Classifying Handwritten Digits on the Grassmann Manifold

Jen-Mei Chang and José Israel Pacheco*
Department of Mathematics and Statistics
California State University, Long Beach
1250 Bellflower Boulevard
Long Beach, CA 90840-1001, USA.

Abstract

Empirical studies have shown that the collection of handwritten digits when acquired under a uniform condition forms a differentiable manifold which can be well approximated with linear structures. That is, each point on the manifold is associated with a geometry that parameterizes linear structures. Because of this, the problem of comparing a pair of digits can be turned into the problem of calculating the distance between two linear structures in their respective geometric space. In this paper, we present a new classification paradigm that builds upon the linear structure that arises from the Grassmann manifold and benchmark our empirical results on the publicly available MNIST database with two other geometrically sound methods. Without any further preprocessing, the classification performed on the Grassmann manifold achieves the best result among these three approaches.

Keywords: Geometric data analysis, handwritten digit recognition, nearest neighbor classifier, Grassmann manifold

1. Introduction

The problem of handwritten digit recognition has long been an open area in the field of pattern classification and of great importance in industry. The heart of the problem lies within the ability to design an efficient algorithm that can recognize digits written and submitted by users via a tablet, scanner, and other digital devices in real time. The applications of successful handwritten digit classification algorithms are far-reaching. For example, the post office can scan envelopes and automatically sort them by the recognized zip code and banks can automatically pick up dollar amounts from scanned checks [1]. Generally speaking, handwritten digit recognition is a subproblem of handwritten character recognition where an algorithm is needed not only to classify digits, but letters as well. Findings in the field of digit recognition can be projected to that of characters. Currently, one of the most interesting applications of such field is the ability to convert a document written by a user on a tablet into a typed document.

In this study, we will present a novel geometric approach rooted in the Grassmann manifold, a parameter space where linear subspaces reside. Under this setup, we design two algorithms for which one utilizes a *one-to-many* while the other utilizes a *many-to-many* classification paradigm [2]. These algorithms are tested on the publicly available MNIST database [3] and the classification results are benchmarked with two other geometry-driven algorithms – a subspace approach based on an optimal basis representation and a linearization approach based on a tangent approximation.

Experiments conducted in the present study assume a *nearest neighbor algorithm*. That is, given a set of training patterns $\{x_1, x_2, \ldots, x_N\}$ each belonging to a unique digit class via the map $\phi : \mathbb{R}^n \to \mathcal{C}$, where $\mathcal{C} = \{\text{“0”}, \text{“1”}, \ldots, \text{“9”}\}$ is the set of digit classes. Moreover, if the space is endowed with a metric $d(\cdot, \cdot)$, then an unknown pattern y is given the label of $\phi(x_{i'})$ if $d(y, x_{i'}) < d(y, x_i)$ for all $1 \leq i \leq N$, $i' \neq i$.

Due to the nature of the data, we use three transformations to mimic the results of human handwriting. Although factors such as thickening and thinning can be considered in the algorithm designs to improve accuracy [1], we adopted three basic affine transformations: rotation, scaling, and horizontal and vertical translation for a proof-of-concept in the present work. See Figure 1 for an illustration of the effect of such transformations applied to a digit.

We organize the rest of the paper as follows. In Section 2, we review two similar approaches where each assumes the digit manifolds are approximated by a linear space. In Section 3, we present the notion of angles in the high-dimensional spaces which are the fundamental building blocks of metrics on the Grassmann manifold. Within this

*This work is partially supported by the Graduate Research Fellowship at CSULB.

(a) (b) (c) (d) (e)

Figure 1. (a) A sample digit in MNIST. (b) A rotation of (a) by 45° in the counterclockwise orientation. (c) A stretched version of (a) by a factor of 1.5. (d) The result of (a) being translated 5 pixels to the right. (e) The result of (a) being translated 5 pixels down.

section, we discuss how the proposed algorithms can be applied to the handwritten digit classification problem. Lastly, empirical results conducted on the MNIST database are presented in Section 5 and comparisons are drawn among the three methods.

2. Background

Commonly, a gray-scale image, A, of resolution $m \times n$ is realized on the computer as an $m \times n$ matrix whose entries correspond to the intensity level of the respective pixel. Let $\mathbf{a}_1, \ldots, \mathbf{a}_n$ denote the columns of the matrix A where each $\mathbf{a}_i$ is a vector in $\mathbb{R}^m$. Such array representation of an image can be turned into a *vector* representation if we concatenate the columns of A so that

$$A = [\mathbf{a}_1 | \ldots | \mathbf{a}_n] \rightsquigarrow \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}.$$

Under this structure, we can realize monochrome images as points in their resolution space. If multiple images of a single object class is needed for processing, a data matrix can be used to store such information where each column (or row) of the matrix represents a distinct image or pattern. Throughout the discussions, it is assumed that data matrices are such that each column represents a different pattern.

2.1. A Subspace Approach via SVD

A relatively straightforward, yet effective algorithm based on a subspace representation of digits was presented in [4]. The underlying assumption is that each digit is a vector in a subspace with other digits of its kind yielding a total of ten distinct digit subspaces. For example, the optimal bases for the "1"-space, in the least square sense, turned out to be the left singular vectors of the data matrix where each column of the matrix consists of a distinct image of the digit "1". Mathematically, if X is the data matrix for the "1"-space, then its *Singular Value Decomposition (SVD)* yields

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

where the columns of U, known as the left singular vectors of X, form an orthonormal basis for the column space of X. The subspace dimension, k, of such representation is typically given by the *numerical rank* (r), which can be obtained through an energy calculation and is almost always much smaller than the resolution dimension (d) of the images. Roughly speaking, this implies that each data point, originally realized in $\mathbb{R}^d$, can be projected down to $\mathbb{R}^k$, $k \leq r \ll d$, without losing too much critical information. For example, we can represent all data points that are digit "2" in the MNIST training set with 97% of the information retained when $k = 10 \ll n = 784$.

An off-line singular value decomposition is performed on the training set for each digit and the corresponding optimal basis is stored. An on-line classification of a novel pattern, P, is done first by calculating its distance to each of the digit spaces followed by a nearest neighbor classification. That is, the pairwise distance between P and each of the digit subspace $S^{(i)}$ is given by

$$d(P, S^{(i)}) = \min_{\alpha^{(i)} \in \mathbb{R}^k} \|U_k^{(i)} \alpha^{(i)} - P\|_2, \qquad (1)$$

where $1 \leq i \leq 10$ and $U_k^{(i)}$ denotes the first k columns from the SVD of the i^{th} digit's data matrix ordered decreasingly by the magnitude of the singular values. A (gradient) descent-based analysis yields the solution in Equation (1) for $\alpha^{(i)} = {U_k^{(i)}}^T P$. Note that the final choice for k, the number of singular vectors necessary to retain 97% energy, is taken to be the maximum over all the digit subspaces. A single point-to-subspace distance calculation for an image of size $m \times n$ requires $4mnk + 2mn - k$ flops, making such algorithm fairly efficient. For more details, readers are referred to [5, 4].

2.2. A Tangent Space Model

If we imagine the aforementioned approach as a *global* method, then the work proposed by Simard *et al.* [1] would be considered as a *local* method. In their approach, every digit is assumed to lie on a high-dimensional manifold and associated with its tangent space. The notion of *tangent distance* is incorporated for finding the pairwise distance between patterns. Precisely, the tangent distance between two patterns P and E is found by calculating their respective tangent space T_P and T_E followed by the minimization problem

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

A pictorial illustration is shown in Figure 2 with a contrast to the conventional Euclidean distance. Under this setup, pattern E is contained in S_E, the set of points obtained via a collection of allowable transformation, i.e.,

$$E \in S_E = \{x \mid x = s(E, \alpha) \text{ for some } \alpha\},$$

where $s(E, \alpha)$ is the result of transforming E via the parameter α. Similarly,

$$P \in S_P = \{x \mid x = s(P, \alpha) \text{ for some } \alpha\},$$

where $s(P, \alpha)$ is the result of transforming P via the parameter α. The ways for which the tangent spaces are formed are beyond the scope of this paper. Readers who are interested in such details are referred to [1, 6]. Note that a fundamental difference between the SVD model and the tangent distance model is the overall number of pairwise distances that are computed. Although the tangent space for each digit in the training set is computed and stored off-line, a tangent distance between a novel pattern and *every* point in the training set is carried out and stored before classification can take place. Such exhaustive approach is what causes the decrease in algorithm efficiency.

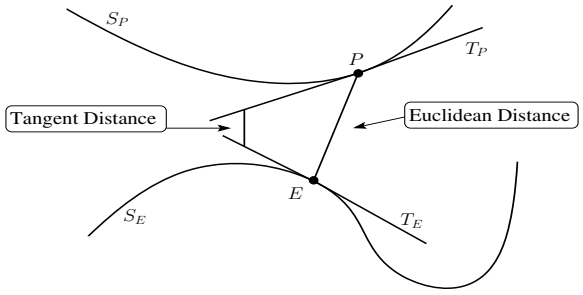


Figure 2. A comparison between the tangent distance and Euclidean distance between two digits, E and P. S_P is the underlying manifold in which P lives while S_E is the underlying differentiable manifold where E is found.

3. Classification on the Grassmann Manifold

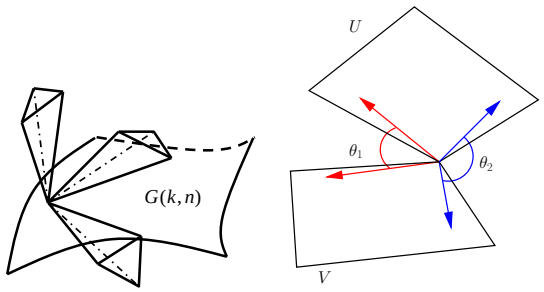
Taking advantage of the success accomplished in the area of face recognition [7], we examine the effect of the handwritten digit recognition done on the Grassmann manifold in the present study. Next, we describe in details how the classification is done on this manifold.

Let k (generally independent) images of a given digit be grouped together to form a data matrix X with each image stored as a column of X. If the column space of X, $\mathcal{R}(X)$, has rank k and if n denotes the image resolution, then $\mathcal{R}(X)$ is a k-dimensional vector subspace of $\mathbb{R}^n$, which is a point on the Grassmann manifold $G(k, n)$.

Specifically, the real Grassmann manifold (Grassmannian), $G(k, n)$, parameterizes k-dimensional vector subspaces of the n-dimensional vector space $\mathbb{R}^n$. Its precise mathematical definition is given in Definition 3.1 and a pictorial illustration is shown in Figure 3(a).

Definition 3.1. The Grassmann Manifold, denoted $G(k, n)$, is the set of k-dimensional subspaces in $\mathbb{R}^n$,

$$G(k, n) \quad = \quad \{W \subset \mathbb{R}^n \mid \dim(W) = k\}. \qquad (3)$$



(a) (b)

Figure 3. (a) Points on the Grassmann manifold are subspaces. (b) Principal angles are found recursively with the first principal angle being the smallest in the collection.

Naturally, this parameter space is suitable for subspace-based algorithms. In the case of handwritten digit recognition, by realizing sets of images as points on the Grassmann manifold, we can exploit the geometries imposed by individual metrics (drawn from a large class of metrics) in computing distances between these sets of images.

It turns out that any attempt to construct a unitarily invariant metric on $G(k, n)$ yields something that can be expressed in terms of the *principal angles* [8]. For convenience, we include a recursive definition of the principal angles here.

Definition 3.2. [9] Let U and V be subspaces in $\mathbb{R}^n$ such that $p = \dim(U) \geq \dim(V) = q \geq 1$. Then the **principal angles** $\theta_k \in [0, \pi/2]$ for $k = 1, \ldots, q$ between U and V are defined recursively by

$$cos(\theta_k) \quad = \quad \max_{u \in U} \max_{v \in V} |u^T v| = |u_k^T v_k|$$

subject to $||u||_2 = ||v||_2 = 1$, $u^T u_i = 0$, and $v^T v_i = 0$ for $i = 1, \ldots, k-1$.

To explain this definition more thoroughly, suppose we are looking for the first principal angle, θ_1. We must search through all of U and V to find the unit vectors that maximize the projection $|u^T v|$, or equivalently the vectors with the smallest angle between them. These vectors will be u_1 and v_1. To find θ_2, we again look for vectors in U and V to maximize the projection, but now our search is restricted to the orthogonal complement of u_1 and v_1 in U and V, respectively (see e.g., Figure 3(b)). Thus, in general, in order to find θ_k we must search in the orthogonal complements of span$\{u_1, \ldots, u_{k-1}\}$ and span$\{v_1, \ldots, v_{k-1}\}$, respectively. Just as an angle is a measure of the separation between two vectors, principal angles measure the separation between two subspaces.

Algorithm 1 gives a numerically stable algorithm for computing the cosine of the principal angles between two subspaces $\mathcal{R}(A)$ and $\mathcal{R}(B)$ based on the recursive algorithm given by Björck and Golub [9].

Algorithm 1 [9] Large Principal Angles

Inputs: matrices A (n-by-p) and B (n-by-q).

Outputs: cosine of the principal angles between subspaces $\mathcal{R}(A)$ and $\mathcal{R}(B)$, C.

1. Find orthonormal bases Q_a and Q_b for A and B such that $Q_a^T Q_a = Q_b^T Q_b = I$, $\mathcal{R}(Q_a) = \mathcal{R}(A)$, and $\mathcal{R}(Q_b) = \mathcal{R}(B)$.
2. Compute the SVD of $Q_a^T Q_b$: $Q_a^T Q_b = UCV^T$, so that $\text{diag}(C) = \cos\theta$.

Table 1. Table of Grassmannian distances explored in the current study.

Metric Name	Mathematical Expression
Fubini-Study	$d_{FS}(U,V) = \cos^{-1}(\Pi_{i=1}^{q} \cos\theta_i)$
Chordal 2-norm	$d_{c2}(U,V) = \|\|2\sin\frac{1}{2}\theta\|\|_\infty$
Chordal F-norm	$d_{cF}(U,V) = \|\|2\sin\frac{1}{2}\theta\|\|_2$
Geodesic	$d_g(U,V) = \|\|\theta\|\|_2$
Chordal	$d_c(U,V) = \|\|\sin\theta\|\|_2$
Projection 2-norm	$d_{p2}(U,V) = \|\|\sin\theta\|\|_\infty$

Various *Grassmannian distance* measures are realized when a different topology of the Grassmann manifold is given along with the appropriate metric. For example, if one restricts the usual Euclidean distance function on $\mathbb{R}^{n^2+n-2}/2$ to the Grassmann manifold under the realization

$$G(k,n) \subset \mathbb{R}^{n^2+n-2}/2 \tag{4}$$

via an embedding described in [10], then the appropriate distance measure in this setting is the *chordal* distance, d_c (so called because the image of the Grassmann manifold under (4) lies in a sphere, so that the restricted distance is simply the distance along a straight-line chord connecting one point of that sphere to another.), which in terms of the principal angles, has the expression $d_c(U,V) = \|\sin\theta\|_2$.

Table 1 lists the metrics investigated in the current study. $\theta = (\theta_1, \theta_2, \ldots, \theta_q)$ denotes the principal angle vector between vector spaces U and V with $dim(U) = p \geq dim(V) = q$. The results of using these metrics on a face recognition problem under variation of illumination can be found in [11].

Under this framework, we proposed two algorithms for classifying handwritten digits on the Grassmann manifold. The first is a *single-to-many* approach coined as *vector-to-subspace* algorithm in the subsequent discussions. In this point of view, we assume that each digit in the training set $\{x_1, \ldots, x_N\}$ is associated with a subspace $S^{(i)}$ found by a way discussed later. When an unknown digit, y, is presented to the system, *the* principal angle between y and each $S^{(i)}$ is found. y is then classified based on its pairwise angle $\theta(y, S^{(i)})$ with the nearest neighbor classifier.

The second proposed algorithm further assumes that the unknown digit y is also associated with a subspace $S^{(y)}$. Classification of y is based on the Grassmannian distance, $d(S^{(y)}, S^{(i)})$, between the subspace associated with y and the subspace associated with every point in the training set. We consider this type of comparison as a *many-to-many* paradigm and refer this algorithm as *subspace-to-subspace*.

These two algorithms are cast under the Grassmann framework, thus it is natural to assume that each data point is associated with a subspace. However, an interesting question that arises is that which *subspace* is appropriate for the handwritten digit recognition problem? For example, if data points $\{x_i\}$'s are relatively nearby when measured with the metric $d(\cdot,\cdot)$, then a subspace, $S^{(i)}$, associated with x_i can be formed by taking the linear span of the points that fall within a fixed distance around x_i, i.e., $S^{(i)} = \text{span}\{z \mid d(z, x_i) \leq d_0\}$ for a given constant threshold d_0. On the other hand, if data points are not nearby by, we can manually create a subspace about a data point by taking the linear span of all data points obtained via a set of allowable transformations. We describe how this is done next.

Let $r(x,\theta)$ denote the resulting image of rotating x counterclockwise by θ, $s(x,\alpha)$ denote the resulting image of scaling x by a factor of $\alpha \neq 0$, $h(x,\beta)$ denote the resulting image of translating x horizontally by β pixels, and $v(x,\gamma)$ denote the resulting image of translating x vertically by γ pixels. These four operations make up what we mean by *allowable* transformations. Now, for a pattern y, $r^{(y)} = \{z \mid z = r(y,\theta) \text{ for some } \theta_1 \leq \theta \leq \theta_2\}$ is the set of points obtained when y is rotated by an angle within a specified range. Similarly, we can obtain a set of transformed images around y for each of the other three operations; namely, $s^{(y)}$, $h^{(y)}$, and $v^{(y)}$. Finally, a subspace that is associated with y is then the linear span of the set $r^{(y)} \cup s^{(y)} \cup h^{(y)} \cup v^{(y)}$. Note that the threshold values used in the experiments are $\theta_1 = -\pi/8$, $\theta_2 = \pi/8$, $\alpha_1 = 0.8$, $\alpha_2 = 1.2$, $\beta_1 = -5$, $\beta_2 = 5$, $\gamma_1 = -5$, and $\gamma_2 = 5$. In particular, we allow ten transformations of each kind, i.e., $|r^{(y)}| = |s^{(y)}| = |h^{(y)}| = |v^{(y)}| = 10$.

Algorithms 2 and 3 provide a full description of how the two algorithms are implemented in Section 4. Since we perform a total of 40 transformations to construct $X^{(i)}$ and $X^{(P)}$, their sizes are both 784×40 resulting in the use of 40 principal angles in calculating the distance between subspaces.

It is worth noting that the transformed images of the training digits as well as their orthornormal basis can be computed *a-prior* off-line to increase algorithm efficiency. In such cases, only Steps 3–4 of Algorithm 2 are done during an on-line classification routine. The similar strategy goes for Algorithm 3.

Algorithm 2 *vector-to-subspace* Algorithm

Inputs: An unknown pattern, P; θ_1, θ_2, α_1, α_2, β_1, β_2, γ_1, and γ_2.

Outputs: Classification of P as one of the digits in $\mathcal{C} = \{\text{“0”}, \ldots, \text{“9”}\}$, i.e., $\phi(P)$.

1. For each x_i in the training set $\mathcal{T} = \{x_1, \ldots, x_N\}$, find $r^{(x_i)} = \{z \mid z = r(x_i, \theta)$ for some $\theta_1 \leq \theta \leq \theta_2\}$, $s^{(x_i)} = \{z \mid z = s(x_i, \alpha)$ for some $\alpha_1 \leq \alpha \leq \alpha_2\}$, $h^{(x_i)} = \{z \mid z = h(x_i, \beta)$ for some $\beta_1 \leq \beta \leq \beta_2\}$, and $v^{(x_i)} = \{z \mid z = v(x_i, \gamma)$ for some $\gamma_1 \leq \gamma \leq \gamma_2\}$.
2. Let $X^{(i)}$ be the vector quantization of the set $r^{(x_i)} \cup s^{(x_i)} \cup h^{(x_i)} \cup v^{(x_i)}$, i.e, $X^{(i)} = [r^{(x_i)}|s^{(x_i)}|h^{(x_i)}|v^{(x_i)}]$. Find orthonormal basis Q_i such that $\mathcal{R}(Q_i) = \mathcal{R}(X^{(i)})$ and $Q_i^T Q_i = I$.
3. Calculate the principal angle, $\theta(P, X^{(i)})$, between P and $\mathcal{R}(X^{(i)})$ using Algorithm 1.
4. $\phi(P) \leftarrow \phi(x_i')$ if $\theta(P, X^{(i')}) < \theta(P, X^{(i)})$ for all $1 \leq i \leq N, i' \neq i$.

Algorithm 3 *subspace-to-subspace* Algorithm

Inputs: An unknown pattern, P; k, number of principal angles used; d, the Grassmannian distance chosen; θ_1, θ_2, α_1, α_2, β_1, β_2, γ_1, and γ_2.

Outputs: Classification of P as one of the digits in $\mathcal{C} = \{\text{“0”}, \ldots, \text{“9”}\}$, i.e., $\phi(P)$.

1. For P and each x_i in the training set $\mathcal{T} = \{x_1, \ldots, x_N\}$, find their corresponding sets of transformed images as shown in Step 1. of Algorithm 2 and form the corresponding matrix of transformation $X^{(i)}$ and $X^{(P)}$, respectively.
2. Find orthonormal basis Q_i for x_i and Q_P for P such that $\mathcal{R}(Q_i) = \mathcal{R}(X^{(i)})$ & $Q_i^T Q_i = I$ and $\mathcal{R}(Q_P) = \mathcal{R}(X^{(P)})$ & $Q_P^T Q_P = I$.
3. Calculate the principal angles $\theta = (\theta_1, \ldots, \theta_k)$ between $\mathcal{R}(X^{(P)})$ and $\mathcal{R}(X^{(i)})$ using Algorithm 1 and their Grassmannian distance $d(P, x^{(i)})$.
4. $\phi(P) \leftarrow \phi(x_i')$ if $d(P, x^{(i')}) < d(P, x^{(i)})$ for all $1 \leq i \leq N, i' \neq i$.

4. Empirical Results

We tested the proposed *vector-to-subspace* and *subspace-to-subspace* methods on the MNIST database [3] along with the SVD- and tangent space-based models for comparison. MNIST is a database of handwritten digits collected by Yann LeCun of the Courant Institute at New York University and Corinna Cortes of Google Labs, New York. See Figure 4 for a sample of images from the MNIST training set.

Table 2. Results of proposed algorithms on MNIST database benchmarked with a SVD-based and a tangent space-based model. **CR** reported here is averaged over ten trials along with the average standard deviation, σ.

Algorithm	**CR**	σ	Time (sec)
SVD [4]	87.74%	1.95%	0.0007
Tangent Space [1]	80.36%	2.06%	0.5749
vector-to-subspace	*90.07 %*	*0.80%*	*0.0305*
Fubini-Study	42.12%	1.94%	0.7134
Chordal 2-norm	51.20%	1.24%	0.7134
Chordal F-norm	82.00%	1.54%	0.7134
Geodesic	81.84%	1.62%	0.7134
Chordal	82.48%	1.46%	0.7134
Projection 2-norm	51.72%	1.68%	0.7134

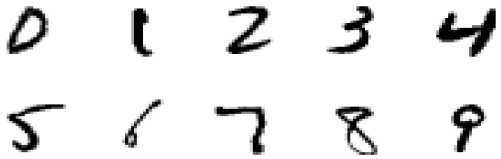


Figure 4. Sample digits from MNIST's training set.

The database consists of $60,000$ training digits and $10,000$ testing digits, each with a uniform size of 28×28 pixels and centered based on image's center of mass. No other preprocessing was applied to the images besides from the two just mentioned. In order to produce classification results in a timely manner, we randomly select 50 images of each digit from the training set to form a smaller training set. We then test the algorithm on the first 50 images of each digit in the testing set. Overall, we are using 500 images from the training set and testing against 500 from the testing test. The measurement used to determine the effectiveness of the algorithms is the commonly used *Classification Rate* (**CR**), defined to be

$$\textbf{CR} = \frac{\text{Number of True Positives}}{\text{Number of Classifications}}.$$

Each algorithm is repeated ten times to compile statistics, each time using a different set of images for training while keeping the same testing set across algorithms. In Table 2, we report the average **CR** of each algorithm along with the corresponding average standard deviation and the time it takes to classify one digit in seconds. The algorithms were executed on a platform with a 2GHz CPU and 1GB of RAM.

Since a primary goal of this paper is to provide an alternative platform for classifying handwritten digits, no significant efforts were made to optimize algorithm efficiency in the smaller scale. Rather, the experiments designed during this study are meant to serve as a proof-of-concept in demonstrating the feasibility of the proposed algorithms. Having said that, the classification rate reported here for the tangent space model was obtained under no further preprocessing; however, it is strongly recommended in [1] and [5] that one preprocesses the images with a smoothing filter, particularly in the Tangent Space algorithm, in order to achieve a more desired classification outcome. Furthermore, transformations beyond the ones used in the current study were also implemented to achieve the results reported in [1]. It is worth mentioning that the tangent space model was shown to be successful in handing other types of data set such as face images acquired under variations of illumination [12].

While the SVD-based algorithm achieves the best overall time, the proposed *vector-to-subspace* accomplishes the best accuracy without compromising much in speed. With the advent of cloud computing, it is fair to say that accuracy is more likely to outweigh speed in the determination of future algorithm designs. And the practices described here serves as a jumping-off point for pattern classification problems that are natural with a set-to-set paradigm.

5. Summary and Future Work

This paper presents a novel platform for classifying handwritten digits. The success of the work builds on the notion that variations in the state of an object can provide discriminatory information. In particular, the nature of this information arises from local features that possess their own special characteristics under a variation of state. We proposed two algorithms in the Grassmann framework from which one achieves the best overall efficiency on the MNIST database when compared to two existing geometry-driven approaches. The work accomplished here serves as a blueprint for object recognition problems where families of patterns reside in their own characteristic subspaces.

It is reasonable to conjecture that an improved classification rate will be observed if we include a wider range of transformations when associating digits with a subspace structure; however, future research will be emphasized on improving the representation of points on the digit manifold in the context of Grasmann framework. Ideas such as the Karcher mean on the Grassmann manifold can be used to compute a reduced representation of the gallery points while still maintaining original classification outcome. With this representation our current *vector-to-subspace* algorithm would enjoy an on-line process time comparable to that of the SVD-based routine.

References

[1] P. Simard, Y. L. Cun, J. Denker, and B. Victorri, "Transformation invariance in pattern recognition - tangent distance and tangent propagation," *Imaging System Technology*, vol. 11, pp. 181–194, 2001. 1, 2, 3, 5, 6

[2] J.-M. Chang, *Classification on the Grassmannians: Theory and Applications*. PhD thesis, Colorado State University, 2008. 1

[3] Y. LeCun and C. Cortes, "The MNIST Database of Handwritten Digits." http://yann.lecun.com/exdb/mnist/, 1998. [Online; accessed 17-February-2011]. 1, 5

[4] L. Elden, *Matrix Methods in Data Mining and Pattern Recognition*. SIAM, 2007. 2, 5

[5] B. Savas, "Analyses and test of handwritten digit algorithms," Master's thesis, Linköping University, 2002. 2, 6

[6] J.-M. Chang, M. Kirby, L. Krakow, J. Ladd, and E. Murphy, "Classification of images with tangent distance," tech. rep., Colorado State University, 2004. 3

[7] J. Beveridge, B. Draper, J.-M. Chang, M. Kirby, H. Kley, and C. Peterson, "Principal angles separate subject illumination spaces in YDB and CMU-PIE," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 351–356, February 2009. 3

[8] G. Stewart and J.-G. Sun, *Matrix Perturbation Theory*. Academic Press, 1990. 3

[9] A. Björck and G. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of Computing*, vol. 27(123), pp. 579–594, 1973. 3, 4

[10] J. Conway, R. Hardin, and N. Sloane, "Packing lines, planes, etc.: Packings in Grassmannian spaces," *Experimental Mathematics*, vol. 5, pp. 139–159, 1996. 4

[11] J.-M. Chang, J. Beveridge, B. Draper, M. Kirby, H. Kley, and C. Peterson, "Illumination face spaces are idiosyncratic," in *Int'l Conf. on Image Processing & Computer Vision*, vol. 2, pp. 390–396, June 2006. 4

[12] J.-M. Chang and M. Kirby, "Face recognition under varying viewing conditions with subspace distance," in *Proc. Int'l Conf. on Artificial Intelligence and Pattern Recognition (AIPR-09)*, (Orlando, FL), pp. 16–23, ISRST, July 2009. 6