus. The $5'$ noncoding region of each sequence was folded and 100 secondary structures of that sequence were collected. The structures were then transformed into trees and their pairwise distances were calculated as described in [30].

The trees had between 70 and 180 nodes. The distance metrics used here satisfy the triangle inequality, but are not Euclidean.

As in [3], we evaluated the performance of the studied techniques by considering distance calculations occurring in the on-line search phase. Specifically, the cost measure used was the average percentage of distance calculations. In each run, a different object was chosen as the query object and the number of distance calculations was divided by the number of objects in a dataset and then multiplied by 100%. The average percentage was calculated over all runs.

We studied two types of similarity search: the ϵ-range query and nearest-neighbor query. To make the ϵ range more meaningful and the two datasets more comparable, we scaled the distances by dividing or multiplying them by a constant. For the protein sequences whose distances ranged from 1 to 2573, we divided the distances by 10. For the RNA secondary structures whose distances ranged from 1 to 89, we multiplied the distances by 10.

We tuned the parameter values to get the best results for each technique. We first considered the 230 protein sequences. It was observed that M-tree achieved the best results when the node size was 10. For MVP-tree, the results were best when the size of the leaf nodes was 10. Since *MetricMap* needs to embed the query object into the target space, there is always an initial cost, which is equal to the number of dimensions of the target space. When the dimensionality is low, this initial cost is low. However, that may cause the distances to be underestimated and may yield a large number of false positives, which have to be verified later on.[2] On the other hand, a higher dimensionality causes a higher initial cost while reducing the number of false positives. The best results of *MetricMap* occurred when the dimensionality of the target space was between 15 and 25. Thus, we set the dimensionality of the target space to 20. We set the size of the sampling set used in the *MetricMap* algorithm to be the number of vantage points in MVP-tree.

B. Performance on Query Processing

We first present the results for the ϵ-range query. Fig. 9 graphs the performance of the three studied techniques for the protein sequences. In the figure, X-axis represents the ϵ values, and Y-axis represents the average percentages of on-line distance calculations needed to answer the ϵ-range query. From the figure, we see that MVP-tree consistently outperforms M-tree, while *MetricMap* beats both of them. Since *MetricMap* is an approximate model, it does not guarantee a 100% recall. Fig. 10 illustrates the recall as a function of ϵ values for *MetricMap*. We can see that *MetricMap* achieves a recall of over 96%. Speeding up a search may be more important than achieving a 100% recall in some applications, a philosophy adopted in many of today's search engines. This philosophy holds particular force for similarity search since a distance measure itself embodies the notion of approximation.

Using the same parameter settings, we conducted experiments on another group of 400 protein sequences pertaining to the human immunodeficiency virus obtained from the database maintained at the National Center for Biotechnology Information. Figs. 11 and 12 show the results. Since the algorithms for all the three studied techniques conduct distance verification when getting the result of a query (cf. Steps 12 and 14 in Fig. 8), their precisions are all 100%.

Next, we considered the 200 RNA secondary structures for the ϵ-range query. The parameter values used for each of the three studied techniques were as before. That is, the node size for M-tree was 10, the size of the leaf nodes of MVP-tree was

[2]False hits (false positives) occur when an object, which should not be in the result of a query, is included by a search algorithm. False dismissals (false negatives) occur when an object, which should be in the result of a query, is excluded by the search algorithm.

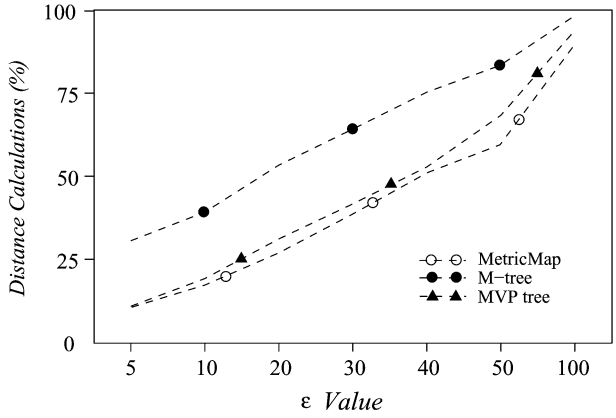

Fig. 9. Distance calculations as a function of the ϵ values for the 230 protein sequences.

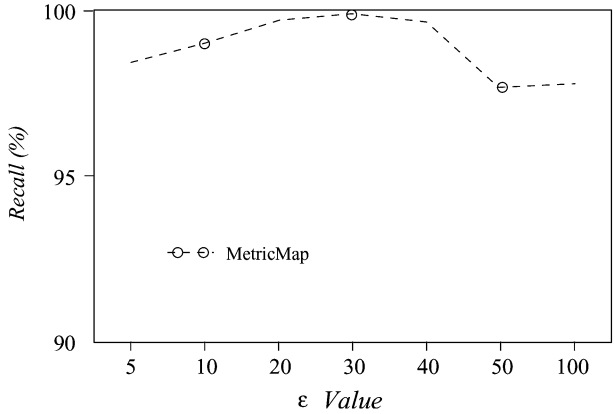

Fig. 10. Recall of *MetricMap* as a function of the ϵ values for the 230 protein sequences.

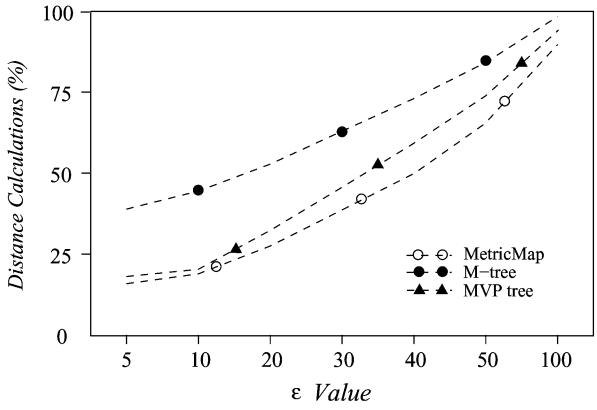

Fig. 11. Distance calculations as a function of the ϵ values for the 400 protein sequences pertaining to the human immunodeficiency virus.

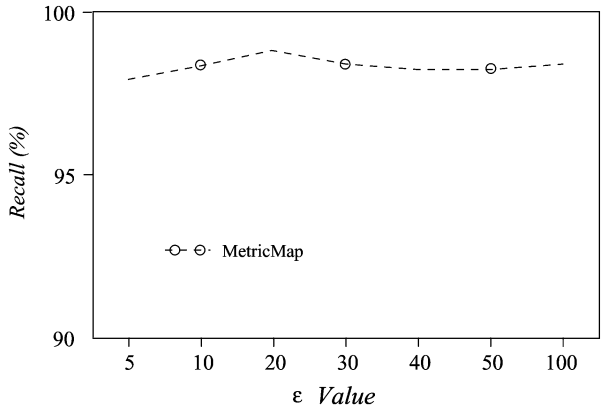

Fig. 12. Recall of *MetricMap* as a function of the ϵ values for the 400 protein sequences pertaining to the human immunodeficiency virus.

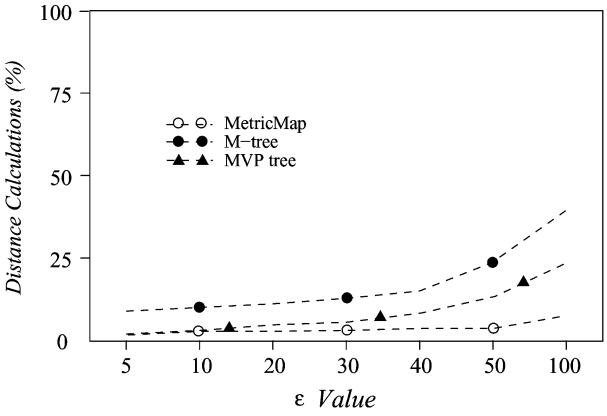

Fig. 13. Distance calculations as a function of the ϵ values for the 200 RNA secondary structures.

10, and the dimensionality of the target space of *MetricMap* was 20. Fig. 13 shows the results. Because of the distance distribution of the RNA data, all the three studied techniques have improved performance. Again, MVP-tree and *MetricMap* outperform M-tree consistently. When ϵ is small, e.g., 5 and 10, MVP-tree performs almost as well as *MetricMap*. Notice that the performance of MVP-tree degrades more quickly than *MetricMap* as the ϵ value increases. When $\epsilon > 12$, *MetricMap* outperforms MVP-tree. Fig. 14 depicts the recall of *MetricMap* as a function of the ϵ values for the RNA secondary structures. From the figure, we see that the recall of *MetricMap* is over 98%.

We also compared the performance of the studied techniques in processing the nearest-neighbor query. Table I shows the results. It is interesting to observe that M-tree beats MVP-tree in nearest-neighbor search, while MVP-tree outperforms M-tree in ϵ-range search.

Remark 2: The experimental results presented here showing *MetricMap* can not achieve a recall of 100% are consistent with the arguments made in [28]. With sampling and dimensionality reduction, Hjaltason and Samet [28] showed that *MetricMap* is not contractive. (An embedding F is contractive if the distances in the target space lower-bound the corresponding distances in the original database.) Consequently, its recall is not 100%, as confirmed by our experimental results. However, without sampling, *MetricMap* is contractive in a Euclidean space for the

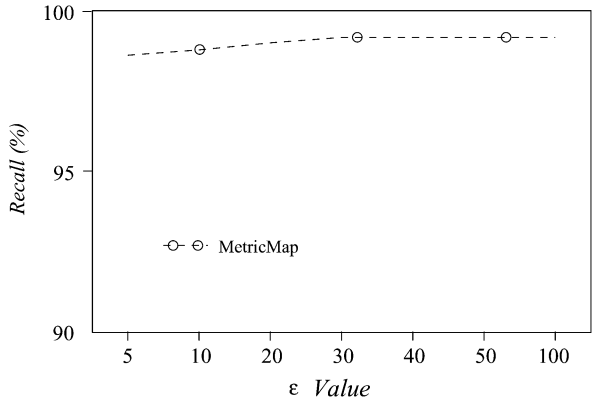

Fig. 14. Recall of *MetricMap* as a function of the ϵ values for the 200 RNA secondary structures.

TABLE I
DISTANCE CALCULATIONS (%) NEEDED BY THE TECHNIQUES

Technique	230 protein sequences	200 RNA structures
M-tree	15.90%	8.45%
MVP-tree	43.62%	18.33%
$MetricMap$	8.70%	2.24%

reference objects and objects that are embeddable to the target space. In other words, if the objects are points in $R^h, h \geq k$, and the distance function d is Euclidean, then *MetricMap* guarantees a lower bound on inter-object distances. That is, $\|p_i - p_j\| \leq d(O_i, O_j)$ where p_i and p_j are images of O_i and O_j, respectively. To see this, notice that in a Euclidean space, the bilinear form ψ is positive definite, because for any nonzero vector $x, x^T M(\psi_{\langle a \rangle})x$ is positive [31]. This implies that all the nonzero eigenvalues are positive. When projecting the points from R^h onto R^k, the images have fewer coordinates. From (13), we conclude that the dissimilarity between two images is less than or equal to the distance between the corresponding objects. This is true for both the reference objects and objects that are embeddable to the target space.

VI. CONCLUSION

We have presented algorithms for processing two types of distance-based queries, namely the nearest neighbor query and the ϵ-range query, using *MetricMap* and VA-file techniques. In our previous work [7]–[9], we briefly introduced *MetricMap* and compared it with *FastMap* [6] in data mining and clustering applications. The new results presented in this paper include: 1) the theoretical foundation for *MetricMap*; 2) the algorithms for processing the distance-based queries in metric spaces; and 3) an empirical study to compare *MetricMap* with MVP-tree and M-tree on both protein and RNA data. Our experimental results indicated that *MetricMap* is an effective technique, which is competitive and sometimes better than the distance-based data structures, so may be a worthwhile component of any database and data mining system for metric spaces. We have implemented *MetricMap* into a software package, which is accessible at http://www.cis.njit.edu/~jason/metricmap.html and can be obtained from the authors.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their thoughtful comments that helped to improve both the presentation and the content of this paper. They also thank Y. Yang for his contributions in the early stage of this work and B. Shapiro of the National Cancer Institute for providing the RNA data used in the experiments.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Reading, MA: Addison-Wesley, 1999.
[2] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann, 2000.
[3] T. Bozkaya and M. Ozsoyoglu, "Indexing large metric spaces for similarity search queries," *ACM Trans. Database Syst.*, vol. 24, no. 3, pp. 361–404, 1999.
[4] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proc. 23rd Int. Conf. Very Large Data Bases*, 1997, pp. 426–435.
[5] J. T. L. Wang, M. J. Zaki, H. T. T. Toivonen, and D. Shasha, Eds., *Data Mining in Bioinformatics*. London, U.K.: Springer, 2005.
[6] C. Faloutsos and K.-I. Lin, "*FastMap*: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 1995, pp. 163–174.
[7] Y. Yang, K. Zhang, X. Wang, J. T. L. Wang, and D. Shasha, "An approximate oracle for distance in metric spaces," in *Combinatorial Pattern Matching*, M. Farach-Colton, Ed. New York: Springer-Verlag, 1998, vol. 1448, pp. 104–117. Lectures Notes in Computer Science.
[8] J. T. L. Wang, X. Wang, K.-I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang, "Evaluating a class of distance-mapping algorithms for data mining and clustering," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 1999, pp. 307–311.
[9] X. Wang, J. T. L. Wang, K.-I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang, "An index structure for data mining and clustering," *Knowl. Inform. Syst.*, vol. 2, no. 2, pp. 161–184, 2000.
[10] R. Weber, H. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proc. 24th Int. Conf. Very Large Data Bases*, 1998, pp. 194–205.
[11] T. Chiueh, "Content-based image indexing," in *Proc. 20th Int. Conf. Very Large Data Bases*, 1994, pp. 582–593.
[12] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 1992, pp. 311–321.
[13] G. R. Hjaltason and H. Samet, "Distance browsing in spatial databases," *ACM Trans. Database Syst.*, vol. 24, no. 2, pp. 265–318, 1999.
[14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.
[15] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," in *Proc. Nat. Acad. Sci.*, vol. 100, 2003, pp. 5591–5596.
[16] Z. Zhang, J. T. Kwok, and D.-Y. Yeung, "Parametric distance metric learning with label information," in *Proc. 18th Int. Joint Conf. Artificial Intelligence*, 2003, pp. 1450–1452.
[17] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, vol. 14.
[18] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann, "Optimal cluster preserving embedding of nonmetric proximity data," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1540–1551, Dec. 2003.
[19] D. K. Agrafiotis, "Stochastic proximity embedding," *J. Computat. Chem.*, vol. 24, pp. 1215–1221, 2003.
[20] P. Courrieu, "Straight monotonic embedding of data sets in Euclidean spaces," *Neural Netw.*, vol. 15, no. 10, pp. 1185–1196, 2002.
[21] M. Quist and G. Yona, "Distributional scaling: An algorithm for structure-preserving embedding of metric and nonmetric spaces," *J. Mach. Learn. Res.*, vol. 5, pp. 399–420, 2004.
[22] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "Boostmap: A method for efficiently approximating similarity rankings," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Washington, DC, Jun. 2004.
[23] S. Dubnov, R. El-Yaniv, Y. Gdalyahu, E. Schneidman, N. Tishby, and G. Yona, "A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles," *Mach. Learn.*, vol. 47, no. 1, pp. 35–61, 2002.
[24] J. L. Kelley, *General Topology*. Princeton, NJ: Van Nostrand, 1955.
[25] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
[26] P. D. Lax, *Linear Algebra*. New York: Wiley, 1997.
[27] W. Greub, *Linear Algebra*. New York: Springer-Verlag, 1975.
[28] G. R. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 5, pp. 530–549, May 2003.
[29] D. Shasha and T. L. Wang, "New techniques for best-match retrieval," *ACM Tran. Inform. Syst.*, vol. 8, no. 2, pp. 140–158, 1990.
[30] B. A. Shapiro and K. Zhang, "Comparing multiple RNA secondary structures using tree comparisons," *Comput. Applic. in Biosci.*, vol. 6, no. 4, pp. 309–318, 1990.
[31] J. M. Ortega, *Matrix Theory*. New York: Plenum, 1987.

Jason T. L. Wang (S'88–M'90) received the B.S. degree in mathematics from National Taiwan University, Taipei, Taiwan, R.O.C., and the Ph.D. degree in computer science from the Courant Institute of Mathematical Sciences, New York University, in 1991.

He is currently a Professor of Computer Science in the College of Computing Sciences, New Jersey Institute of Technology (NJIT), Newark, and Director of NJIT's Data and Knowledge Engineering Laboratory. His research interests include data mining and databases, pattern recognition, bioinformatics, security, and Web information retrieval. He has published over 100 refereed papers and presented numerous software demos in these areas. He is a coauthor of the book *Mining the World Wide Web: An Information Search Approach* (Boston, MA: Kluwer, 2001), and an editor and author of three books, *Pattern Discovery in Biomolecular Data: Tools, Techniques and Applications* (New York: Oxford Univ. Press, 1999), *Computational Biology and Genome Informatics* (Singapore: World Scientific, 2003), and *Data Mining in Bioinformatics* (London, U.K.: Springer, 2005).

Dr. Wang is on the editorial boards of four journal,s including *Information Systems*, *Knowledge and Information Systems*, *Intelligent Data Analysis* and *Pattern Recognition*, and has served on the program committees of over 50 national and international conferences.

Xiong Wang received the B.S. degree in mathematics from Xiamen University, China, the M.S. degree in computer science from Fudan University, China, and the Ph.D. degree in computer and information science from New Jersey Institute of Technology, Newark.

He is an Assistant Professor in the Computer Science Department, California State University at Fullerton. His research interests include databases, knowledge discovery and data mining, pattern matching, and bioinformatics.

Dr. Wang is a member of ACM, ACM SIGMOD, and the IEEE Computer Society.

Dennis Shasha is a Professor of Computer Science at the Courant Institute, New York University, where he works with biologists on pattern discovery for microarrays, combinatorial design, and network inference, and with physicists and financial people on algorithms for time series. His other areas of interest include database tuning, tree and graph matching, and cryptographic file systems. He has written four books of puzzles, a biography about great computer scientists, and technical books about database tuning, biological pattern recognition and time series. He also writes the puzzle columns for *Scientific American* and *Dr. Dobb's Journal*.

Kaizhong Zhang received the M.S. degree in mathematics from Peking University, Beijing, China, in 1981, and the M.S. and Ph.D. degrees in computer science from the Courant Institute of Mathematical Sciences, New York University, in 1986 and 1989, respectively.

He is a Professor in the Department of Computer Science, University of Western Ontario, London, ON, Canada. His research interests include computational biology, pattern recognition, image processing, and sequential and parallel algorithms.