Fast Distance Metrics in Low-dimensional Space for Neighbor Search Problems

Guihong Wan*, Crystal Maung†, Chenxu Zhang* and Haim Schweitzer*
*Department of Computer Science, The University of Texas at Dallas, Richardson, Texas
† 7-Eleven Inc., Irving, Texas
Email:{Guihong.Wan, Chenxu.Zhang, HSchweitzer}@utdallas.edu, Crystal.Maung2@7-11.com

***Abstract*—We consider popular dimension reduction techniques that project data on a low dimensional subspace. They include Principal Component Analysis, Column Subset Selection, and Johnson-Lindenstrauss projections. These techniques have been classically used to efficiently compute various approximations. We propose the following three-step procedure for enhancing the accuracy of such approximations: 1. Unknown quantities in the approximation are replaced with random variables. 2. The Maximum Entropy method is applied to infer the most likely probability distribution. 3. Expected values of the random variables are used to compute the enhanced estimates. Our use of the Maximum Entropy method requires knowledge of vector norms that can be easily computed during the dimension reduction. We demonstrate significant enhancements in average accuracy for Euclidean distance and Mahalanobis distance, and improvements in evaluating k-nearest neighbors and k-furthest neighbors by using the enhanced Euclidean distance formula.**

***Keywords*-dimension reduction, maximum entropy method, Euclidean distance, Mahalanobis distance, k-nearest neighbors, k-furthest neighbors**

I. Introduction

Let A be a data matrix of dimensions $m\times n$. Its columns $(a_1,\ldots,a_n)$ are n data items. The goal of dimension reduction is to reduce the dimension of each data item from m to r, where $r<m$. The most common techniques are linear, and can be described as follows. Let Q be a matrix of size $m\times r$ with orthogonal columns. Let w_i be the r-dimensional reduction of the m-dimensional column a_i, and let W be the $r\times n$ reduction of the entire matrix A. Then:

$$w_i = Q^T a_i, \qquad W = Q^T A. \tag{1}$$

Three common choices for the matrix Q that we discuss in detail are the r dominant left eigenvectors of A, r judiciously selected columns of A, and r vectors with random coordinates drawn from a Gaussian distribution with orthogonalization. The first choice gives the Principal Component Analysis (PCA) technique; the second choice is known as Column Subset Selection (CSS); the third choice gives a Johnson-Lindenstrauss (JL) random projections. One of the main reasons for performing dimension reduction is that the complexity of calculating important functions of the data is reduced from dependency on m to dependency on r. Several cases are shown in Table I. We denote a m-dimensional vector by x or y.

	exact function	classical approximation
Euclidean distance: complexity:	$\|x-y\|^2$ $O(m)$	$\|w_x-w_y\|^2$ $O(r)$
Correlation matrix B: Covariance matrix C: complexity:	$B=AA^T$ $C=A_cA_c^T$ $O(m^2n)$	QWW^TQ^T $QW_cW_c^TQ^T$ $O(r^2n)$
Inverse of C: complexity:	C^{-1} $O(m^3)$	$Q(W_cW_c^T)^{-1}Q^T$ $O(r^3)$
Mahalanobis distance: complexity per vector:	$x_c^TC^{-1}x_c$ $O(m^2)$	$w_x^T(W_cW_c^T)^{-1}w_x$ $O(r^2)$

Table I: Classical approximations using dimension reduction. $w_x{=}Q^Tx$, $w_y{=}Q^Ty$, $W{=}Q^TA$. x_c, A_c are obtained by subtracting the mean μ from each vector. $W_c{=}Q^TA_c$.

In this paper we derive new approximation formulas for the functions listed in Table I. The new formulas are more accurate "on average", with the approximately same running time. For example, we derive the following approximation formula for the distance between a_i and a_j:
$\|a_i{-}a_j\|^2 \approx \|w_i{-}w_j\|^2 + \|a_i\|^2 - \|w_i\|^2 + \|a_j\|^2 - \|w_j\|^2$.
Since that after dimension reduction w_i, w_j, $\|a_i\|^2{-}\|w_i\|^2$ and $\|a_j\|^2{-}\|w_j\|^2$ are known the complexity of the new formula is $O(r)$. We observe that it is not immediately clear that this formula is better than the classical formula shown in Table I. For example, when $a_i{\approx}a_j$ the classical formula gives approximately 0, the exact answer, while our approximation formula gives approximately $2(\|a_i\|^2{-}\|w_i\|^2)$ which is usually nonzero. However, we claim that our new formula is better "on the average". This follows from the derivation that uses the Maximum Entropy method (MEM) (see Section IV), and was confirmed by extensive experimental evaluation on real datasets (see Section VII).

Our approach of using the MEM requires computing the norms of all the column vectors of the data matrix, as discussed in Section IV. The derivation of the explicit approximation formulas is carried out in Section V. The neighbor search problems are discussed in VI.

Main contributions: The first contribution is the derivation of explicit formulas that "on the average" give better approximations to the exact functions in Table I than the classical approximations. The second is the technique used to derive these formulas. The idea of using the Maximum Entropy method in this context appears to be novel. After dimension reduction the following information is available: i. the $r\times n$ matrix W and its columns w_i, as defined in (1).

ii. $z_i = \|a_i\|^2 - \|w_i\|^2$, or $z_x = \|x\|^2 - \|w_x\|^2$ where x is not (necessarily) a column of A. We denote the true distance by the subscript $_{\text{exact}}$, the classical approximation formula by the subscript $_{\text{classical}}$, and our improved formula by the subscript $_{\text{entropy}}$:

Squared Euclidean distance: $d_{\text{exact}} = \|x-y\|^2$
$d_{\text{classical}} = \|w_x - w_y\|^2, \quad d_{\text{entropy}} = d_{\text{classical}} + z_x + z_y, \quad x \neq y.$
Mahalanobis distance: $\text{md}_{\text{exact}} = x_c^T C^{-1} x_c$
$\text{md}_{\text{classical}} = w_x^T (W_c W_c^T)^{-1} w_x$ (this is a pseudo-inverse),
$\text{md}_{\text{entropy}} = \text{md}_{\text{classical}} + z_x/\delta$, where $\delta = (\sum_{i=1}^{n} z_i)/(m-r)$.

This generalizes the Euclidean distance results of [1].

II. Dimension reduction techniques

We discuss three dimension reduction techniques that reduce the dimension by linear projections on a low dimensional subspace. Let Q be $m \times r$, with orthogonal columns that span the desired low dimensional subspace.

Given an m-dimensional vector x we denote its r dimensional projection by w_x. Uncentered dimension reduction and reconstruction is computed as follows:

$$w_x = Q^T x, \quad x \approx Q w_x.$$

The centered variants use a mean vector μ that is viewed as the "center". Centered dimension reduction and reconstruction are then computed as follows:

$$w_x = Q^T(x-\mu), \quad x \approx \mu + Q w_x.$$

In both cases we refer to w_x as the low dimensional representation of x, and to the approximations on the right hand side as the reconstruction of x.

Principal Component Analysis (PCA): The first method that we describe is the PCA, which is arguably the most common dimension reduction technique. Given an $m \times n$ matrix A the PCA computes Q as the matrix that minimizes the *average* squared reconstruction error of all columns:

$$e_{\text{PCA}}(A) = \frac{1}{n} \sum_i \|a_i - Q w_i\|^2, \quad w_i = Q^T a_i.$$

It is known (e.g., [2]) that the matrix Q consisting of the r eigenvectors with the largest eigenvalues of the data covariance matrix minimizes e_{PCA}.

Column Subset Selection (CSS): The main idea behind the CSS is to compute Q from a subset of data columns. The error criterion is again the average reconstruction error as shown below:

$$e_{\text{CSS}}(A) = \frac{1}{n} \sum_i \|a_i - S\tilde{w}_i\|^2 = \frac{1}{n} \sum_i \|a_i - Q w_i\|^2$$

where S is a subset of r columns from A, $\tilde{w}_i = S^+ a_i$, Q is an orthogonal basis of S, and $w_i = Q^T a_i$. An approximate solution can be computed from the pivoted QR factorization [3], an algorithm that we call QRP. Another algorithm that is typically called GKS applies the QRP on the top r right eigenvectors of A [3]. For algorithms that perform the selection at random according to special probabilities see, e.g., [4].

Johnson-Lindenstrauss projections (JL): In the case of JL projections the matrix Q is selected at random, and does not use any property of the matrix A except for n, the number of columns. For any set of n vectors $x_1, \ldots, x_n$ and an $m \times r$ orthogonal matrix Q define: $e_{\text{JL}} = \max_i \mid \frac{\|Q^T x_i\|^2}{\|x_i\|^2} - 1 \mid$. Thus, small values of e_{JL} mean that the projections on Q approximately preserve the norms of all x_i. When Q is generated at random its quality can be quantified by the following criterion: $\text{Prob}(e_{\text{JL}} > \epsilon) < \delta$. This means that with confidence (probability) of at least $1-\delta$ the e_{JL} error is at most ϵ. The value of r necessary to guarantee this condition depends on ϵ, δ, and n, but surprisingly not on m. For the case in which Q is selected from a Gaussian distribution (and then normalized) the value of r is known to be $O(\frac{\log(n)\log(1/\delta)}{\epsilon^2})$. See, e.g., [5], [6].

Discussion: Observe that the PCA and the CSS minimize an average error while the JL minimizes a worst case error. Thus, the PCA always has the best average reconstruction error. There are several reasons that CSS is sometimes preferred. For example the selected columns are considered "meaningful", and the CSS preserves the sparsity and other characteristics of the data. Unlike the JL, there are no worst-case error guarantees for the PCA and the CSS. Another big advantage of the JL over the PCA and the CSS is that it can be computed much faster. Other sparser variants of the JL have been recently developed [7], [8], which enable very fast algorithms to many problems in numerical linear algebra.

III. Random variable dimension reduction

In this section we refer to the matrix Q of Section II as Q_1. Its size is $m \times r$. The dimension reduction is performed by projecting vectors on its column subspace. Let Q_2 of size $m \times (m-r)$ be an orthogonal complement of Q_1. Then the following relations hold:

$$\begin{aligned} &Q_1^T Q_1 = I, \quad Q_2^T Q_2 = I, \quad Q_1 Q_1^T + Q_2 Q_2^T = I \\ &A = Q_1 W_1 + Q_2 W_2, \quad a_i = Q_1 w_1^i + Q_2 w_2^i. \end{aligned} \tag{2}$$

Here $W_1 = Q_1^T A$, a_i is the ith column of A, w_1^i is the ith column of W_1, and w_2^i is the ith column of W_2. Observe that the three identity matrices in the first line have different dimensions: $r \times r$, $(m-r) \times (m-r)$, and $m \times m$, respectively. In the setting of classical linear dimension reduction we know Q_1 and W_1. After dimension reduction A is discarded and W_1 represents A in the r-dimensional space. The reconstruction can be computed as follows:

$$A \approx Q_1 W_1, \quad a_i \approx Q_1 w_1^i.$$

In our setting, the goal is to better represent A and a_i after the dimension reduction. Without loss of generality we can take Q_2 to be any orthogonal complement of Q_1. Therefore, the only unknown quantities in (2) are the entries of the matrix W_2. Here is our key point. Classical dimension reduction takes $W_2 = 0$. Instead, we propose to view W_2 as a random matrix, with entries that are random variables. Clearly, if W_2 is a random matrix then so is A, and so are the

columns w_2^i and a_i. Equation (3) identifies random variables with $\widehat{\ }$ as shown below:

$$\widehat{A} = Q_1 W_1 + Q_2 \widehat{W}_2, \quad \widehat{a}_i = Q_1 w_1^i + Q_2 \widehat{w}_2^i. \qquad (3)$$

We use the Maximum Entropy method to compute the most likely probability distribution of W_2 under the constraint that the column norms of A are known.

IV. The Maximum Entropy method

The Maximum Entropy method is a well known technique for inferring probability distributions. When given constraints that the probability distribution must satisfy, the Maximum Entropy method asserts that "the most likely distribution" is the distribution with the maximum entropy that satisfies the constraints. See, e.g., [9], [10]. In this paper we use a special case described in Chapter 14 of [10].

Theorem 1: Let $x = (x_1, \ldots, x_n)^T$ be a random vector, where the coordinates x_i are n random variables. Let $R = E\{xx^T\}$ be the correlation matrix associated with x. Let Δ be the determinant of R and assume $\Delta \neq 0$. Then according to the MEM the density $f(x)$ is normal with zero mean, and $H(x)$ is the corresponding entropy:

$$f(x) = \frac{1}{\sqrt{(2\pi)^n \Delta}} e^{-\frac{1}{2} x^T R^{-1} x}$$
$$H(x) = \ln \sqrt{(2\pi e)^n \Delta}. \qquad (4)$$

See [10] for the proof. From (4) it follows that for a fixed n the entropy of x is determined by Δ, the determinant of the correlation matrix R. As observed by Papoulis [10], [11] if R is only partially known (like only the diagonal elements of R are given), the missing parts can be determined according to the MEM by maximizing the determinant Δ over the unknown parts. We use this technique to derive Theorem 2.

Theorem 2: Let $W = (w_1, \ldots, w_n)$ be a random matrix of dimensions $r \times n$. Suppose $z_i = E\{\|w_i\|^2\}$ is known for $i = 1, \ldots, n$, but nothing else is known about the probability density of W. Then according to the MEM:

1. All entries of the matrix W have zero mean:
$E\{w_{ij}\}=0, \quad \text{for } i = 1, \ldots, n,\ j = 1, \ldots, r.$

2. The random variable w_{i_1,j_1} is independent of the random variable w_{i_2,j_2} unless $i_1 = i_2$ and $j_1 = j_2$.

3. The expected value of w_{ij}^2 is given by: $E\{w_{ij}^2\} = z_i/r$.

4. The probability density of W is given by:

$$f(W) = \frac{1}{\sqrt{(2\pi)^{rn} \Delta}} e^{-s(W)}$$

where: $\Delta = \frac{\prod_{i=1}^n z_i^r}{r^{rn}}$, $s(W) = \frac{r}{2} \sum_{i,j} \frac{w_{i,j}^2}{z_i}$.

Proof: Entries of W are notated in the following way:

$$W = \begin{pmatrix} w_{11} & w_{21} & \ldots & w_{n1} \\ w_{12} & w_{22} & \ldots & w_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ w_{1r} & w_{2r} & \ldots & w_{nr} \end{pmatrix}$$

$w_i = (w_{i1}, w_{i2}, \ldots, w_{ir})^T$. Let q be the vector of rn random variables, created by concatenating all of W columns:

$$q = \left(w_{11} \ldots w_{1r}, \ldots, w_{n1} \ldots w_{nr}\right)^T.$$

Let R be the correlation matrix of q: $R = E\{qq^T\}$. Observe that the matrix R is $nr \times nr$. The value of R at row I and column J is: $R_{IJ} = E\{w_{i_1,j_1} w_{i_2,j_2}\}$ for some i_1, j_1, i_2, j_2. Define: $\nu_{ij} = E\{w_{ij}^2\}$. Then from the definition of z_i:

$$z_i = E\{\|w_i\|^2\} = \sum_j E\{w_{ij}^2\} = \sum_j \nu_{ij}. \qquad (5)$$

The following is known about the diagonal elements R_{II}, where the location I in q correspond to the location i, j in W: $R_{II} = E\{w_{ij}^2\} = \nu_{ij}$. From Theorem 1 it follows that the maximum entropy of q is obtained by maximizing Δ, the determinant of R, under the constraints (5). The proof of the theorem follows from this maximization.

According to the Hadamard determinant inequality (see, e.g., [12]), $\Delta \leq \prod_I R_{II}$. Since $\Delta = \prod_I R_{II}$ if all off diagonal elements are 0, it follows that there is a maximum where $E\{w_{i_1,j_1} w_{i_2,j_2}\} = 0$ unless $i_1{=}i_2$ and $j_1{=}j_2$. This proves parts 1 and 2 of the theorem. To prove part 3 observe that from parts 1,2 it follows that

$$\Delta = \prod_I R_{II} = \prod_{i,j} \nu_{ij} = \prod_i (\prod_j \nu_{ij}).$$

Therefore, for each i we need to maximize $\prod_{i,j} \nu_{ij}$ subject to the constraint that $\sum_{j=1}^r \nu_{ij} = z_i$. Thus, we need to maximize the product of variables given their sum. It can be easily shown (for example using the method of Lagrange multipliers) that the maximizing solution has all variables identical, which implies that $\nu_{ij} = \frac{z_i}{r}$. This proves part 3 of the theorem. Part 4 follows from Theorem 1 by observing:

- From part 3: $\Delta = \prod_{i=1}^n \prod_{j=1}^r (z_i/r) = \frac{\prod_{i=1}^n z_i^r}{r^{rn}}$.
- Since R is diagonal: $q^T R^{-1} q = \sum_I \frac{q_I^2}{R_{II}} = \sum_{i,j} \frac{w_{i,j}^2}{z_i/r}$.

This completes the proof of Theorem 2. ■

V. Derivation of explicit formulas

In this section we derive the formulas. Dimension reduction is applied to the columns of the matrix A of size $m \times n$, then A is discarded. The only retained information is the $m \times r$ matrix Q_1, the $r \times n$ matrix W_1, and the norms $\|a_i\|, i{=}1, \ldots, n$. Matrices Q_1, W_1 are as defined in (2).

Observe that this information is more than what is retained by classical dimension reduction into r dimensions, but less than what is retained by classical dimension reduction into $r{+}1$ dimensions. As shown later, in some of these cases the improvements we achieve are significantly more than what one can get by going from r to $r{+}1$ dimensions for Q_1.

Repeating (3) we have the following exact representation:

$$\widehat{A} = Q_1 W_1 + Q_2 \widehat{W}_2, \quad \widehat{a}_i = Q_1 w_1^i + Q_2 \widehat{w}_2^i. \qquad (6)$$

Since Q_1, Q_2 are orthogonal it follows from the right-hand side of (6) that $\|\widehat{a}_i\|^2 = \|w_1^i\|^2 + \|\widehat{w}_2^i\|^2$. Since $\|\widehat{a}_i\|^2$ is

known to be $\|a_i\|^2$ we conclude that:

$$\|\widehat{w}_2^i\|^2=\|a_i\|^2-\|w_1^i\|^2,\ z_i = E\{\|\widehat{w}_2^i\|^2\} = \|a_i\|^2-\|w_1^i\|^2.$$

This shows that the condition of Theorem 2 holds for the random matrix $\widehat{W}_2$, and therefore its most likely probability density is given by the theorem.

Notation: For an arbitrary vector x: $w_1^x=Q_1^T x$, $w_2^x=Q_2^T x$, $z_x=\|x\|^2-\|w_1^x\|^2$. The random variable representation of x is: $\widehat{x} = Q_1 w_1^x + Q_2\widehat{w}_2^x$. For the column a_i: $w_1^i=Q_1^T a_i$, $w_2^i=Q_2^T a_i$, $z_i=\|a_i\|^2-\|w_1^i\|^2$. The random variable representation of a_i is: $\widehat{a}_i = Q_1 w_1^i + Q_2\widehat{w}_2^i$.

Expected values: The following expected values related to w_2 and W_2 are obtained from Theorem 2:

$$\begin{aligned}
&E\{\widehat{w}_2^i\} = 0,\quad E\{\|\widehat{w}_2^i\|^2\} = z_i,\quad E\{\|\widehat{w}_2^x\|^2\} = z_x\\
&E\{(\widehat{w}_2^i)^T\widehat{w}_2^j\} = 0 \quad \text{when } i\neq j\\
&E\{(\widehat{w}_2^x)^T\widehat{w}_2^y\} = 0 \quad \text{when } x\neq y\\
&E\{\widehat{w}_2^i(\widehat{w}_2^i)^T\} = \frac{z_i}{m-r}I \quad \text{(sub-diagonal matrix)}
\end{aligned} \tag{7}$$

$$\begin{aligned}
&E\{\widehat{W}_2\} = 0,\quad E\{W_1\widehat{W}_2^T\} = 0,\quad E\{\widehat{W}_2 W_1^T\} = 0,\\
&E\{\widehat{W}_2\widehat{W}_2^T\} = \frac{\sum_{i=1}^n z_i}{m-r}I = \delta I, \text{where } \delta = \frac{\sum_{i=1}^n z_i}{m-r}.
\end{aligned} \tag{8}$$

Euclidean distance: The squared Euclidean distance between the vectors x, y is: $d_{\text{exact}} = \|x-y\|^2$. Using the random variable representation of x, y and after some algebraic manipulation we get:

$$\widehat{d}_{\text{exact}} = \|\widehat{x}-\widehat{y}\|^2 = \|w_1^x - w_1^y\|^2 + \|\widehat{w}_2^x - \widehat{w}_2^y\|^2.$$

Observe that the first term is the classical formula for estimating this distance from low dimensional projections. Taking expectations:

$$\begin{aligned}
&E\{\widehat{d}_{\text{exact}}\} = d_{\text{classical}} + E\{\|\widehat{w}_2^x-\widehat{w}_2^y\|^2\},\\
&E\{\|\widehat{w}_2^x-\widehat{w}_2^y\|^2\} = E\{\|\widehat{w}_2^x\|^2\}+E\{\widehat{w}_2^y\|^2\}-2E\{(\widehat{w}_2^x)^T\widehat{w}_2^y\}\\
&\quad = z_x + z_y \quad \text{when } x\neq y \quad \text{(this follows from (7))}.
\end{aligned}$$

Therefore:

$$d_{\text{entropy}} = E\{\widehat{d}_{\text{exact}}\} = d_{\text{classical}} + z_x + z_y \quad \text{when } x\neq y.$$

Interpretation: It is possible to partially justify the entropy formula without Maximum Entropy arguments. Observe:

$$\begin{aligned}
&\|w_2^x-w_2^y\|^2 = \|w_2^x\|^2 + \|w_2^y\|^2 - 2(w_2^x)^T w_2^y,\\
&-\|w_2^x\|\|w_2^y\| \le (w_2^x)^T w_2^y \le \|w_2^x\|\|w_2^y\|.
\end{aligned}$$

The second line follows from the Cauchy–Schwarz inequality. This gives the following lower and upper bounds:

$$\begin{aligned}
d_{\text{classical}} \le d_{\text{classical}} + z_x + z_y - 2\sqrt{z_x z_y} \le d_{\text{exact}}\\
\le d_{\text{classical}} + z_x + z_y + 2\sqrt{z_x z_y}.
\end{aligned}$$

This shows that d_{entropy} is the middle between a lower and an upper bound on d_{exact}. Clearly, the lower bound is at least as good an estimate as $d_{\text{classical}}$. In particular, the new formula is much more accurate when x, y are orthogonal. Observe that the error of the new formula compared with the true distance is: $|2(w_2^x)^T w_2^y|$, since:

$$d_{\text{exact}} = \|x-y\|^2 = \|w_1^x - w_1^y\|^2 + \|w_2^x - w_2^y\|^2,$$

where $\|w_2^x-w_2^y\|^2 = \|w_2^x\|^2 + \|w_2^y\|^2 - 2(w_2^x)^T w_2^y$. When x, y are orthogonal, the error $|2(w_2^x)^T w_2^y|$ could be small. It is confirmed experimentally.

Correlation matrix and its inverse:

$$\begin{aligned}
&\widehat{B} = \widehat{A}\widehat{A}^T = (Q_1W_1 + Q_2\widehat{W}_2)(Q_1W_1 + Q_2\widehat{W}_2)^T\\
&= Q_1W_1W_1^TQ_1^T + Q_2\widehat{W}_2\widehat{W}_2^TQ_2^T\\
&\quad - Q_1W_1\widehat{W}_2^TQ_2^T - Q_2\widehat{W}_2W_1^TQ_1^T \quad \text{(these become 0)}.
\end{aligned}$$

$$\begin{aligned}
&B_{\text{entropy}} = E\{\widehat{B}\} = Q_1W_1W_1^TQ_1^T + Q_2E\{\widehat{W}_2\widehat{W}_2^T\}Q_2^T\\
&\quad = B_{\text{classical}} + \delta Q_2Q_2^T = B_{\text{classical}} + \delta(I - Q_1Q_1^T).
\end{aligned}$$

$$\text{invB}_{\text{classical}} = Q_1(W_1W_1^T)^{-1}Q_1^T \quad \text{(this is a pseudo-inverse)}.$$

$$\begin{aligned}
\text{invB}_{\text{entropy}} &= \text{invB}_{\text{classical}} + \frac{1}{\delta}Q_2Q_2^T\\
&= \text{invB}_{\text{classical}} + \frac{1}{\delta}(I - Q_1Q_1^T).
\end{aligned}$$

The formulas for Covariance matrix and its inverse are identical to the formulas for the correlation matrix.

Mahalanobis distance: The formulas for Mahalanobis distance follow directly from the formulas for the covariance matrix. With $x_c=x-\mu$, $w_x=Q_1^T x_c$, $W_c=Q_1^T A$:

$$\begin{aligned}
&\text{md}_{\text{exact}} = x_c^T C^{-1} x_c,\\
&\text{md}_{\text{classical}} = x_c^T\text{invC}_{\text{classical}}x_c = w_x^T(W_cW_c^T)^{-1}w_x,\\
&\text{md}_{\text{entropy}}=x_c^T\text{invC}_{\text{entropy}}x_c=\text{md}_{\text{classical}}+\frac{x_c^T(I-Q_1Q_1^T)x_c}{\delta}\\
&\quad = \text{md}_{\text{classical}} + (\|x_c\|^2-\|w_x\|^2)/\delta = \text{md}_{\text{classical}}+z_x/\delta.
\end{aligned}$$

VI. The neighbor search problem

We are interested in two well-known neighbor search problems: the k-nearest neighbors search (KNN) and the k-furthest neighbors search (KFN). The KNN is the optimization problem of finding k points in a given set that are top k closest to a given query point. Opposite to the KNN, the KFN is to find the k furthest points.

There is an extensive literature on the furthest neighbor problem. In [13]–[15], the exact furthest neighbor problem are solved by using Voronoi diagram or tree-based methods. The definition of approximate furthest neighbor problem is given by [16] and the authors introduced an algorithm for the problem relied on the fair spill tree. Recently, three algorithms are proposed to solve the approximate furthest neighbor problem: QDAFN [17], DrusillaSelect [18] and RQALSH [19]. We evaluate the performance of our enhanced formula for these three algorithms using the code provided by the authors.

Complexity: Let n, n_q be the number of points in the data set and in the query set, respectively. Let m be the dimension of the points. The goal is to find k-nearest/furthest neighbors in the given data set for each point in the query set. Consider the linear scan algorithm. Without dimension reduction the complexity is $O(n_q n m)$. With dimension reduction, the dimension of the data points is reduced from m to r. The complexity is $O(n_q n r)$, ignoring the cost for classical dimension

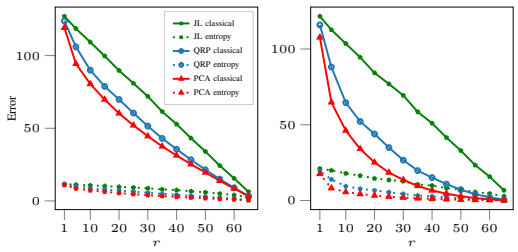

Figure 1: Accuracy for Euclidean distance formulas for various dimension reduction techniques. Left panel: Distances to arbitrary vectors. Right panel: Column distances.

Table II: Accuracy for Euclidean distance using PCA

		distances to arbitrary vectors		column distances	
		$r=5$	$r=25$	$r=5$	$r=25$
CB	$\lvert d_{\text{classical}} - d_{\text{exact}}\rvert$	5.680E+01	3.609E+01	9.602E-01	5.705E-02
	$\lvert d_{\text{entropy}} - d_{\text{exact}}\rvert$	**1.028E+00**	**2.482E-01**	**1.870E-01**	**1.012E-02**
MD	$\lvert d_{\text{classical}} - d_{\text{exact}}\rvert$	3.266E+05	2.843E+05	6.529E+05	5.678E+05
	$\lvert d_{\text{entropy}} - d_{\text{exact}}\rvert$	**9.085E+02**	**8.507E+02**	**3.157E+04**	**2.779E+04**
YP	$\lvert d_{\text{classical}} - d_{\text{exact}}\rvert$	2.265E+06	3.305E+05	4.527E+06	6.532E+05
	$\lvert d_{\text{entropy}} - d_{\text{exact}}\rvert$	**1.935E+03**	**8.130E+02**	**5.452E+05**	**7.569E+04**

reduction. In our setting, besides conducting the classical dimension reduction, we need to compute the norms of data points. The complexity is $O(n_q nr{+}nm{+}n_q m)$. When $n_q r{>}m$ and $nr{>}m$, the complexity is $O(n_q nr)$ same as the classical dimension reduction.

VII. Experimental results and applications

We first show experimental results with Euclidean distance and Mahalanobis distance. Then we demonstrate the performance of entropy formula of Euclidean distance in k-nearest neighbors (KNN) and k-furthest neighbors (KFN). The main observations are as follows:

- The Maximum Entropy formulas produce better results than the classical formulas. The amount of improvement is data dependent.
- For KNN, performing with PCA, the entropy formula gives the better results than the classical formula.
- For KFN, all dimension reduction techniques combined with Maximum Entropy give better estimates than the classical formula.

Results on the following datasets from the UC Irvine repository are shown: Geographical origin of music (GOM), $68{\times}1,059$; ConnectionistBench (CB), $60{\times}208$; Madelon (MD), $500{\times}4,400$; YearPredictionMSD (YP), $90{\times}515,345$; Sift, $128 \times 1,000,000$. All experiments are with centered variant of dimension reduction techniques.

A. Experiments

1) Euclidean distance: We experimented with two cases: a. Evaluating distances between randomly generated vectors and all data columns. The randomly generated vectors are more likely far away from the data columns (tend to be orthogonal to the data columns). b. Evaluating distances between pairs of all columns. The error is the difference between the approximation and the true value.

Table III: Ratio for KFN using JL on various datasets

	$r:k$	QDAFN		Drusilla		RQALSH	
		$d_{\text{classical}}$	d_{entropy}	$d_{\text{classical}}$	d_{entropy}	$d_{\text{classical}}$	d_{entropy}
YP	$5:1$	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	$5:5$	1.0393	**1.0011**	1.0393	**1.0011**	1.0424	**1.0076**
	$5:10$	1.0471	**1.0006**	1.0471	**1.0006**	1.0471	**1.0006**
	$15:1$	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	$15:5$	1.0010	**1.0006**	1.0010	**1.0006**	1.0010	**1.0006**
	$15:10$	1.0008	**1.0003**	1.0008	**1.0003**	1.0008	**1.0003**
Sift	$5:1$	1.1267	**1.0936**	**1.1187**	1.1209	1.1313	**1.1166**
	$5:5$	1.1314	**1.1025**	**1.1172**	1.1193	1.1357	**1.1189**
	$5:10$	1.1323	**1.1052**	1.1204	**1.1177**	1.1383	**1.1223**
	$25:1$	1.0699	**1.0556**	1.0745	**1.0647**	1.0762	**1.0646**
	$25:5$	1.0700	**1.0569**	1.0742	**1.0635**	1.0734	**1.0648**
	$25:10$	1.0694	**1.0578**	1.0738	**1.0647**	1.0730	**1.0660**

Figure 1 shows the average errors in computing these distances for various dimension reduction techniques on GOM dataset. Entropy formulas are plotted by dotted lines and they are clearly superior. Table II shows the error for various datasets. Observe that the distances computed by the entropy formulas are much closer to the true distances. In particular the errors are much smaller when the vectors are randomly generated, which confirmed the argument that the new formula is much more accurate when x, y are orthogonal.

2) Mahalanobis distance: Results for estimating Mahalanobis distances are shown in Figure 2. Various dimension reduction techniques were applied to the GOM dataset. We note that the entropy formulas produced significantly smaller errors. Table V shows the errors on several datasets.

B. Applications

1) k-Nearest Neighbors: We use the Linear Search algorithm for KNN to evaluate the performance of our entropy formula. From the entire dataset, 20 points (100 points for YearPredictionMSD) are randomly selected as the query set and the remaining points are the data set. For each query point, the algorithm identifies the k nearest neighbors from the data set. For the evaluation we use two criteria:

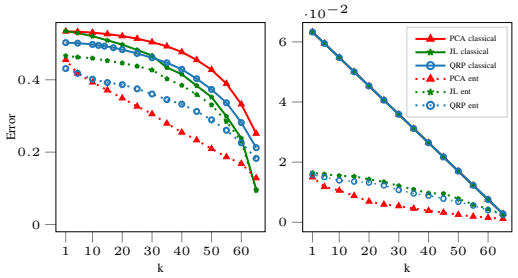

Figure 2: Accuracy for Mahalanobis distance. Left panel: x from arbitrary vectors. Right panel: x from the dataset.

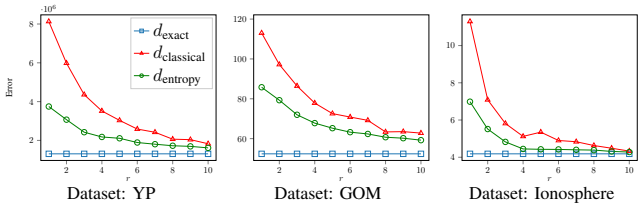

Figure 3: Error of KNN using PCA. $k{=}10$.

methods		PCA				QRP				JL			
		$d_{\text{classical}}$		d_{entropy}		$d_{\text{classical}}$		d_{entropy}		$d_{\text{classical}}$		d_{entropy}	
	$r:k$	ratio	recall	ratio	recall	ratio	recall	ratio	recall	ratio	recall	ratio	recall
Linear Scan	5 : 10	1.0189	76.8%	**1.0019**	**91.4%**	1.0481	62.6%	**1.0056**	**87.2%**	1.1540	39.4%	**1.0252**	**77.4%**
	25 : 10	1.0011	92.8%	**1.0003**	**96.4%**	1.0043	85.0%	**1.0002**	**96.6%**	1.0366	65.6%	**1.0096**	**83.6%**
QDAFN	5 : 10	1.0189	76.8%	**1.0020**	**91.2%**	1.0481	62.6%	**1.0043**	**87.2%**	1.1537	39.4%	**1.0367**	**73.6%**
	25 : 10	1.0011	92.8%	**1.0003**	**96.4%**	1.0043	85.0%	**1.0002**	**96.6%**	1.0366	65.6%	**1.0096**	**83.6%**
Drusilla	5 : 10	1.0185	77.0%	**1.0019**	**91.4%**	1.0481	62.6%	**1.0056**	**87.2%**	1.1494	39.8%	**1.0264**	**77.2%**
	25 : 10	1.0013	92.0%	**1.0006**	**95.4%**	1.0045	84.8%	**1.0006**	**95.6%**	1.0369	65.6%	**1.0161**	**79.8%**
RQALSH	5 : 10	1.0193	76.8%	**1.0064**	**87.2%**	1.0538	60.6%	**1.0302**	**68.8%**	1.1542	39.0%	**1.0590**	**62.6%**
	25 : 10	1.0036	90.0%	**1.0029**	**92.8%**	1.0088	82.4%	**1.0051**	**92.4%**	1.0572	57.6%	**1.0430**	**63.0%**

Table IV: Improvement for KFN on GOM dataset

error and recall. The error is: $(\sum_{i=1}^{k} d_i)/k$. The recall is $[\text{COUNT}_{i=1}^{k}(d_i \leq d^*_{max})]/k$, where d^*_{max} is the maximum distance of the k exact distances. Figure 3 shows the average error over query points for various datasets. The neighbors identified using entropy formula are much closer than using classical formula, even in the case using classical formula with $r+1$. Table VI shows the average recall over query points. Typically, the entropy formula gave higher recall.

2) k-Furthest Neighbors: We evaluate the formulas on four KFN algorithms: Linear Scan, QDAFN, Drusilla, and RQALSH. Two criteria are used: ratio and recall, as introduced in [19]. The ratio is $\sum_{i=1}^{k} d^*_i/d_i$, where d^*_i is the exact ith furthest distance, d_i is the resulted one. The recall is $[\text{COUNT}_{i=1}^{k}(d_i \geq d^*_{min})]/k$, where d^*_{min} is the minimum distance of the k exact distances. It is better if the ratio is closer to 1.0. The bigger the recall is, the better the performance is. Table IV shows the results for the four algorithms on the GOM dataset. Table III shows the ratio for the state-of-the-art on various datasets. Compared with the classical formula, the entropy formula gives much better results for both criteria.

VIII. Concluding remarks

The Maximum Entropy method is known to provide simple solutions to complex problems. We used it to derive enhanced formulas to several problems in dimension reduction and verified them experimentally.

Table V: Accuracy for Mahalanobis distance using PCA

		x from arbitracy vectors		x from dataset	
		$k=5$	$k=25$	$k=5$	$k=25$
CB	$\lvert\text{md}_{\text{classical}} - \text{md}_{\text{exact}}\rvert$	2.2193E+01	2.2133E+01	2.6442E-01	1.6827E-01
	$\lvert\text{md}_{\text{entropy}} - \text{md}_{\text{exact}}\rvert$	**2.2000E+01**	**2.0815E+01**	**8.9313E-02**	**3.8883E-02**
MD	$\lvert\text{md}_{\text{classical}} - \text{md}_{\text{exact}}\rvert$	6.5609E+03	6.5588E+03	1.1244E-01	1.0783E-01
	$\lvert\text{md}_{\text{entropy}} - \text{md}_{\text{exact}}\rvert$	**6.3868E+03**	**6.3745E+03**	**6.3627E-03**	**5.6926E-03**
YP	$\lvert\text{md}_{\text{classical}} - \text{md}_{\text{exact}}\rvert$	8.4728E+04	8.4683E+04	2.3630E-04	1.9112E-04
	$\lvert\text{md}_{\text{entropy}} - \text{md}_{\text{exact}}\rvert$	**8.4277E+04**	**8.3277E+04**	**8.5088E-05**	**4.0074E-05**

Table VI: Recall for KNN using PCA on various datasets

$r:k$	YP		GOM		Ionosphere	
	$d_{\text{classical}}$	d_{entropy}	$d_{\text{classical}}$	d_{entropy}	$d_{\text{classical}}$	d_{entropy}
1 : 10	0	**0.005**	0.040	**0.080**	0.215	**0.310**
5 : 10	0.035	**0.060**	0.365	**0.390**	**0.735**	**0.735**
10 : 10	0.220	**0.235**	0.490	**0.570**	0.860	**0.885**
20 : 10	0.545	**0.560**	**0.705**	0.700	0.910	**0.955**
1 : 20	0	**0.005**	0.040	**0.108**	0.323	**0.478**
5 : 20	0.043	**0.073**	0.393	**0.470**	0.740	**0.838**
10 : 20	0.243	**0.280**	0.570	**0.648**	0.863	**0.915**
20 : 20	0.573	**0.625**	0.723	**0.770**	0.925	**0.958**

References

[1] G. Wan, C. Maung, and H. Schweitzer, "Improving the accuracy of principal component analysis by the maximum entropy method," in *ICTAI*, 2019, pp. 808 – 815.

[2] Jolliffe, *Principal Component Analysis*. Springer, 2002.

[3] G. H. Golub and C. F. Van-Loan, *Matrix Computations*, 4th ed. Johns Hopkins University Press, 2013.

[4] M. B. Cohen, C. Musco, and C. Musco, "Input sparsity time low-rank approximation via ridge leverage score sampling," in *SODA '17*. SIAM, 2017, pp. 1758–1777.

[5] T. Sarlos, "Improved approximation algorithms for large matrices," in *FOCS*, 2006, pp. 143–152.

[6] K. G. Larsen and J. Nelson, "Optimality of the johnson-lindenstrauss lemma," in *FOCS*, 2017, pp. 633–638.

[7] K. L. Clarkson and D. P. Woodruff, "Low-rank approximation and regression in input sparsity time," *Journal of the ACM*, vol. 63, no. 6, pp. 54:1–54:45, 2017.

[8] J. Nelson and H. L. Nguyen, "OSNAP: faster numerical linear algebra algorithms via sparser subspace embeddings," in *FOCS*, 2013, pp. 117–126.

[9] E. T. Jaynes, "On the rationale of maximum entropy methods," *Proceedings of IEEE*, vol. 70, no. 9, pp. 939–952, 1982.

[10] A. Papoulis and S. Pillai, *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[11] A. Papoulis, "Maximum entropy and spectral estimation: A review," *IEEE TSSP*, vol. 29, no. 6, pp. 1176–1186, 1981.

[12] M. Rozanski, R. Witula, and E. Hetmaniok, "More subtle versions of the Hadamard inequality," *Linear Algebra and its Applications*, vol. 532, pp. 500–511, Nov. 2017.

[13] O. Cheong, C.-S. Shin, and A. Vigneron, "Computing farthest neighbors on a convex polytope," *Theoretical Computer Science*, vol. 296, no. 1, pp. 47 – 58, 2003.

[14] B. Yao, F. Li, and P. Kumar, "Reverse furthest neighbors in spatial databases," in *2009 IEEE 25th International Conference on Data Engineering*, March 2009, pp. 664–675.

[15] R. R. Curtin, J. R. Cline, N. P. Slagle, W. B. March, P. Ram, N. A. Mehta, and A. G. Gray, "Mlpack," *JMLR*, vol. 14, no. Mar, pp. 801–805, 2013.

[16] S. Bespamyatnikh, "Dynamic algorithms for approximate neighbor searching," in *Proceedings of the 8th Canadian Conference on Computational Geometry*, 1996.

[17] R. Pagh, F. Silvestri, J. Sivertsen, and M. Skala, "Approximate furthest neighbor in high dimensions," in *Similarity Search and Applications*. Springer, 2015, pp. 3–14.

[18] R. Curtin and A. B. Gardner, "Fast approximate furthest neighbors with data-dependent candidate selection," in *Similarity Search and Applications*. Springer, 2016, pp. 221–235.

[19] Q. Huang, J. Feng, Q. Fang, and W. Ng, "Two efficient hashing schemes for high-dimensional furthest neighbor search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2772–2785, Dec 2017.