CLASSIFICATION OF IMAGES WITH TANGENT DISTANCE

JEN-MEI CHANG, MICHAEL KIRBY, LUCAS KRAKOW, JOSH LADD, ETHAN MURPHY

Abstract. In digital image recognition it is typical to represent an image as a point in a high dimensional vector space $\mathbb{R}^m$. Nonlinear deformations of this image define a k dimensional manifold where $k \leqslant m$. In this paper such a manifold will be referred to as an image's corruption surface. Empirical investigations have demonstrated that this manifold tends to be differentiable in a neighborhood of the original image. Thus it is possible to calculate an optimal linear approximation about the original image that captures the relevant linear effects of deformation. This subspace, called the tangent space, typically offers a lower dimensional characterization of the image, and most importantly this space contains nearly the same information as the original manifold for small deformations. This is important in digital image recognition as large deformations of an image tend to transform the image into something other than the original image to be classified. Thus information about large deformations may lead to misclassification. The prototypical example of this being a 6 deformed into a 9 by the deformation operation of rotation. In this paper we attempt to build a digital image classifier by defining a metric on pixel space that is based on the linearization about the image (tangent space) on the corruption surface. We will attempt to calculate the tangent space in several ways and attempt to classify low resolution handwritten digital images. We will first attempt a cost effective method that requires a priori knowledge of the deformations and amounts to nothing more than numerical differentiation with respect to these operations. The second method will employ a powerful scaling argument. We will calculate the best local KL basis about an image. This will be a basis for the tangent space. This method is highly desirable as it requires no a priori knowledge of the deformations, and may be immediately applied to empirical data.

1. Introduction

In the familiar facial recognition problem we are given a set of labeled training images, the goal is to present the training data with corrupted versions of an original image and find the best match amongst the labeled data. Classification using the standard Euclidean distance has been shown to be highly unreliable. Classification using the distance between the corrupted pattern and the corruption manifold of each of the training patterns lead to expensive nonlinear optimization problems and tends to misclassify patterns due to the globally invariant nature of the manifold.

While in principle the manifold contains all deformations we will see that manifolds induced by different images share many commonalities. This can be understood intuitively by noting that different pictures may share similar qualities.

It will be our primary objective to construct a metric that will capture small deformations of an image with the idea that small deformations do not change

Date: May 10, 2004.

Key words and phrases. Singular value decomposition, tangent distance, k-nearest neighbor algorithm, dendrogram.

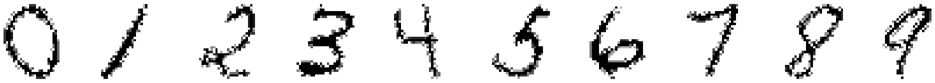

Figure 1. Prototype digits from the coarser/noisy USPS dataset. The grayscale ranges from 0 to 35.

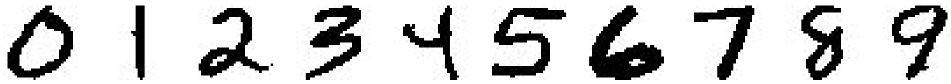

Figure 2. Prototype digits from the finer NIST dataset. The grayscale ranges from 0 to 255.

the fundamental nature of an image while large deformations might. Thus we shall work under the assumption that an image's robust signature resides primarily amongst the small order linear effects of deformation. In this way we attempt to characterize an image not as a high dimensional nonlinear surface, but as a low dimensional linear subspace that contains the most salient and robust features of an image.

Description of Databases

We obtained a list of USPS handwritten sample digits from the internet provided by the Center of Excellence for Document Analysis and Recognition (CEDAR) State University of New York at Buffalo. This database is intended to encourage research in off-line handwriting recognition. We also obtained a database from the National Institute of Standards and Technology (NIST) Online Scientific Databases website (http://www.nist.gov/srd/online.htm) under the Image Recognition section.

From these two databases, we create a noisy database that consists of 50 32×32 pixel size handwritten digits, coming from the United State Postal Service (USPS) database. The gray level of each image is between 0 and 35, see figure 1 for the prototype digits for this database. The second database consists of 50 32×32 pixel size handwritten digits, coming from National Institute of Standards and Technology (NIST) database. The gray level of each image is between 0 and 255. The original NIST set we obtained has 3471 images, we selected a training set of 50 images from it, see figure 2 for the prototype digits for this database.

Overview

In section 2 we introduce the tangent space and the associated tangent distance metric. In this section we also present algorithms for calculating the tangent distance once the tangent space is obtained. In section 3 we demonstrate how to calculate the tangent space using numerical differentiation. In section 4 we do the same only this time employing the SVD scaling argument. In section 5 we compare the differences in these methods. In section 6 we present a numerical implementation of the SVD method. In section 7 we present the results of testing and training. In section 8 we do further analysis of the results with the novel use of dendrograms. In section 9 we summarize our results.

2. The Tangent Distance

As previously mentioned the goal will be to develop a metric on $\mathbb{R}^m$ that accurately identifies two images as being the same (one corrupted) even when the two images may differ greatly. Also it must be able to distinguish between two different patterns that share many similarities. In abstract terms we seek a function that quantifies how different two images E and P are in the following way; imagine deforming image E to best fit image P let $\delta(E,P)$ be a measure of the difference in the best fit. Let $\epsilon(E,P)$ be a measure of the amount E had to be deformed to best fit P then the following mapping from the space of two images to the reals defines a metric on the space of pixels

$$D(E,P) = \delta(E,P) + \epsilon(E,P) \tag{2.1}$$

that satisfies the desired criteria. Note that $D(E,P)$ may be large even if the $\delta(E,P)$ is small. If E had to be greatly deformed to achieve the good fit then $\epsilon(E,P)$ is large, and vice versa .

We now construct a metric that has the previous characteristics in the following way; we begin with a set of training data

$$X = \{x^p\}_{p=1}^q,\ x^p \in \mathbb{R}^m \tag{2.2}$$

and suppose that there are n known labels which the data is classified under. There is a known map from the indices of X to the labels

$$\nu(p) = k, k \in \{1, ..., n\} \tag{2.3}$$

Now to each label $k = 1, ..., n$ there is an associated set of patterns defined as

$$H_k = \{p = 1, ..., q | \nu(p) = k\} \tag{2.4}$$

Now for an incoming pattern whose label is unknown x^p we would like to construct a map that will classify the pattern under a label.

$$x^p \to \{1, ..., n\} \tag{2.5}$$

We do not want to have a prohibitively large training set, so an immediate way of reducing the dimension of the data is to identify the centroid of each pattern. The image defined by

$$C_k = 1/|H_k| \sum_{\nu(p)=k} x^p \tag{2.6}$$

is the centroid of the data with label k. One classification strategy might be to introduce a new pattern and measure its Euclidean distance to C_k. As we shall see this approach is naive and highly unreliable.

The next approach might be to construct a manifold for each training image. We classify a new image by finding the minimum distance to the manifold over all manifolds and associate to the new pattern the winning pattern's label. This approach will lead to complicated nonlinear optimization problems and it is flawed in many ways, in particular images may reside on the manifold that no longer resemble the original image. We again remind the reader of the prototype example of the digit 6 deformed to a 9 by rotation.

Another approach, and the one we will focus on in the remainder of this paper is the idea of the tangent space about an image. We would like to calculate the tangent space about two images and measure the shortest distance between the two subspaces. The minimum distance between subspaces over all tangent spaces will classify the pattern. This calculation is nothing more than a least squares problem for which several robust algorithms exists for computation. This metric in which a tangent space is calculated for both patterns is called a two-sided tangent distance. An even simpler method involves only the tangent spaces of the training data. The one sided tangent distance is defined as the residual of the projection of the new pattern onto the training pattern's tangent space.

2.1. **Defining The Tangent Space.** In this section we attempt to formalize the preceding section. Let $S_{x^p}(\alpha)$ be a nonlinear mapping from $\mathbb{R}^m \to \mathbb{R}^m$ that deforms the image x^p by α where $\alpha \in \mathbb{R}^s$ is a vector of s deformation operation parameters. Furthermore the set defined as

$$S_{x^p} = \{y \in \mathbb{R}^m | \exists \alpha \ni y = S_{x^p}(\alpha)\} \tag{2.7}$$

is the corruption surface of the image. From elementary calculus the space tangent to the manifold at x^p is spanned by the colums of the Jacobian matrix

$$J = \frac{\delta S_{x^p}(\alpha)}{\delta \alpha}|_{\alpha=0} \tag{2.8}$$

So now in a neighborhood of $\alpha = 0$ the manifold satisfies the following

$$S_{x^p}(\alpha) \approx x^p + J\alpha \tag{2.9}$$

2.2. **Calculating the Tangent Distance.** In this section we present algorithms for calculating one and two sided tangent distances.

We begin with the one sided tangent distance. Let x be a labeled piece of data. Let y be a new image to be classified. The one sided tangent distance from y to x is

$$D(x, y) = ||y - JJ^{\intercal}y||_2 \tag{2.10}$$

Where J is the tangent space about x.

We anticipate that the two-sided tangent distance will perform better than the one-sided tangent distance; this in turn should then perform better than the Euclidean distance. We again adopt the notation in [PYSV98] for the following discussion. The two-sided tangent distance (TD) between two patterns E and P is found by first defining tangent spaces for the two respective patterns. If we define,

$$TD(E, P) = \min_{x \in T_E,\, y \in T_P} ||x - y||_2^2 \tag{2.11}$$

where T_E is the tangent plane of S_E at E and T_P is the tangent plane of S_P at P, then the task amounts to finding the the shortest distance between the two subspaces. Hence,

$$TD(E, P) = \min_{\alpha_E, \alpha_P} ||(E + L_E\alpha_E) - (P + L_P\alpha_P)||$$

Where α_E and α_P are solutions of the following system of equations, given in [PYSV98]:

$$\left(L_{PE}L_{EE}^{-1}L_E^T - L_P^T\right)(E - P) = \left(L_{PE}L_{EE}^{-1}L_{EP} - L_{PP}\right)\alpha_P \tag{2.12}$$

$$\left(L_{EP}L_{PP}^{-1}L_P^T - L_E^T\right)(E - P) = \left(L_{EE} - L_{EP}L_{PP}^{-1}L_{PE}\right)\alpha_E \tag{2.13}$$

where $L_{EE} = L_E^T L_E$, $L_{EP} = L_E^T L_P$, $L_{PE} = L_P^T L_E$ and $L_{PP} = L_P^T L_P$. Once α_E and α_P are obtained, the two-sided tangent distance can be calculated easily since the matrices above will have full rank.

The one sided tangent distance is clearly cheaper to compute however the added classification accuracy obtained from a two-sided calculation makes the more expensive computation attractive.

To control deformation and increase overall performance of the classifier to be built we introduce spring constants on α_E and α_P. This places a weight on the distance from the projection of the image onto the tangent space to the original image. In this way we see that the tangent distance measures roughly how much a new pattern is deformed to fit the original image and the tension in the spring quantifies how good the fit is.

If one incorporates a spring constant k into equation 2.11 then we obtain the following new definition of the tangent distance.

$$D(E,P) = \min_{\alpha_E,\alpha_P} ||E + L_E\alpha_E - P - L_P\alpha_P||^2 + k||L_E\alpha_E||^2 + k||L_P\alpha_P||^2 \tag{2.14}$$

Note that for $k = 0$ (2.14) reduces to the standard tangent distance, while for $k = \infty$, we have the Euclidean distance. Now α_E and α_P are solutions of the system

$$\left(L_{PE}L_{EE}^{-1}L_E^T - (1+k)L_P^T\right)(E-P) = \left(L_{PE}L_{EE}^{-1}L_{EP} - (1+k)^2L_{PP}\right)\alpha_P$$

$$\left(L_{EP}L_{PP}^{-1}L_P^T - (1+k)L_E^T\right)(E-P) = \left((1+k)^2L_{EE} - L_{EP}L_{PP}^{-1}L_{PE}\right)\alpha_E$$

This weighting of the tangent distance is particularly useful when some of the tangent vectors of E or P are collinear. Although this situation is non-generic it is still possible that a pattern may be duplicated in both the training and the testing set.

All that remains is an effective way of calculating the Jacobian J. In this paper we consider two such methods.

3. The Secant Line Method

In this section we numerically construct the tangent space by making successive secant line approximations. This approach has a major downfall in that we must have a prior knowledge of the deformation operations. In this paper we assume that the deformation operations are translation, rotation, and dilation. We built these operators as programs in MATLAB employing cubic splines to interpolate and move the images around. In this way we have the manifold defined by these operations. Let $S(P,\alpha)$ be a prototypical deformation operation. In order to calculate the tangent space to any degree of accuracy desired we simply invoke the following algorithm

Algorithm 1.

```
let $n = 1$,
let $h = 1/n$,
let $f_1 = S(P,h) - S(P,-h)/||S(P,h) - S(P,-h)||$,
let $f_2 = 5f_1$,
while $|(f_1, f_2) - 1| > \epsilon$,
n=n+1,
h=1/n,
$f_2 = f_1$,
$f_1 = S(P,h) - S(P,-h)/||S(P,h) - S(P,-h)||$
end
```

In this way we were able to calculate the tangent space induced by a deformation operation to any given error tolerance ϵ.

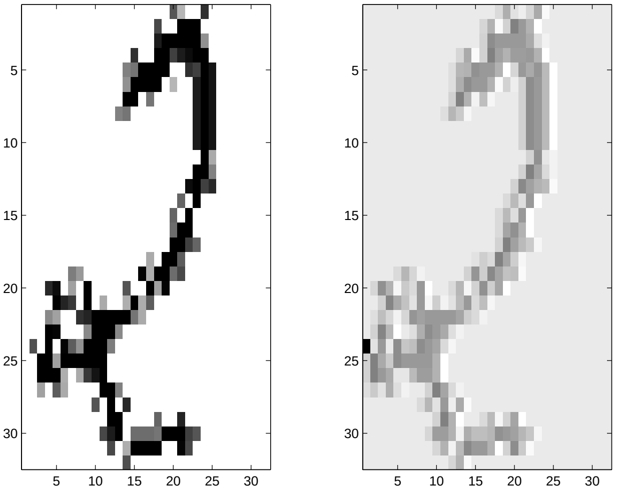

FIGURE 3. Actual translation (left) and image on tangent space about translation (right) as calculated by secant line approximation.

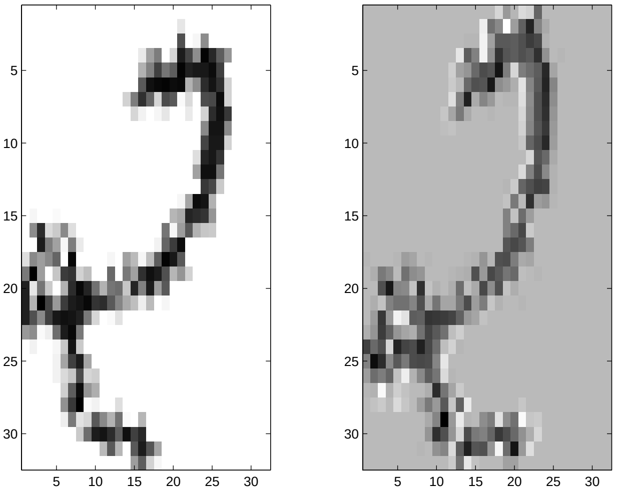

FIGURE 4. Actual rotation (left) and image on tangent space about rotation (right) as calculated by secant line approximation.

4. The SVD Method

In this section we employ a scaling argument that allows us to calculate a basis of the tangent space. This method offers a highly attractive alternative to the secant line approach of computing the tangent space as it requires no a priori knowledge of the deformation operators. We use the following result of *Broomhead, Jones, & King* [BJK87]

Theorem 2. *Consider the mapping $f : U \to X$ about the point $x_0 \in U$. A basis for the tangent space T_{x_0} is provided by the singular values of the $\epsilon-$neighborhood matrix $B_\epsilon(x_0)$ that scale linearly with the radius ϵ.*

Theorem 2 employs the familar singular value decomposition, which we will briefly review. Additional details and applications of the SVD can be found in [Kir01].

Theorem 3. *Singular Value Decomposition (SVD). Let A be a real $m \times n$ matrix and $k = \min\{m, n\}$. There exist orthogonal matrices U and V such that*

$$A = U \Sigma V^T,$$

where $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ and $\Sigma =$ diag$(\sigma^{(1)}, \dots, \sigma^{(k)}) \in \mathbb{R}^{m \times n}$.

SVD is an efficient method of dimensionality reduction, even though its computation is quite expensive. We adopt the notation in theorem 3, and suppose $m \geq n$, then the reduced SVD, or the thin SVD is

$$A = \hat{U} \hat{\Sigma} V^T,$$

where $\hat{U} \in \mathbb{R}^{m \times n}$, and $\hat{\Sigma} \in \mathbb{R}^{n \times n}$ is a square matrix. If the rank of A is r, then there are r nonzero singular values and thus only the first r left singular vectors are needed to span $Image(A)$. Furthermore, and a most appealing feature of the decomposition is that the left singular vectors provide an optimal basis for $Image(A)$. This allows us to approximate $Image(A)$ in an optimal way. See figure 5 for a reduced rank approximation.

We adopt the notation in [PYSV98] for the discussion of the tangent vectors and the tangent distances. Given an image P, we denote $S(P, \alpha)$ the transformed image of P based on the transformation parameter α. Note that $P = S(P, 1)$ if α is the dilation parameter and $P = S(P, 0)$ if any other parameter is used. We then create the difference matrix (DM) $S(P, \alpha) - P$, where the α's are such that $S(P, \alpha)$'s are contained in the $\epsilon-$ball centered at P. We identify the largest singular values of $DM(\epsilon)$ for each outward expanding ϵ to see which singular values scale linearly. Where the singular values stop scaling linearly is where the manifold is no longer locally linear and thus the tangent approximation is no longer a good one.

Since SVD more readily reveals the geometry of the data and the largest singular value gives the most information about the direction of the tangent space, we just select the largest singular value to determine what ϵ to choose for creating the difference matrix. It should be noted that the difference matrix here is centered at the mean which invokes a slightly different geometric interpretation. The tangent space is spanned by the left singular vectors of the difference matrix whose corresponding singular values are scaling linearly as a function of ϵ. The number of tangent vectors chosen is seen to be a data dependent problem. See figure 7 for an illustration of the epsilon balls for the prototype 2 in the noisy database (USPS). See figure 8 for the prototype 2 in the finer dataset (NIST). See figure 9 for the largest singular values for the first 10 digits in the USPS database. Notice that ϵ is roughly the same for all digits.

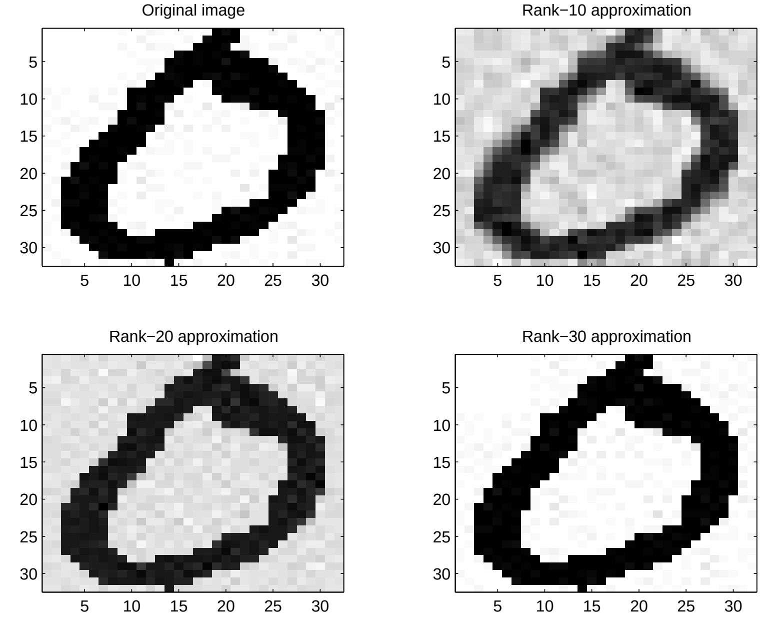

FIGURE 5. The SVD of a rank 32 matrix (digit 0). The other images are rank-10, rank-20 and rank-30 approximations to the original image. In this illustrative example, each data point in the matrix is a column of the digit 0. We will use SVD on an ensemble of pictures, where each column is an entire digit.

Results obtained with the noisy data set and finer data sets are compared. For the USPS data set, singular values stop scaling linearly when $\epsilon \approx 0.3$, while for the NIST dataset, singular values still scale linearly when ϵ is 0.4. The quality of the data set will influence the quality of the tangent approximation as one might expect. However, the overall behavior of the singular values are quite similar for the different data sets.

Note that singular values do not appear to scale linearly when ϵ is close to 0 because we intentionally set it to 0 if no data is contained in the ϵ-ball. This demonstrates some of the difficulties one encounters when working with a small training set. We attempted to remedy this by expanding our data set by dilating, rotating, and translating 100 times the prototype digits for $\alpha_d = 0.8$ to $\alpha_d = 1.2$, $\alpha_r = -5$ to $\alpha_r = 5$, $\alpha_h = -5$ to $\alpha_h = 5$, and $\alpha_v = -5$ to $\alpha_v = 5$, respectively.

4.1. **SVD v.s. Secant line approximation.** In this section we explain why we adopt the SVD approach over the secant line approximation in finding the tangent vectors. While the secant line approach of finding the tangent vector costs much less than the SVD approach, the secant line approach fails to reveal the geometry of the data set. Also in practice, we have no a priori knowledge of the deformation operations, or the operations are inherently complicated-e.g. the illumination operation. Intrinsic variations of the dataset could be illumination, translations, dilation, rotation, and thickening, any combination of these, or other less transparent operations as well. The secant line approach will not identify

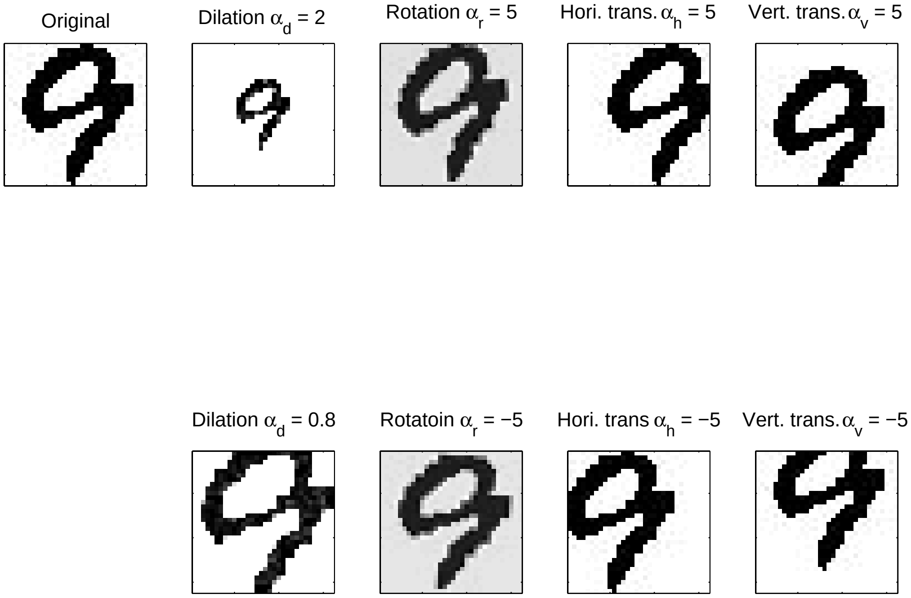
Original
Dilation α_d = 2
Rotation α_r = 5
Hori. trans. α_h = 5
Vert. trans. α_v = 5
Dilation α_d = 0.8
Rotatoin α_r = −5
Hori. trans α_h = −5
Vert. trans. α_v = −5

Figure 6. Images transformed by the dilation, rotation, vertical and horizonal translation functions. First row, from left to right: original 9, dilation with $\alpha = 2$, rotation by 5 degrees, horizontal translation by 5 pixels to the right, vertical translation by 5 pixels down. Second row, from left to right: dilation with $\alpha = 0.8$, rotation by -5 degrees, horizontal translation by 5 pixels to the left, vertical translation by 5 pixels up.

what the intrinsic variations are; it treats distinct datasets the same way. On the other hand the SVD reveals the geometry of the data set and requires no a priori knowledge of the deformation operations. We can identify with ease how many deformation parameters there are by simply identifying the singular values of the difference matrices that scale linearly. A shortcoming of the SVD approach is that it is computationally expensive. However, as pointed out in [HK01], there are ways of getting around this, for example tangent spaces could be calculated off-line if necessary.

5. Applying the SVD to a Toy Problem

5.1. **SVD approach.** To test our method we apply it to a simple problem with known solution. We apply our method to a simple manifold; a circle. More specifically, we find the tangent vector of the unit circle at the point (0,1) and indeed it is the vector $< 1, 0 >$ or $< -1, 0 >$. See figure 11 for an illustration of two possible tangent vectors at the point (0,1).

6. Numerical Implementation of SVD Method

In this section we present the results of applying the SVD method and attempt to demonstrate that by using more singular vectors the subspace approaches the true tangent space. After expanding an epsilon-ball about a prototypical digit,

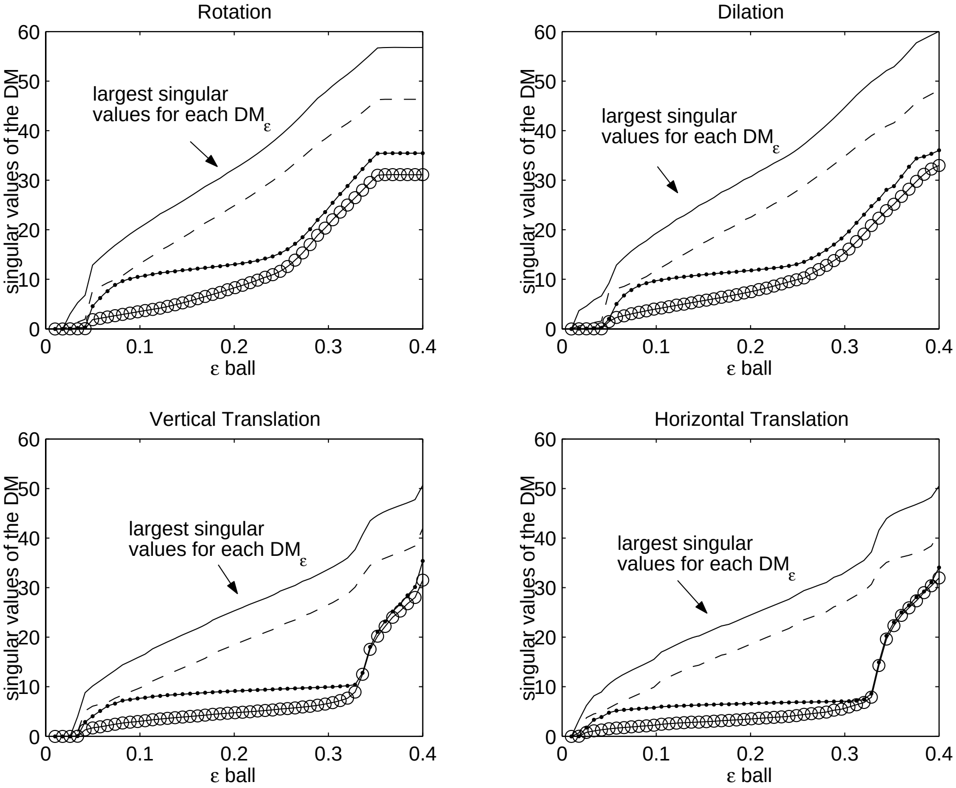

FIGURE 7. First 4 singular values of the difference matrices versus ϵ for the prototype digit 2 in the USPS database.

we noticed that the first two singular values of the difference matrices for each parameter scale fairly linearly. We thus varied the dimension of the subspaces used for reconstruction of a slightly deformed image and consequently the number of the tangent vectors for the 2-sided tangent distance. The results of the reconstruction improved as the number of subspaces increases up to about 4,2,3 and 4 for dilation, rotation, vertical translation and horizontal translation, respectively. Consequently, those are the dimensions of the tangent spaces used for each deformation operation. See figure 12. And when all possible variations are combined to find the difference matrices, relative errors of reconstruction improved as number of tangent vectors increases up to 4. See figure 13.

7. TESTING OF THE TANGENT DISTANCE

In this section we continue testing the SVD method. We test on two simple points in a 3-dimensional space that we already know the distance is 1. Let $E = [6, 1, 0]$ and $P = [0, 0, 1]$ and their tangent vectors $L_E =< 0, 1, 0 >$ and $L_P =< 1, 1, 0 >$, respectively. We find that $\alpha_E = 5$ and $\alpha_P = 6$ via the tangent distance algorithm,

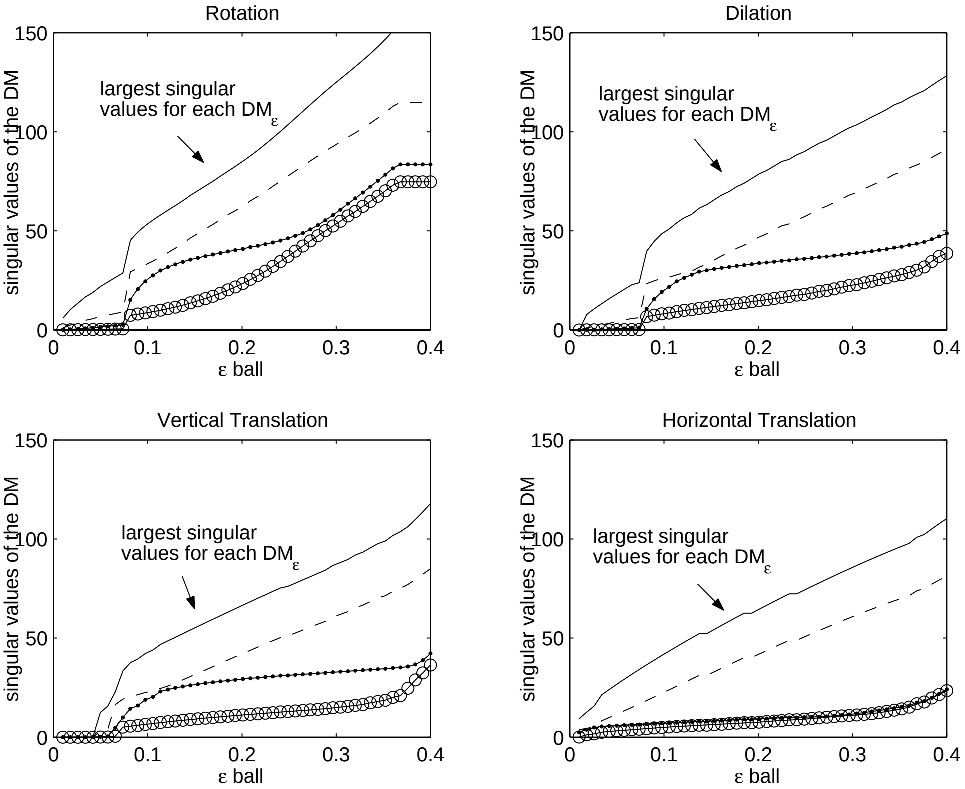

FIGURE 8. First 4 singular values of the difference matrices versus ϵ for the prototype digit 2 in the NIST database.

and

$$\begin{aligned} E' &= E + L_E\alpha_E = [6\ 1\ 0] + [0\ 5\ 0] = [6\ 6\ 0] \\ P' &= P + L_P\alpha_P = [0\ 0\ 1] + [6\ 6\ 0] = [6\ 6\ 1] \end{aligned}$$

Thus, $TD(E, P) = ||E' - P'||_2^2 = 1$, as expected.

Moreover, the local invariance of the tangent distance can be illustrated by transforming a reference digit by various amounts and measuring its distance to a set of prototypes. Figure 14 shows that Euclidean measure confuses the horizontal translated 3 with 7 when it's translated over 4 pixels to the right. If we translate the digit 3 more than 4 pixels to the right, Euclidean measure will misclassify the digit as a 7. Whereas two-sided tangent distance misclassifies the digit 3 if it is horizontally translated more than 6 pixels. This clearly demonstrates the superior performance of the two-sided tangent distance over the other methods in classifing images. Figure 15 shows the improvement of the recognition versus number of tangent vectors used for each transformation parameter. It is important to again note that for each deformation, there is an optimal dimension of the tangent space.

To test the performance of the tangent distance, we compare the averaged precision, recall and accuracy of each distance metric based on the K-nearest neighbor algorithm. The metric used here are Euclidean, one-sided TD and two-sided TD.

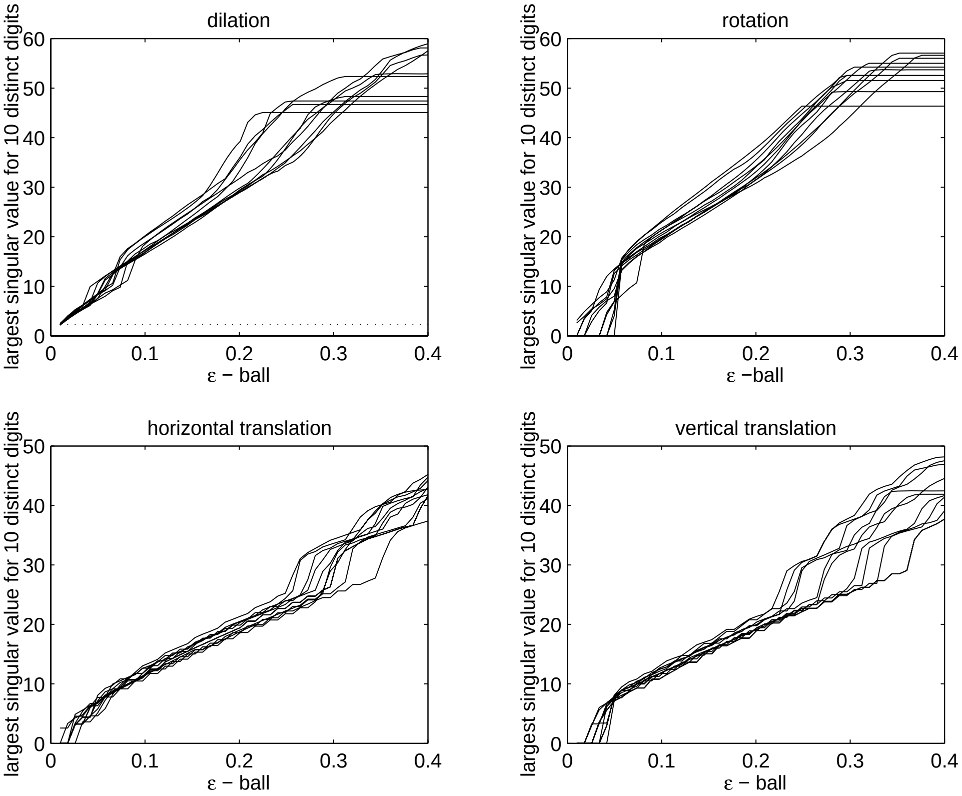

FIGURE 9. The largest singular value of the difference matrices versus ϵ for the first 10 digits in the USPS dataset.

We verify our algorithms work by noting that the two-sided TD is the highest in all three measures.

Precision, recall and accuracy are defined as the following:

$$\begin{aligned}\text{Precision} &= \frac{\text{number of patterns in the correct class}}{\text{total number of patterns in the neighborhood}}\\ \text{Recall} &= \frac{\text{number of patterns in the correct class}}{\text{total number of patterns in the correct class}}\\ \text{Accuracy} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}\end{aligned}$$

Note that no hierarchical classification scheme is employed, hence the classification rate for each metric is independent. This is apparent in figure 17 and figure 18. Since our NIST dataset has approximately 5 numbers per digit it is reasonable to use 5-nearest neighbor in the classification scheme. The precision of the tangent distance is about 13 precent higher than the Euclidean distance when tested on the NIST set for 5 nearest neighbors, see figure 19 and figure 20. A set of similar graphs is shown in figure 21 and figure 22 for the extended NIST database.

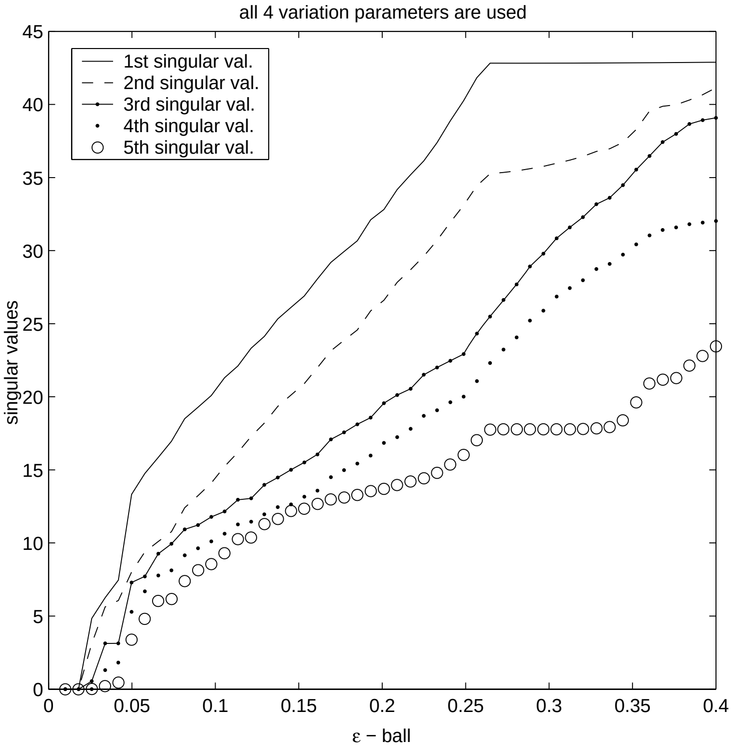

FIGURE 10. First 5 singular values of the difference matrices versus ϵ for the prototype digit 2 in the noisy dataset when all 4 variation parameters are used.

8. Dendrogram Illustrations

Here we propose an interpretation for the single-linkage dendrograms. A more detailed discussion on hierarchical clustering methods can be found in [JW01]. Let $t(m)$ be the number of times one label (digit) transits to a different label (digit) and let $r(m)$ be the number of times a label makes a run of 5 in the dendrogram for the metric m. Define $d(m) = t(m) - r(m)$ to be the dendrogram number for each metric and we denote the Euclidean measure by E, the 1-sided tangent distance measure by TD_1 and the 2-sided tangent distance measure by TD_2. Then in the handwritten digits recognition,

$$\begin{aligned} d(E) &= t(E) - r(E) = 23 - 0 = 23 \\ d(TD_1) &= t(TD_1) - r(TD_1) = 23 - 1 = 22 \\ d(TD_2) &= t(TD_2) - r(TD_2) = 21 - 3 = 19 \end{aligned}$$

The dendrogram numbers indicate the ability of each metric to group alike labels. In our case, the 2-sided tangent distance is able to group all the 2's together and all the 1's together; while the 1-sided tangent distance is able to group all the 1's together and the Euclidean distance is not at all able to group alike digits all together. See figure 23, figure 24 and figure 25. Such an agglomerative tree

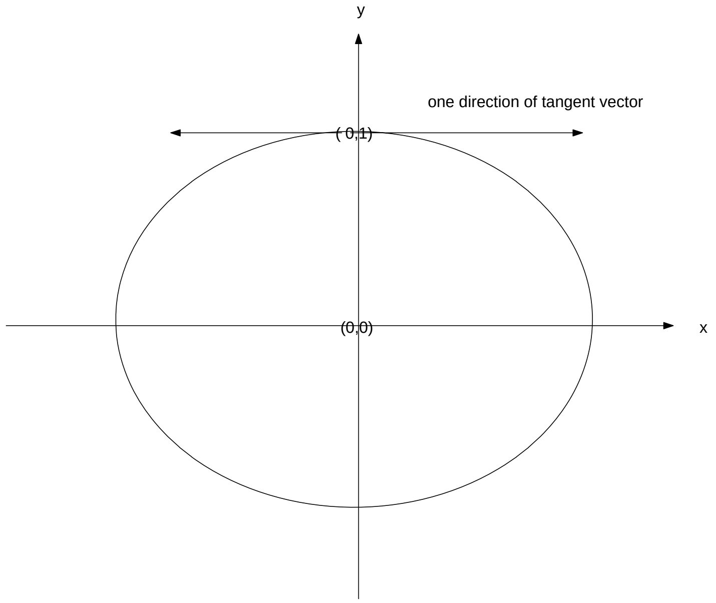

Figure 11. An illustrative example of SVD correctly identifying the tangent vectors on a simple manifold, circle.

structure gives a visualization of similar objects and will be useful for organizing large quantities of images.

9. Conclusion

The notion of tangent distance is easy to comprehend as it is built upon and implemented from the widely used 2-norm, Euclidean norm. Although the Euclidian norm is the basis for tangent distance, the advantage lies in capturing the local invariance's of patterns, forming representative subspaces and then applying the Euclidian distance to two particular points on the subspaces. The concept of local invariance being the deciding factor in pattern classification may seem a bit unsteady, but in fact, is overly important, especially with our sample data. Take rotation for example, with an extensive amount of variance 6 could be recognized as 9. Even with more complex patterns any type of transformation beyond some local boundary could distort it beyond correct classification, causing a decrease in overall classification rate.

The recorded performance of tangent distance without any preprocessing, and only 4 transformation parameters is more than 10 percent higher than the classification rate of the Euclidean distance. One of the main reasons for its success is its ability to incorporate *a priori* knowledge into the measure.

Although, types of transformations that could occur in handwriting were considered in the implementation, the SVD approach employed to translate transformations into subspaces can also be used with empirical training data. This process is a natural next step in the research of tangent distance in pattern recognition. Other

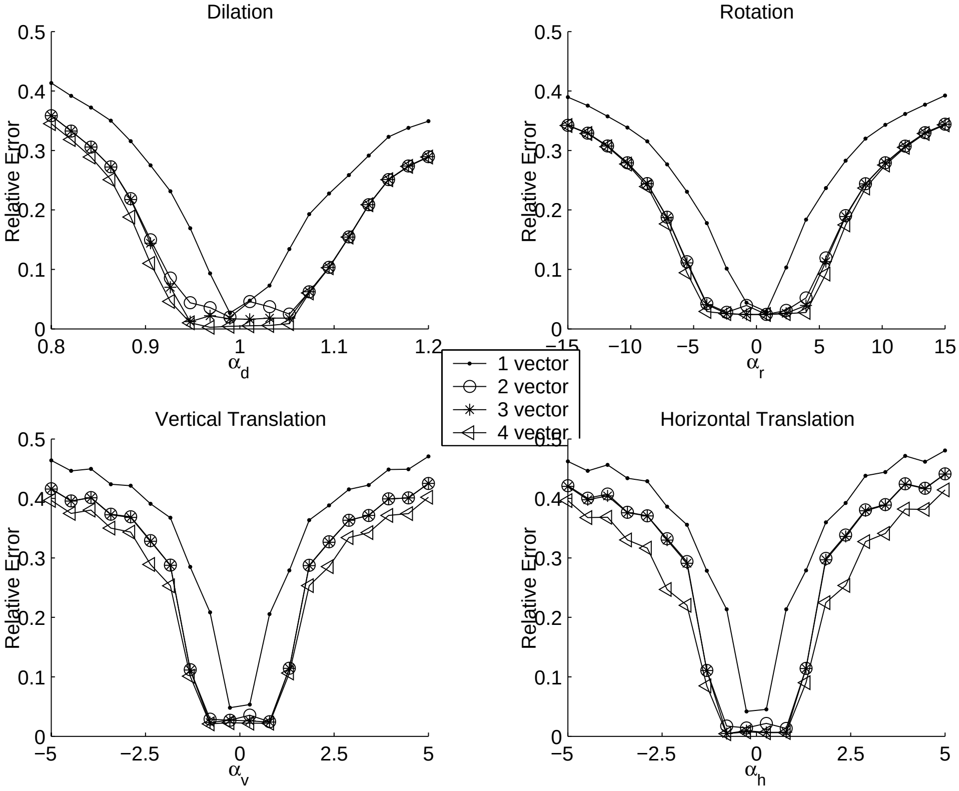

Figure 12. Error of reconstruction for each parameter versus number of tangent vectors used. (done on the USPS database)

steps in the exploration of tangent distance could be, optimization of computational time, and possibly an experiment with some different types of preprocessing to elimination some of the computational load.

Overall, we have shown a successful prototype tested on a set of 50 handwritten digits. Our methods can now be extended and tested on much larger digital data sets, of many varieties, such as digital photos. Here we leave our peers to the future exploration of tangent distance.

References

[BJK87] D. S. Broomhead, R. Jones, and G. P. King, *Topological dimension and local coordinates from time series data*, J. Phys. A: Math. Gen **20** (1987), L563–L569.

[HK01] D. Hundley and M. Kirby, *Estimation of topological dimension*, Proceedings of the Third SIAM International Conference on Data Mining (San Fransico), 2001, pp. 194–202.

[JW01] Richard A. Johnson and Dean W. Wichern, *Applied multivariate statistical analysis*, Prentice Hall, 2001.

[Kir01] Michael Kirby, *Geometric data analysis: An empirical approach to dimensionality reduction and the study of patterns*, Wiley, 2001.

[PYSV98] J. S. Denker P. Y. Simard, Y. A. Cun and B. Victorri, *Transformation invariance in pattern recognition - tangent distance and tangent propagation.*

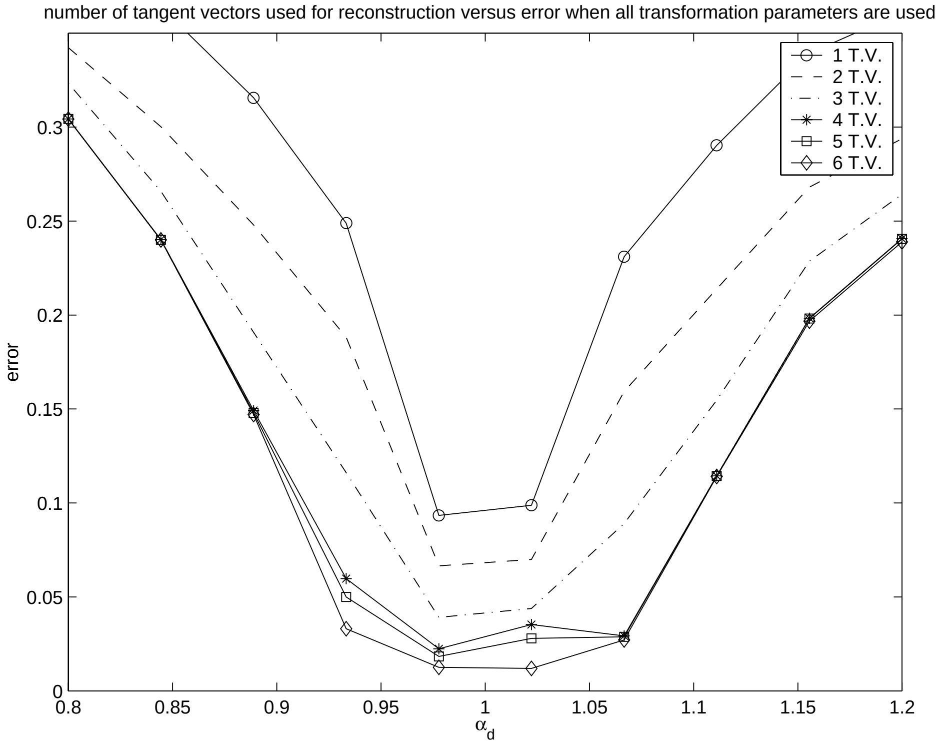

Figure 13. Errors of linear approximation versus number of tangent vectors used when all transformation parameters are used.

Department of Mathematics, Colorado State University, 101 Webber Building, Fort Collins, Colorado 80523

E-mail address: `chang@math.colostate.edu, kirby@math.colostate.edu, krakow@math.colostate.edu, ladd@math.colostate.edu, murphy@math.colostate.edu`

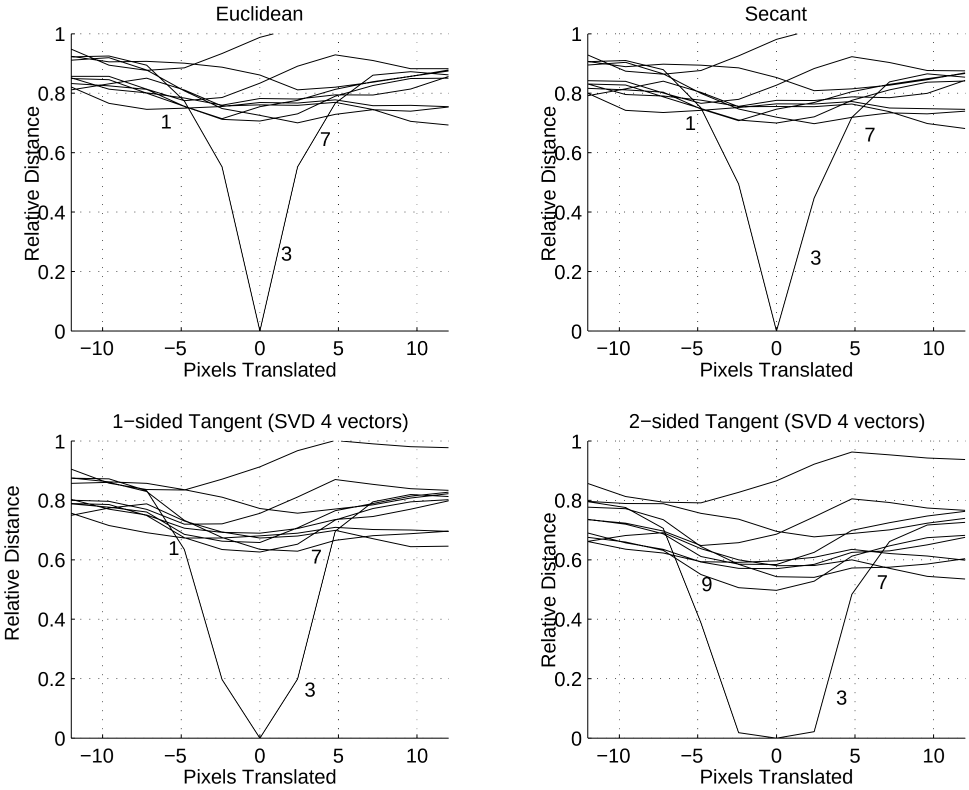

Figure 14. Euclidean, secant, 1-sided tangent and 2-sided tangent distance between 10 prototype digits and a digit 3 translated horizontally by various pixel values. Only horizontal translation tangent vectors are used here. (done on the NIST database)

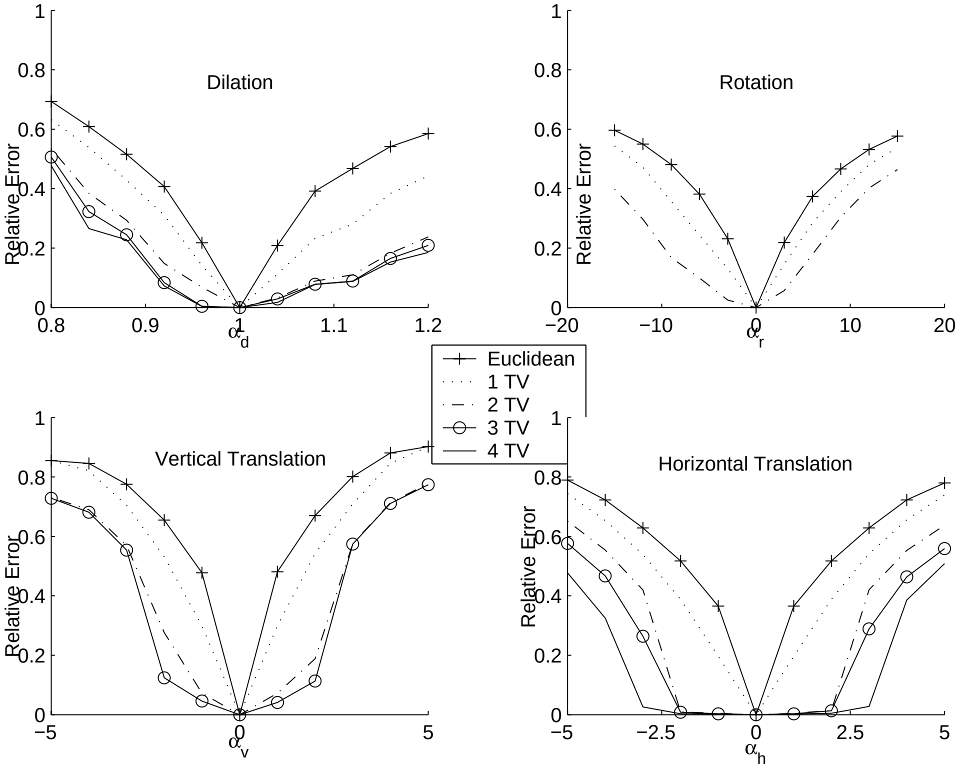

FIGURE 15. A variation of figure 14. Note that for more tangent vectors used, the slower the transformed 3 is confused with other digits. The distances used here are Euclidean and 2-sided tangent distance.

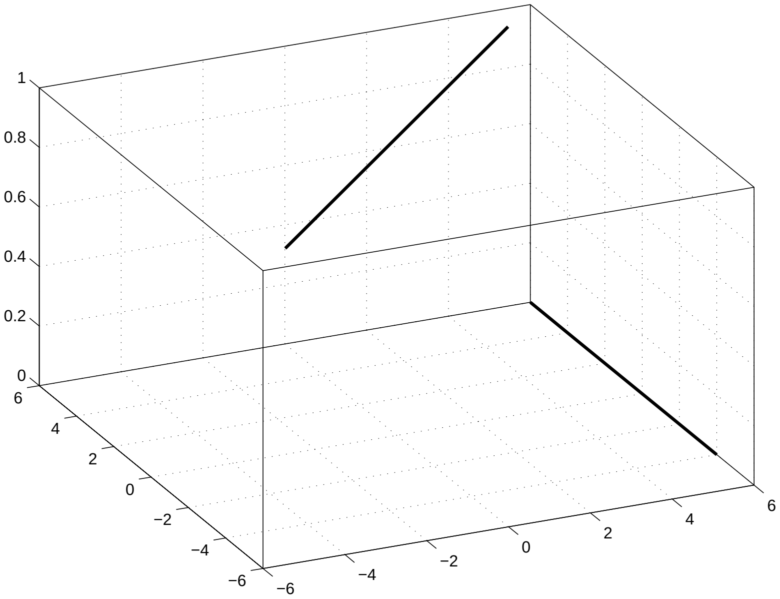

FIGURE 16. An illustrative example of tangent distance between two tangent spaces.

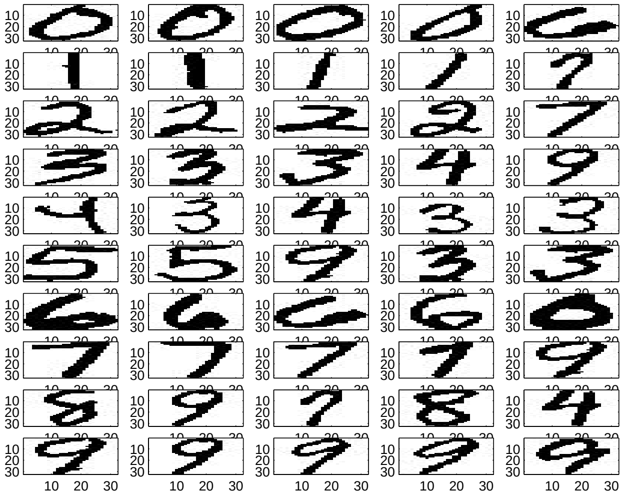

FIGURE 17. 5-nearest neighbor with 2-sided tangent distance. The first column is the prototype digits in the NIST dataset, it also is its own 1-nearest neighbor. The second column is the 2nd-nearest neighbor of that prototype and the third column is the 3rd-nearest neighbor of that prototype, etc.

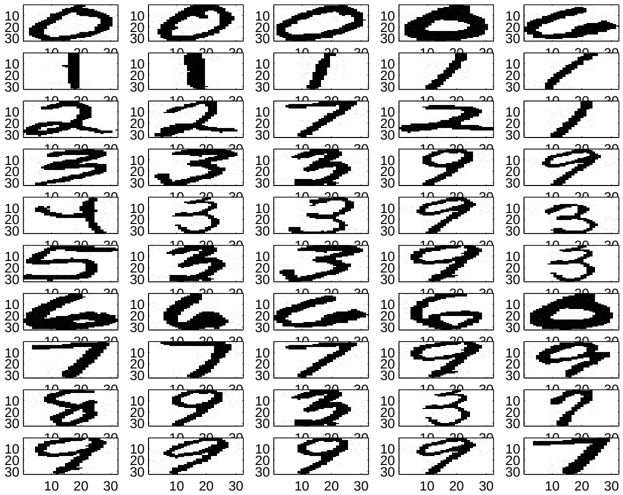

FIGURE 18. 5-nearest neighbor with Euclidean distance. The first column is the prototype digits in the NIST dataset, it also is its own 1-nearest neighbor. The second column is the 2nd-nearest neighbor of that prototype and the third column is the 3rd-nearest neighbor of that prototype, etc.

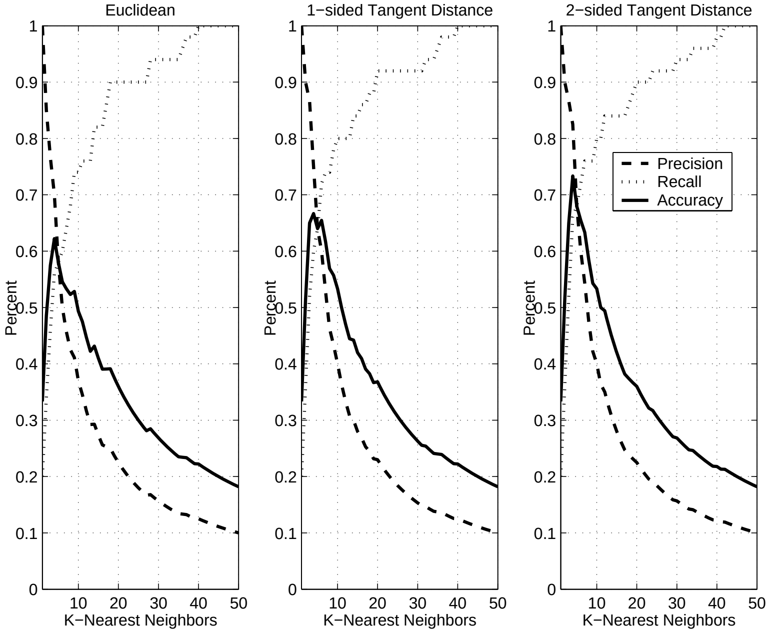

FIGURE 19. Average precision, recall and accuracy of the 10 prototype digits in the NIST database with Euclidean, 1-sided, and 2-sided tangent distance over all transformations.

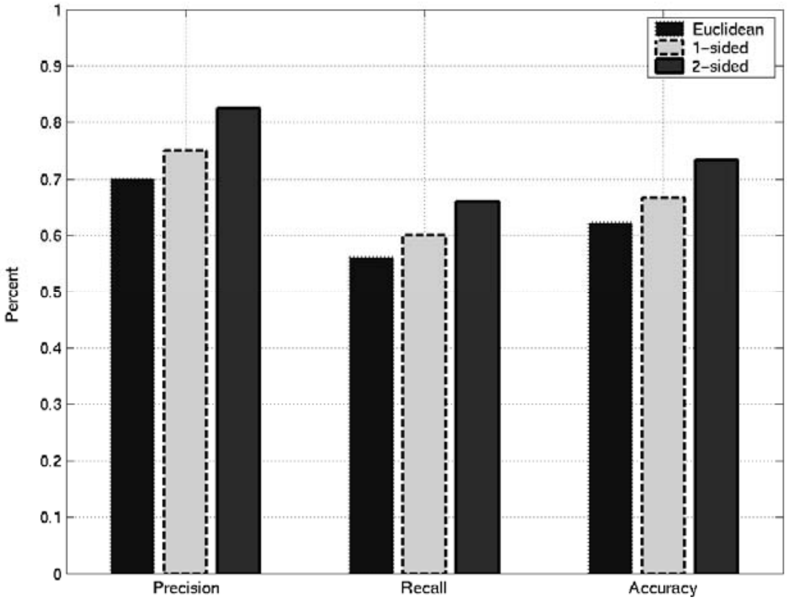

FIGURE 20. Average precision, recall and accuracy for each metric with 5-nearest neighbor over all transformations on the NIST database. (dilation, translations and rotation)

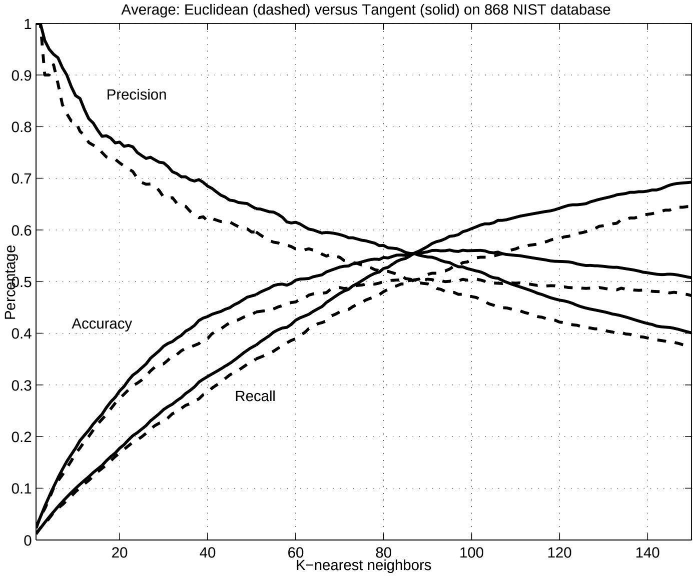

FIGURE 21. Average precision, recall and accuracy of the prototype digits in the extended NIST database of 868 images with Euclidean and 2-sided tangent distance over all transformations. Note that tangent distance is higher in all 3 measures.

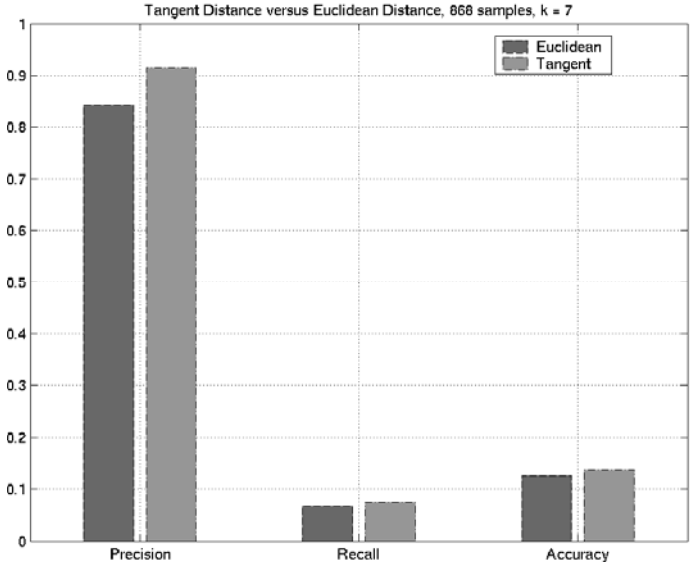

FIGURE 22. Average precision, recall and accuracy with 7 nearest neighbors for Euclidean and 2-sided tangent distance over all transformations. (done on the extended NIST database of 868 images. Note that tangent distance is higher in all 3 measures.)

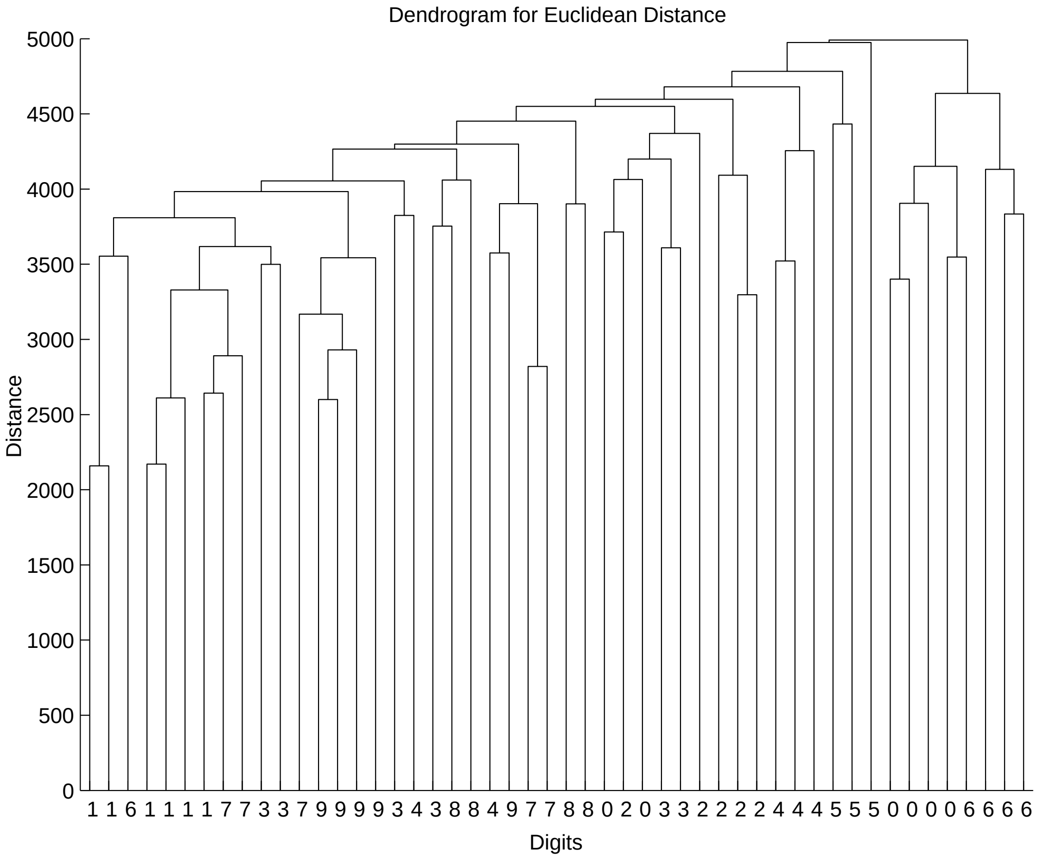

Figure 23. Dendrogram with Euclidean distance. The labels on the bottom of the dendrogram represent digits. The numbers that are linked together resemble the most.

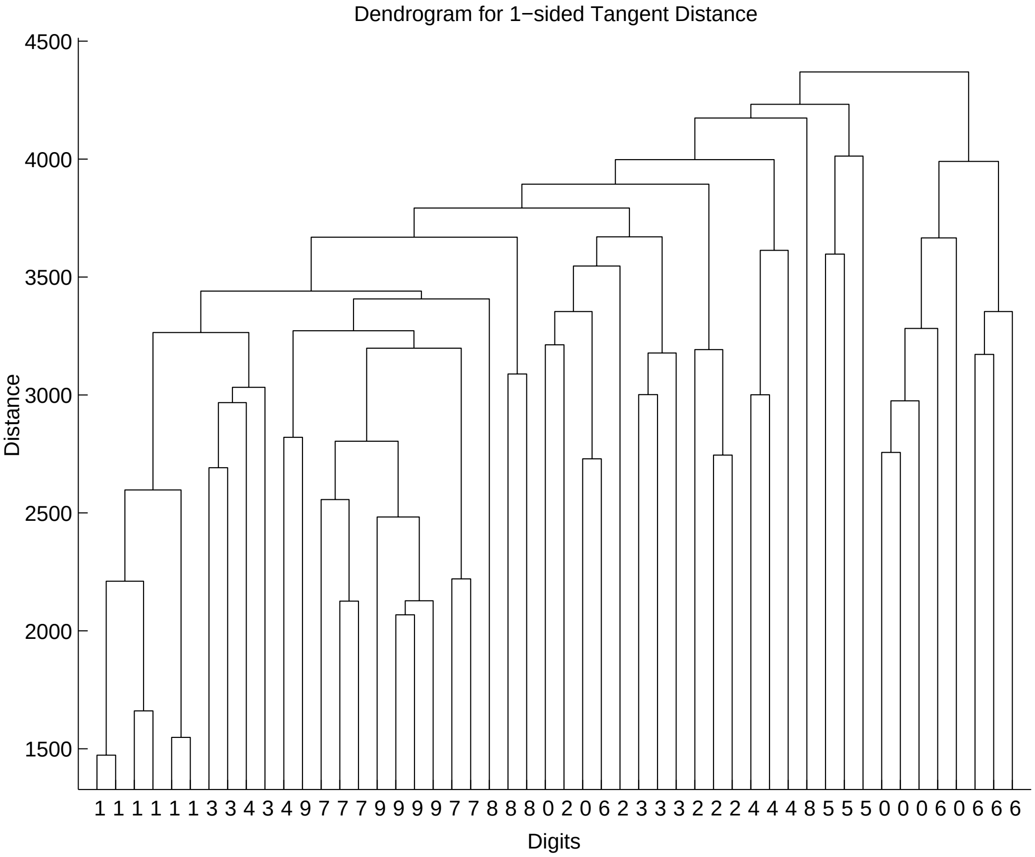

FIGURE 24. Dendrogram with 1-sided tangent distance. The labels on the bottom of the dendrogram represent digits. The numbers that are linked together resemble the most.

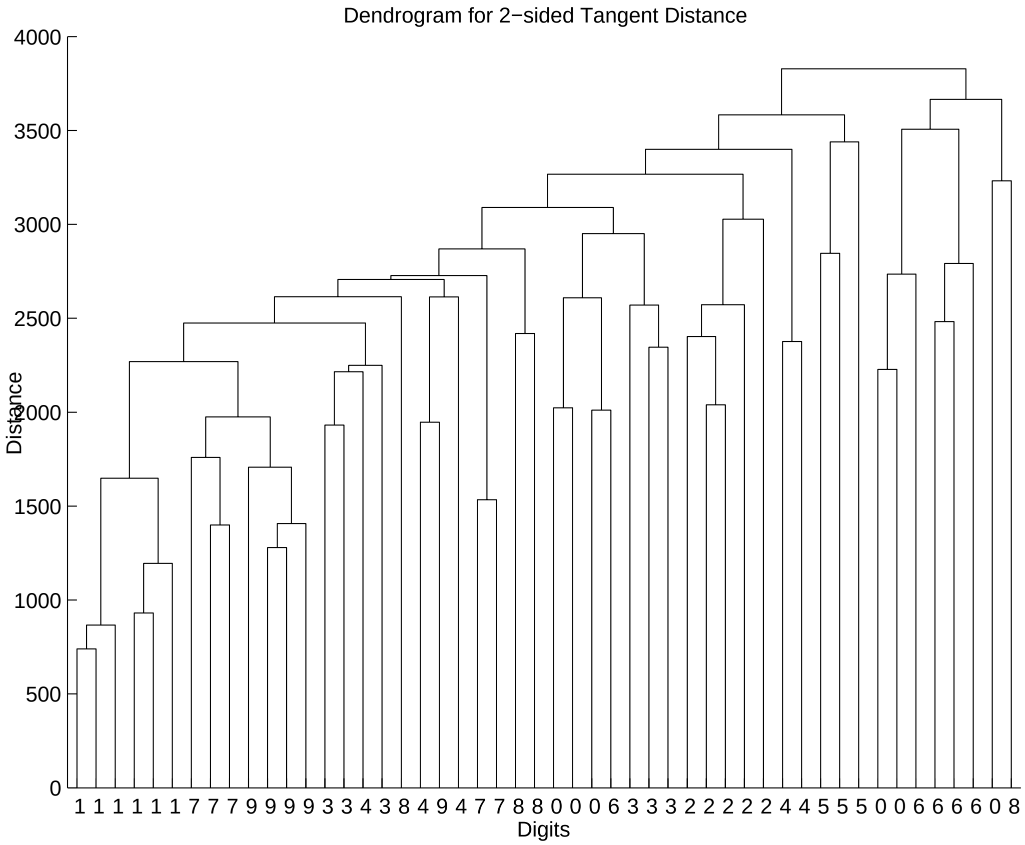

Figure 25. Dendrogram with 2-sided tangent distance. The labels on the bottom of the dendrogram represent digits. The numbers that are linked together resemble the most.