RC24859 (W0909-031) September 10, 2009
Computer Science

IBM Research Report

A Lower Bound on the Euclidean Distance for Fast Nearest Neighbor Retrieval in High-dimensional Spaces

George Saon, Peder Olsen
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

A Lower Bound on the Euclidean Distance for Fast Nearest Neighbor Retrieval in High-dimensional Spaces

George Saon and Peder Olsen
IBM T.J. Watson Research Center
route 134, Yorktown Heights, NY, 10598
gsaon@us.ibm.com

September 9, 2009

Abstract

Finding the nearest neighbor among a large collection of high dimensional vectors can be a computationally demanding task. In this paper, we pursue fast vector matching by representing vectors in $\mathbb{R}^n$ with lower dimensional projections in $\mathbb{R}^m$, $m \leq n$. The key to creating and using the representative vectors is a lower bound on the Euclidean distance between arbitrary vectors in $\mathbb{R}^n$ based on the submultiplicative property of induced matrix norms. For any non-zero projection matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the bound is proportional to the distance between the projected vectors. We study other existing bounds involving orthogonal transforms and piecewise constant approximation maps in light of this formulation. Additionally, we address the question of how to optimize the projection matrix given a dataset in order to make the bound as tight as possible. Experimental results on a speech database show that exact nearest neighbor computation can be accelerated by a factor of 5 using the proposed bound.

1 Introduction

The Euclidean distance is one of the most ubiquitous concepts in many areas such as pattern recognition, machine learning, image retrieval, database search and signal processing to name only a few. For instance, in a query by image content application, the user wants to retrieve images from a database (which can potentially have hundreds of thousands of records) that are similar to the query image. This similarity is often computed as a distance between high-dimensional feature vectors which describe the contents of the query and the database images. The feature vectors, whose dimensionality can be in the hundreds, usually contain attributes pertaining to the color, texture and shape.

The need for fast Euclidean distance computation is also apparent for large vocabulary continuous speech recognition. In this case, for every 10ms speech frame, the system finds the Gaussians with the highest likelihoods, which requires computing weighted Euclidean distances between the frame and the means of the Gaussians, in order to provide acoustic likelihoods for the various phonetic classes. For a system trained on thousands of hours of data with several hundreds of thousands of Gaussians (of typically tens of dimensions), a brute force approach would result in 10^{14} weighted distance evaluations per training iteration, which is clearly prohibitive.

There are two main strategies for speeding up the nearest neighbor computation in Euclidean spaces. The first is to organize the image database from the first example (or the collection of Gaussians from the second example) in a hierarchichal data structure such as a tree. Cover trees are a prime example of this approach and they can achieve exact nearest neighbor retrieval in logarithmic time if the dataset has some intrinsic low-dimensional structure (Beygelzimer et al., 2006).

The second strategy, complementary to the first, is to find a tight lower bound on the Euclidean distance which can be computed much faster than the actual distance. The bound serves as a filtering function (also called indexing or signature function (Faloutsos et al., 1994)). The algorithm for exact nearest neighbor works as follows: if the value of the bound between the query and the sample being examined is greater

than the smallest distance so far, the sample cannot be the nearest neighbor of the query. If the value is less, however, the true distance has to be computed, and, if it is smaller than the current smallest distance, the current smallest distance and the nearest neighbor are updated. It is easy to see that the number of false hits (samples for which the lower bound is less and the actual distance is greater than the smallest distance) which have to be considered depends on the tightness of the bound.

There have been several approaches to lower bounding the Euclidean distance in the literature. Most of them project the feature vectors to a low-dimensional space and consider the bound to be a scaled distance between the transformed feature vectors. The scale depends on the type of transform that is being applied. For orthogonal transforms, such as KLT (Karhunen Loève Transform) (Fukunaga, 1990), DCT (discrete cosine transform) or the Haar Wavelet Transform, the optimal scale is one. Piecewise constant approximation transforms (Keogh et al., 2001; Yi and Faloutsos, 2000; Faloutsos et al., 1994) consist in reducing the vectors from n to $m \leq n$ dimensions by dividing the dimensions into m equisized "frames" and by recording the average value of the dimensions falling within a frame. In this case, the optimal scale turns out to be $\sqrt{n/m}$. Finally, it is worth mentioning the work of (Jeong et al., 2006), where the authors propose a lower bound on the distance by approximating the angle between a query vector and a data vector as the absolute value of the difference of two angles involving a third vector (called reference vector).

The paper is organized as follows: in section 2 we formulate our new bound and study some example applications involving orthogonal transforms and piecewise constant approximation maps. In section 3 we show some experimental evidence of the utility of this method, and concluding remarks will be presented in section 4.

2 Problem formulation

Consider $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ two n-dimensional vectors. In order to introduce some notation used in this paper, we recall that the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ is the norm of the difference vector

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})} \tag{1}$$

where $\|\cdot\|$ denotes the L_2-norm in $\mathbb{R}^n$. Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ a projection matrix of rank $m \leq n$, the Euclidean distance in the range of $\mathbf{A}$ is

$$d_{\mathbf{A}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y}\| = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{A}^T \mathbf{A} (\mathbf{x} - \mathbf{y})} \tag{2}$$

Here $\|\cdot\|$ represents the L_2-norm in $\mathbb{R}^m$. The idea is to do the projection $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ beforehand for all vectors rather than redoing it for every comparison of two vectors.

The main objective of the paper is to relate the distance in the original space to the distance in the projected space in the form of an inequality. Concretely, we ask the question if there exists a non-trivial mapping $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ such that

$$f(\mathbf{A}) d_{\mathbf{A}}(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{y}) \qquad \forall \mathbf{A} \in \mathbb{R}^{m \times n},\ \mathbf{A} \neq \mathbf{0},\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \tag{3}$$

As we will see later, such a function does indeed exist and can be expressed in terms of the induced matrix norm of $\mathbf{A}$.

2.1 Matrix norms

Matrix norms are frequently used in the analysis of algorithms involving matrices. For instance, the quality of a linear system solver can be poor if the matrix of coefficients is nearly singular. This notion of near-singularity can be quantified by a matrix norm which provides a measure of distance on the space of matrices. Since $\mathbb{R}^{m \times n}$ is isomorphic to $\mathbb{R}^{mn}$, one would expect matrix norms to obey the same properties as vector norms. However, $\mathbb{R}^{m \times n}$ is not just a high-dimensional vector space; it has a natural multiplication operation.

Hence, we expect matrix norms to have additional properties. $\|\cdot\| : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}$ is a matrix norm if it satisfies (cf. (Golub and Van Loan, 1989)):

$$\begin{array}{ll} \|\mathbf{A}\| \geq 0 \ (\|\mathbf{A}\| = 0 \text{ iff } \mathbf{A} = 0) & \mathbf{A} \in \mathbb{R}^{m\times n} \\ \|\mathbf{A}+\mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\| & \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n} \\ \|\alpha\mathbf{A}\| = |\alpha| \cdot \|\mathbf{A}\| & \alpha \in \mathbb{R}, \mathbf{A} \in \mathbb{R}^{m\times n} \\ \|\mathbf{AB}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\| & \mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{B} \in \mathbb{R}^{n\times k} \end{array} \tag{4}$$

The last axiom is known as the submultiplicative property, and it is used to compare the "size" of the product of two matrices to the "sizes" of the individual matrices. Note that the norms that appear in it operate in three different spaces ($\mathbb{R}^{m\times k}$, $\mathbb{R}^{m\times n}$ and $\mathbb{R}^{n\times k}$).

2.2 Induced norms

Let $\|\cdot\|$ be the L_2 vector norm (on $\mathbb{R}^n$ or $\mathbb{R}^m$). The following matrix norm defined on $\mathbb{R}^{m\times n}$ is called the induced norm

$$\|\mathbf{A}\| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} = \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\| \tag{5}$$

The intuitive interpretation of $\|\mathbf{A}\|$ is the maximum "magnification" capability of $\mathbf{A}$. It is easy to verify that (5) satisfies all the properties of a matrix norm. Of particular interest to us is the submultiplicative property for $k = 1$ which is the inequality

$$\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\| \qquad \forall \mathbf{x} \in \mathbb{R}^n \tag{6}$$

In the following, let us try to characterize $\|\mathbf{A}\|$. Define the function g as

$$g(\mathbf{x}) = \frac{1}{2}\frac{\|\mathbf{Ax}\|^2}{\|\mathbf{x}\|^2} = \frac{1}{2}\frac{\mathbf{x}^T\mathbf{A}^T\mathbf{Ax}}{\mathbf{x}^T\mathbf{x}} \tag{7}$$

The corresponding gradient equation is given by

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = \frac{\mathbf{A}^T\mathbf{Ax}}{\mathbf{x}^T\mathbf{x}} - \frac{(\mathbf{x}^T\mathbf{A}^T\mathbf{Ax})\mathbf{x}}{(\mathbf{x}^T\mathbf{x})^2} = 0. \tag{8}$$

Due to invariance to scaling of (8) we may without loss of generality assume $\|\mathbf{x}\| = 1$. Therefore the following equality holds at the maximum

$$\mathbf{A}^T\mathbf{Ax} = (\mathbf{x}^T\mathbf{A}^T\mathbf{Ax})\mathbf{x} = \|\mathbf{Ax}\|^2\mathbf{x} = \|\mathbf{A}\|^2\mathbf{x} \tag{9}$$

which means that $\mathbf{x}$ is an eigenvector of $\mathbf{A}^T\mathbf{A}$.

By inspection we see that the eigenvector corresponding to the largest eigenvalue maximizes the objective function. It follows that $\|\mathbf{A}\| = \max\{\sqrt{\lambda} | \lambda \text{ is an eigenvalue of } \mathbf{A}^T\mathbf{A}\} = \sqrt{\lambda_{\max}}$ where $\lambda_{\max}$ denotes the maximum eigenvalue of $\mathbf{A}^T\mathbf{A}$. In the case of square matrices, $\|\mathbf{A}\|$ is also known as the spectral norm of $\mathbf{A}$.

2.3 Proposed lower bound

By applying (6) to the difference vector $\mathbf{x} - \mathbf{y}$, we arrive at the main result of this paper which can be stated in a very compact form

$$\frac{1}{\|\mathbf{A}\|}\|\mathbf{Ax} - \mathbf{Ay}\| \leq \|\mathbf{x} - \mathbf{y}\| \qquad \forall \mathbf{A} \in \mathbb{R}^{m\times n},\ \mathbf{A} \neq \mathbf{0},\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \tag{10}$$

The bound is also valid for the weighted Euclidean (or Mahalanobis) distance in which case

$$\frac{1}{\|\mathbf{A}\|}\|\mathbf{AW}^{\frac{1}{2}}\mathbf{x} - \mathbf{AW}^{\frac{1}{2}}\mathbf{y}\| \leq \sqrt{(\mathbf{x} - \mathbf{y})^T\mathbf{W}(\mathbf{x} - \mathbf{y})} \tag{11}$$

where $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a symmetric positive definite weight matrix and $\mathbf{W}^{\frac{1}{2}}$ denotes the square root of $\mathbf{W}$.

2.4 Example 1: orthogonal transforms

A common approach to speeding up the distance computation between two high-dimensional vectors is to perform dimensionality reduction by applying an orthogonal transformation which preserves distances and by "throwing away" the higher order coefficients. These transforms can be broadly grouped into two classes depending on whether they are learned from data (such as the Karhunen-Loeve transform) or on whether they are data-independent. For the latter category, several transforms have been successfully employed such as the Discrete Fourier Transform, the Discrete Cosine Transform and the Haar Wavelet Transform, to name a few.

In this case, the columns of $\mathbf{A}^T = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$, $\mathbf{a}_i \in \mathbb{R}^n$, are orthonormal vectors i.e. $\mathbf{a}_i^T \mathbf{a}_j = \delta_{ij}$. We want to show that

$$\|\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y}\| \leq \|\mathbf{x} - \mathbf{y}\| \tag{12}$$

From (10), all we have to do is prove that $\|\mathbf{A}\| = 1$. Indeed, $\mathbf{A}^T \mathbf{A}\mathbf{a}_i = \mathbf{a}_i$, $i = 1 \ldots m$, meaning that 1 is an eigenvalue of $\mathbf{A}^T\mathbf{A}$ with multiplicity m. Since the rank of $\mathbf{A}^T\mathbf{A}$ is also m the result follows.

2.5 Example 2: piecewise constant approximation transforms

These transforms originated in the image and time series database retrieval literature (Keogh et al., 2001), and are also known as averaging transforms (Faloutsos et al., 1994) or segmented means transforms (Yi and Faloutsos, 2000). Simply stated, to reduce the vectors from n dimensions to m dimensions, we divide the dimensions into m equisized "frames" and we record the average value of the dimensions falling within a frame. For convenience, we assume that n is a multiple of m, although this is not a requirement[1]. This transform can be described by a projection matrix as follows: let $k = n/m$ and define $\mathbf{A} = (a_{ij})$

$$a_{ij} = \begin{cases} 1/k & \text{if } (\mathrm{i}-1)\mathrm{k}+1 \leq \mathrm{j} \leq \mathrm{ik} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The piecewise constant approximation bound states that

$$\sqrt{k}\|\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y}\| \leq \|\mathbf{x} - \mathbf{y}\| \tag{14}$$

A simple proof of this bound can be found in (Yi and Faloutsos, 2000), where the authors use a convexity argument. We shall give an alternative proof using (10). We have to show that $\|\mathbf{A}\| = 1/\sqrt{k}$ or equivalently, that the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$ is $1/k$. Observe that

$$\mathbf{A}^T\mathbf{A} = \frac{1}{k^2}\,\mathrm{diag}(\underbrace{\mathbf{1}_k\mathbf{1}_k^T, \mathbf{1}_k\mathbf{1}_k^T, \ldots, \mathbf{1}_k\mathbf{1}_k^T}_{m \text{ times}}) \tag{15}$$

with $\mathbf{1}_k \in \mathbb{R}^k$ being the all-ones vector. It follows that the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are the eigenvalues of $\mathbf{1}_k\mathbf{1}_k^T$ repeated m times, multiplied by a factor of $1/k^2$. But $\mathbf{1}_k\mathbf{1}_k^T$ is a rank one matrix with k the only non-zero eigenvalue and $\mathbf{1}_k/\sqrt{k}$ the corresponding eigenvector. Hence the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$ is $1/k^2 \cdot k = 1/k$.

2.6 Optimizing the bound

We attempt to answer the following question: for a fixed dimension $m \leq n$, is there an optimal projection matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ for a given data set? A natural objective function is the sum of squared differences between the distances in $\mathbb{R}^n$ and the scaled distances in $\mathbb{R}^m$. The distances are computed between the vectors from the data set and their nearest neighbors. More formally, for $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^n$, we want to find $\mathbf{A}$ which minimizes

[1] The vectors can be padded with zeros such that the new dimension becomes a multiple of m.

$$
\begin{aligned}
f(\mathbf{A}) &= \sum_{i=1}^{N} \frac{1}{2} \left(\frac{1}{\|\mathbf{A}\|} \|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_{\eta_i}\| - \|\mathbf{x}_i - \mathbf{x}_{\eta_i}\| \right)^2 \\
&= \sum_{i=1}^{N} \frac{1}{2} \left(\frac{1}{\|\mathbf{A}\|} \|\mathbf{A}\mathbf{d}_i\| - \|\mathbf{d}_i\| \right)^2
\end{aligned}
\tag{16}
$$

where $\mathbf{d}_i := \mathbf{x}_i - \mathbf{x}_{\eta_i}$ is the difference vector between sample i and its nearest neighbor η_i. The derivative of f with respect to $\mathbf{A}$ has the expression

$$
\frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} = \sum_{i=1}^{N} \left(\frac{1}{\|\mathbf{A}\|} \|\mathbf{A}\mathbf{d}_i\| - \|\mathbf{d}_i\| \right) \left(\frac{1}{\|\mathbf{A}\|} \frac{\partial \|\mathbf{A}\mathbf{d}_i\|}{\partial \mathbf{A}} - \frac{\|\mathbf{A}\mathbf{d}_i\|}{\|\mathbf{A}\|^2} \frac{\partial \|\mathbf{A}\|}{\partial \mathbf{A}} \right) \tag{17}
$$

It is straightforward to show using again (Searle, 1982), that the partial derivative of the norm of the projected distances is

$$
\frac{\partial \|\mathbf{A}\mathbf{d}_i\|}{\partial \mathbf{A}} = \frac{1}{\|\mathbf{A}\mathbf{d}_i\|} \mathbf{A}\mathbf{d}_i\mathbf{d}_i^T \tag{18}
$$

Calculating the differential of the matrix norm is more involved. Consider $\mathbf{X} = \mathbf{P}^T\mathbf{\Lambda}\mathbf{P}$ to be a symmetric positive semi-definite matrix with $\mathbf{P}$ orthogonal and $\mathbf{\Lambda} = \operatorname{diag}(\underbrace{\lambda_{max}, \ldots, \lambda_{max}}_{k \text{ times}}, \lambda_{k+1}, \ldots, \lambda_n)$ where k is the multiplicity of the maximum eigenvalue. The Schatten p-norm of $\mathbf{X}$ is the p-norm of the vector of eigenvalues, and can be written as

$$
\|\mathbf{X}\|_p^S = \left(\sum_{i=1}^{n} \lambda_i^p \right)^{\frac{1}{p}} = \operatorname{trace}(\mathbf{\Lambda}^p)^{\frac{1}{p}} = \operatorname{trace}(\mathbf{X}^p)^{\frac{1}{p}} \tag{19}
$$

Next, we note that the maximum eigenvalue is attained as the limit

$$
\lambda_{max} = \lim_{p \to \infty} \|\mathbf{X}\|_p^S \tag{20}
$$

i.e. the spectral norm of $\mathbf{X}$ coincides with the Schatten infinity norm. From (19) we have

$$
\frac{\partial}{\partial \mathbf{X}} \left(\sum_{i=1}^{n} \lambda_i^p \right)^{\frac{1}{p}} = \frac{\partial}{\partial \mathbf{X}} \operatorname{trace}(\mathbf{X}^p)^{\frac{1}{p}} = \operatorname{trace}(\mathbf{X}^p)^{\frac{1}{p}-1} (\mathbf{X}^{p-1})^T \tag{21}
$$

Using (21) and (20) we find the derivative of the maximum eigenvalue as

$$
\begin{aligned}
\frac{\partial \lambda_{max}}{\partial \mathbf{X}} &= \lim_{p \to \infty} \frac{(\mathbf{X}^{p-1})^T}{\operatorname{trace}(\mathbf{X}^p)} \operatorname{trace}(\mathbf{X}^p)^{\frac{1}{p}} \\
&= \mathbf{P}^T \operatorname{diag}(\underbrace{\frac{1}{k\lambda_{max}}, \ldots, \frac{1}{k\lambda_{max}}}_{k \text{ times}}, 0, \ldots, 0) \mathbf{P} \lambda_{max} \\
&= \frac{1}{k} \sum_{j=1}^{k} \mathbf{p}_j \mathbf{p}_j^T
\end{aligned}
\tag{22}
$$

where $\mathbf{p}_1, \ldots, \mathbf{p}_k$ are the eigenvectors corresponding to $\lambda_{\max}$. The desired matrix norm derivative is obtained via the chain rule

$$
\frac{\partial \|\mathbf{A}\|}{\partial a_{ij}} = \operatorname{trace} \left[\left(\frac{\partial \|\mathbf{A}\|}{\partial (\mathbf{A}^T\mathbf{A})} \right)^T \frac{\partial (\mathbf{A}^T\mathbf{A})}{\partial a_{ij}} \right] \tag{23}
$$

By plugging (22) into (23) for $\mathbf{X} = \mathbf{A}^T\mathbf{A}$, we arrive at

$$\frac{\partial\|\mathbf{A}\|}{\partial\mathbf{A}} = \frac{\mathbf{A}}{k\|\mathbf{A}\|}\sum_{j=1}^{k}\mathbf{p}_j\mathbf{p}_j^T \tag{24}$$

where $\mathbf{p}_1, \ldots, \mathbf{p}_k$ are the eigenvectors corresponding to the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$. Combining (17), (18) and (24) leads us to the final expression for the gradient

$$\frac{\partial f(\mathbf{A})}{\partial\mathbf{A}} = \frac{\mathbf{A}}{\|\mathbf{A}\|}\sum_{i=1}^{N}\left(\frac{1}{\|\mathbf{A}\|}\|\mathbf{A}\mathbf{d}_i\| - \|\mathbf{d}_i\|\right)\left(\frac{\mathbf{d}_i\mathbf{d}_i^T}{\|\mathbf{A}\mathbf{d}_i\|} - \frac{\|\mathbf{A}\mathbf{d}_i\|}{k\|\mathbf{A}\|^2}\sum_{j=1}^{k}\mathbf{p}_j\mathbf{p}_j^T\right) \tag{25}$$

To the best of our knowledge, $\partial f(\mathbf{A})/\partial\mathbf{A} = 0$ does not have a closed form solution; we have to resort to a gradient descent optimization instead.

3 Experiments and results

In this section, we describe some experiments which were carried out on a speech database containing 50 hours of audio from English broadcast news television shows. We randomly selected three sets of 144-dimensional feature vectors: one set containing 200000 vectors which form the training database and two other sets of 10000 vectors each which are the development set and the query set. The vectors are obtained in the following way: we extract 12-dimensional PLP (Hermansky, 1990) cepstral coefficients every 10 msec and concatenate 12 consecutive frames to arrive at the 144 dimensional feature vectors. All experiments were run on an Intel Xeon 3.6GHz processor. The total CPU time to retrieve the exact nearest neighbors of the 10K query vectors using exhaustive search was 604.5 seconds. In Figure 1, we show the speed-ups as a function of the dimension of the projected feature vectors using the proposed method with various transforms: KLT estimated on the training data, DCT and two piecewise constant approximation transforms, averaging across frames (AAF) and averaging within frames (AWF).

As can be seen, the KLT has the best performance with an optimal speedup of 4.8 and there are minor differences between the top three transforms for more than 36 dimensions. Interestingly, DCT and the AAF transform are almost indistinguishable. In contrast, the AWF transform has a poor performance. This can be explained by the fact that the same cepstral coefficients are correlated across frames but are uncorrelated with each other within a frame, making the average a poor predictor of the individual coefficients.

Next, we address the question of what happens when we try to optimize the transform according to the objective function (16). The objective function was computed using the distances between the 10K vectors from the development set and their nearest neighbors from the training set. We used the C++ implementation of the limited memory BFGS algorithm (Zhu et al., 1997) for the optimization. In Figure 2, we show the evolution of the objective function (16) for five initial transforms: KLT, DCT, AAF, AWF and a random transform. The elements of the random transform are uniformly distributed in $[-1, 1]$. All transforms have $m = 24$ lines.

Table 1 gives the CPU times for nearest neighbor retrieval with the five transforms before and after the optimization. The search speeds for the initial AWF and random transforms are worse than brute force search because the bounds they provide are too loose for pruning. Consequently, these transforms benefit the most from the optimization, followed by the AAF transform and the DCT. There is no improvement in search time for the KLT after optimization but notice that this transform was estimated on all the training data whereas the optimization uses only the nearest neighbors. In the future, we will explore ways of making use of the entire training data in the optimization. Another observation which can be made is that the objective function is a good predictor of the search speed: better transforms have lower objective function values.

4 Discussion

In this paper, we generalized some existing lower bounds on the Euclidean distance between high-dimensional vectors. These bounds are scaled distances between lower dimensional projections of the vectors. The gener-

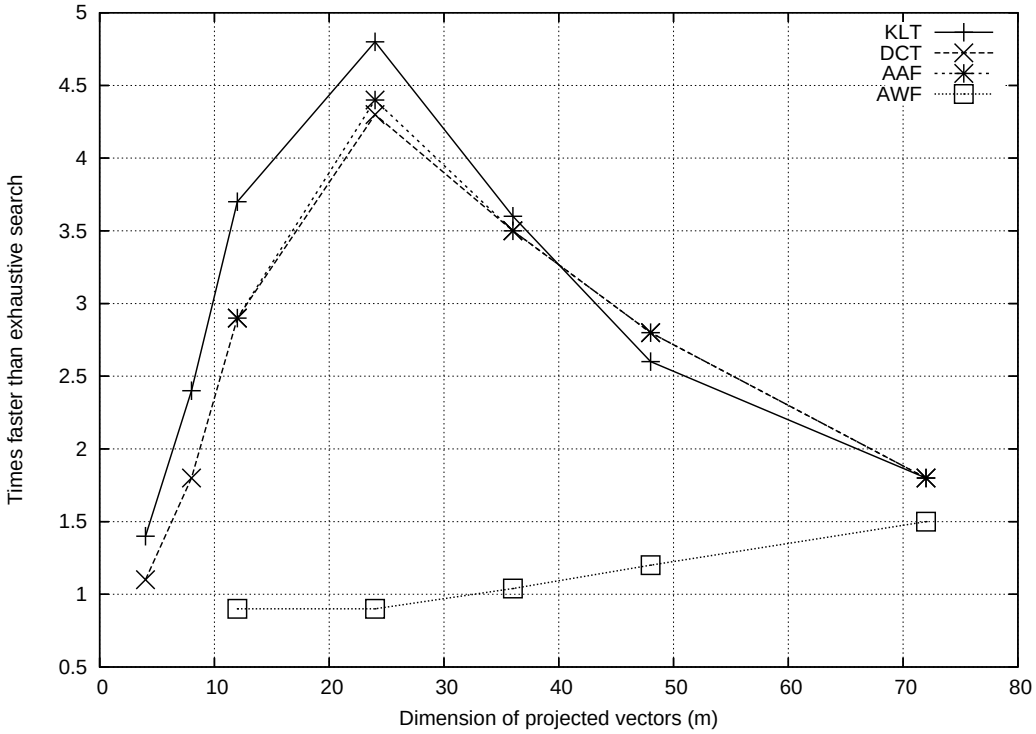

Figure 1: Speed-ups as a function of the dimension of the projected vectors for the proposed method over naive nearest neighbor search using various transforms: KLT, DCT, averaging across frames (AAF) and averaging within frames (AWF).

alization comes from observing that the bounds are particular instances of the submultiplicative property of induced matrix norms for various orthogonal and piecewise constant approximation matrices. A practical application of such bounds is a filtering function for fast nearest neighbor computation. Experimental results on a speech database suggest that the Karhunen-Loeve transform achieves the fastest nearest neighbor retrieval time. For very high-dimensional applications where computing KLT might be expensive, other alternatives such as the discrete cosine transform can be considered. Piecewise constant approximation transforms are also effective and simple to apply for data which exhibits a particular structure, such as images and time series data. Furthermore, we found that nearly optimal transforms can be obtained by directly minimizing the sum of squared differences between original and scaled projected distances for various initial matrices.

	Initial	Optimized	Speed-up
KLT	126.4 sec	126.3 sec	4.8
DCT	140.9 sec	138.1 sec	4.4
AAF	137.8 sec	132.3 sec	4.6
AWF	677.4 sec	132.6 sec	4.6
Random	740.8 sec	133.2 sec	4.5

Table 1: Nearest neighbor search times before and after optimization for KLT, DCT, piecewise constant approximation transforms and random transform. The last column is speed-up over naive search after optimization.

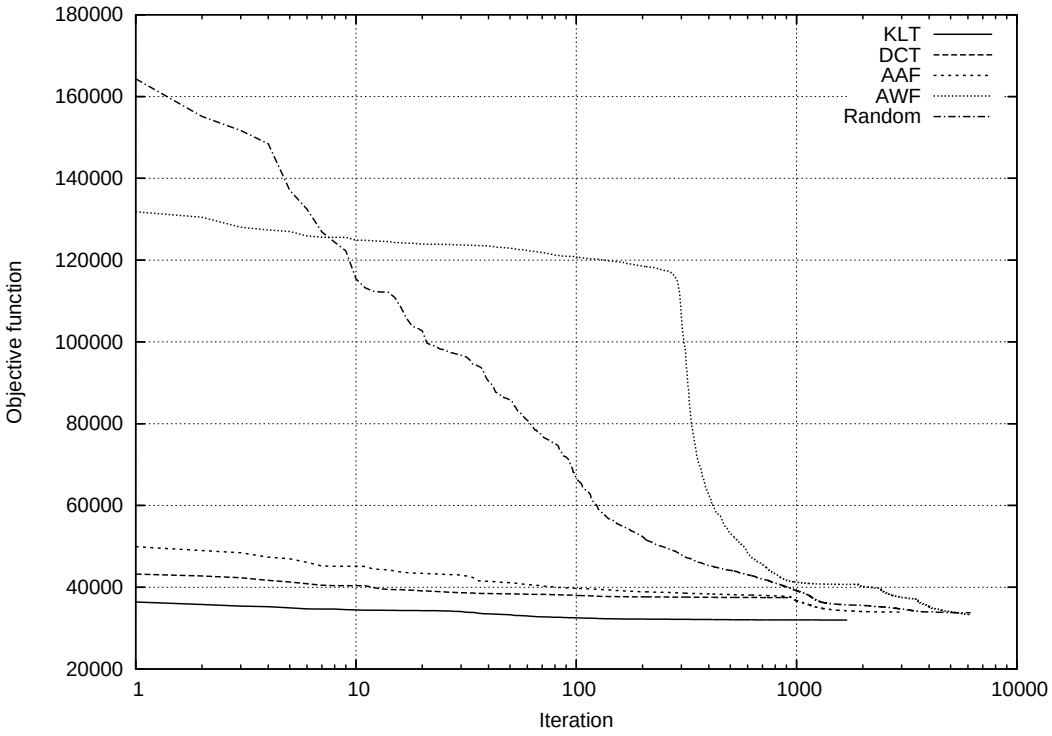

Figure 2: Evolution of the objective function (16) using various initial transforms: KLT, DCT, AAF, AWF and random.

References

A. Beygelzimer, S. Kakade, and J. Langford. 2006. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 97–104.

C. Faloutsos, R. BArber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. 1994. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262.

K. Fukunaga. 1990. *Introduction to Statistical Pattern Recognition*. Elsevier.

G.H. Golub and C.F. Van Loan. 1989. *Matrix Computations*. The Johns Hopkins University Press.

H. Hermansky. 1990. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752.

S. Jeong, S.W. Kim, K. Kim, and B.Y. Choi. 2006. An effective method for approximating the euclidean distance in high-dimensional space. *Database and Expert Systems Applications*, pages 863–872.

E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286.

S. R. Searle. 1982. *Matrix Algebra Useful for Statistics*. Wiley-Interscience.

B.K. Yi and C. Faloutsos. 2000. Fast time sequence indexing for arbitrary lp norms. *The VLDB Journal*, pages 385–394.

C. Zhu, R.H. Byrd, and J. Nocedal. 1997. L-bfgs-b: fortran routines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560.