Classification of Handwritten Digits

Ricky Wong Wa Chun ID : 913374195
Edward Kang ID : 912655433

March 14, 2017

ABSTRACT

In this report, we used the centroid method and the PCA method to classify a set of handwritten digits. We found that PCA method has a higher average classification rate than the centroid method, but the average calculation time is much longer.

In addition, we found that as the length of basis increases, the classification rate increases slightly, while the calculation increases significantly.

INTRODUCTION

Computer classification of handwritten digits is a standard problem in pattern recognition.

The purpose of this paper is to classify digits using two methods, the Centroid Method and the PCA Method. Furthermore, we compared and analyzed the successful classification rate and efficiency between the two methods.

RELATED DEFINITIONS, CONCEPTS AND THEORY

There are ten classes of digit, from 0 to 9.

The **centroid** of a class of digit is the average of all the different variations of that digit in the train set.

The **2-norm**

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$$

is the familiar Euclidean distance, which will be used in the Centroid Method and the PCA Method.

The **singular value decomposition** of a matrix A is the factorization of A into the product of three matrices

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

where A is a mxn matrix and $m > n$, U is a mxn orthogonal matrix, Σ is a nxn diagonal matrix and V is a mxn orthogonal matrix. This will be used in the PCA Method.

The **residual vector** can be computed by the equation

$$\min_{\alpha_i} \left\| z - \sum_{i=1}^{k} \alpha_i u_i \right\|$$

where z represents an unknown digit, and u_i is the singular images. We can write this problem in the form

$$\min_{\alpha_i} \left\| z - U_k \alpha \right\|_2$$

where $U_k = u_1 u_2 ... u_k$. Since the columns of U_k are orthogonal, the solution of this problem is given by $\alpha = UTkz$, and the norm of the residual vector of the least squares problems is

$$\left\| (I - U_k U_k^T) z \right\|_2$$

The **Centroid Method** is a simpler method to classify digits. After computing the centroid, we can build up the defining characteristic of each class. In the next phase, by computing the 2-norm distances between an unknown digit and the centroids of all classes of digits, we can classify the smallest distance that the unknown digit to the centroid of its class.

The **PCA Method** is another method to classify digits. It uses the SVD to compute defining characteristics. we have a set of singular vectors for each class of digit by computing the SVD of each data set. In the next phase, we can classify an unknown digit by computing its residual norm. Then we choose the class with smallest residual.

ALGORITHM(S) AND THIRD-PARTY FUNCTIONS

The Centroid Method is implemented by the following algorithm:

```
function [output, success] = centroid(int,A,T)
    initialize variables L,success
    find the number of rows of the matrix
    initialize the output vector

    FOR i from 1 to number of rows
        double the input A
        initialize a 10x1 vector

        FOR k from 1 to 10
```

```
        enter the vector by calculating the norm
    END
    find the minimum index of the 10 results
    minus 1 from the index as MATLAB indexing starts at 1
    enter the result index into the output vector

    IF the index equals the input int
        number of success increases
    END
    calculate the success rate
END
```

The PCA Method is implemented by the following algorithm:

```
TRAINING
    Initialize basis length
    FOR vector k 1 to 10
        Create the matrix A consists of all the training images
        Compute SVD of matrix to get singular vector of A transposed
    END

CLASSIFICATION
function [classified, success] = pca_digit(int, A, Us)
    initialize variable L, success
    convert the matrix A to double precision
    initialize a 10x1 vector

    FOR i from 1 to number of rows
        FOR k from 1 to 10
            compute the relative residual in all ten bases
        END
        Find the minimum index of the 10 results
        Minus 1 from the index as MATLAB indexing starts at 1
        Enter the result index into the output vector

        IF the index equals the input int
            number of success increase
        END
        calculate the success rate
    END
```

DISCUSSION ON IMPLEMENTATION ISSUES

centroid.m classifies digits by using the Centroid Method. It takes 3 inputs and have 2 outputs. We have the function [outvec, SR] = centroid(int,A,T). INPUTS : int is the integer identifying the test digit, A is a n-by-784 test array, T is a 10-by-784 array T. OUTPUTS : outvec is an n-by-1 vector containing the results of classification, SR is a number for the success rate.

pca_digits.m classifies digits by using the PCA method. It takes 3 inputs and have 2 outputs. We have the function [outvec, SR] = pca_digits(int,A,T). INPUTS: int is the integer identifying the test digit, A is a n-by-784 test array, U is a 784-by-5-by-10 array, where U(:,:,i) contains the first 5 singular vectors of the (i-1)-th training digit. OUTPUTS: outvec is an n-by-1 vector containing the results of the classification for the test images. And SR is a number of success rate.

U.m computes the SVD of each set of train digits and saving it in a three dimensional array for training phase.

test.m computes both Centroid Method and the PCA method to get the success rate and running time, to compare the efficiency and correctness.

EXPERIMENT RESULTS

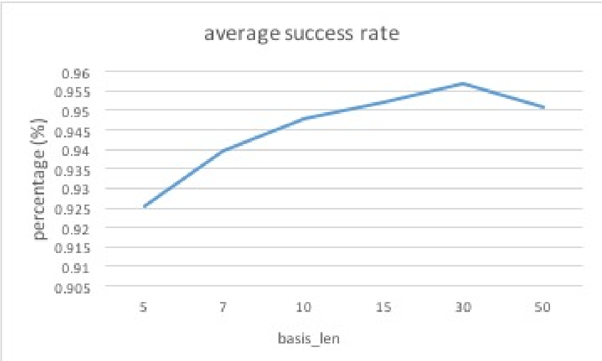
We attempted to solve a pattern recognition problem. We implement the centroid method and the PCA method for classifying a set of handwritten digits, and gave a label corresponding the class of digit that most closely resembles the imageâĂŹs shape. Our objective is comparing the successful classification rate and efficiency between the two methods.

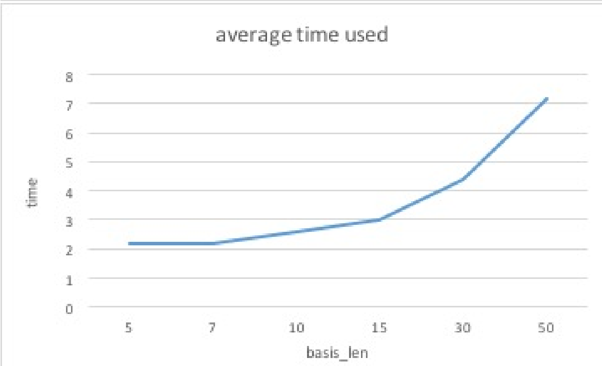
The following two tables show that the centroid method has average classification success rate of 81.726% which is not good enough to compare to PCA method with a basis length of 5 have 10.821% higher average success rate. The reason is that the centroid method does not use any information about the variation of the digits. However, the centroid method is 10,000% faster than PCA method.

The Centroid Method		
Digit	Success rate	time
0	0.8959	0.024730385
1	0.9621	0.02849958
2	0.7568	0.026772301
3	0.8059	0.025559795
4	0.8259	0.024757408
5	0.6861	0.022452213
6	0.8633	0.024124842
7	0.8327	0.026075177
8	0.7372	0.024493555
9	0.8067	0.029446995
average	0.81726	0.025691225

The PCA Method (basis_len=5)		
Digit	Success rate	time
0	0.9786	2.176090928
1	0.9921	2.195239783
2	0.9021	2.176949421
3	0.9376	2.17303538
4	0.8982	2.174186577
5	0.9013	2.159284341
6	0.9624	2.165174765
7	0.8930	2.180755177
8	0.9004	2.170297391
9	0.8890	2.175610135
average	0.92547	2.17466239

The following two graphs show that for PCA method as the length of the basis increased, the average classification success rate also increased. The reason is that as more singular vectors added to the set of train digits, the singular vectors will represent the dominating variations of the training set around the first singular vector. However, as the length increased from 5 to 50, the average success rate only slightly increased about 0.025%, and the elapsed time is significantly increased by 250%.





Concluding remarks

In conclusion, we found that PCA method has a much higher average classification success rate compare to the centroid method, but it takes more time to do the calculation. Furthermore, as the length of basis increases, the average classification success rate also increases.

However, the success rate does not increases significantly, while the calculation time increases much longer. For future study, we are going to compare these two methods to another method called k-nearest neighbor classification algorithm to find out which method is the best for classifying handwritten digits.

References

Lars Elden, Department of Mathematics, Linkoping University. "Numerical Linear Algebra in Data Mining." (n.d.): n. pag. Web. 17 Mar. 2017. <https://www.cambridge.org/core/services/aop-cambridge-core/content/view/C4E760E17B8F32847CF25C8190CCBD4A/S0962492906240017a.pdf/div-class-title-numerical-linear-algebra-in-data-mining-div.pdf>.

Mathematics, Department Of, and Davis University Of California. "MAT 167: Applied Linear Algebra Lecture 21: Classification of Handwritten Digits." (n.d.): n. pag. Web. 17 Mar. 2017. <https://www.math.ucdavis.edu/ saito/courses/167.s12/Lecture21.pdf>.

Moler, Cleve B. "5." Numerical Computing with MATLAB. Philadelphia, PA: Society for Industrial Applied Mathematics, 2011. N. pag. Print.

Appendix

centroid.m

```
function [outvec, SR] = centroid(int,A,T)
S = 0; %  number of success
SR = 0; % success rate
[n,m] = size(A); % find the number of rows of the matrix
outvec = zeros(n,1); % initialize the output vector
for i = 1 : n
    z = double(A(i,:));
    dist = zeros(10,1); % initialize a 10x1 vector

    for k = 1:10
        dist(k) = norm(z - T(k,:)); % enter the vector by calculating the norm
    end
    [M, I] = min(dist(:)); % find the minimum index of the 10 results
    I = I-1; % minus 1 from the index as MATLAB indexing starts at 1
    outvec(i) = I; % enter the result index into the output vector

    if I == int % the index equals the input int
        S = S + 1;
    end
    SR = S / n; %success rate
end
```

U.m

```
tic;
basis_len = 5;
Us=zeros( 28*28, basis_len, 10);
for k=1:10
    % go through each digit 0 to 9
    s = strcat('train',num2str(k-1));
    A = double(eval(s));

    % and get first 5 singular vector of A transposed
[U,~,~] = svds( A', basis_len );
    Us(:,:,k)=U;
end
timetrain=toc;
display(timetrain); %time for training phrase
```

pca_digits.m

```
function [outvec, SR] = pca_digits(int,A,Us)
S = 0; %  number of success
SR = 0; % success rate
[n,m] = size(A); % find the number of rows of the matrix
outvec = zeros(n,1); % initialize the output vector
for i = 1 :n
    z = double(A(i,:))';
    dist = zeros(10,1); % initialize a 10x1 vector
    for k=1:10
        Uk = Us(:,:,k);
        dist(k) = norm( z - Uk*(Uk'*z) ); % enter the vector by calculating the norm
    end
    [M, I] = min(dist(:)); % find the minimum index of the 10 results
    I = I-1; % minus 1 from the index as MATLAB indexing starts at 1
    outvec(i) = I; % enter the result index into the output vector

    if I == int % the index equals the input int
        S = S +1;
    end
    SR = S / n; %success rate
end
```

test.m

```
o1 = zeros(10,1); %success rate of PCA
o2 = zeros(10,1); %timing of PCA
o3 = zeros(10,1); %timing of centroid
o4 = zeros(10,1); %success rate of centroid
    tic;

    [classified, success]=pca_digits(0,test0,Us);
    t0=toc;
    o1(1)=success;
    o2(1)=t0+timetrain;

    tic;
    [classified1, success1]=pca_digits(1,test1,Us);
    t1=toc;
    o1(2)=success1;
    o2(2)=t1+timetrain;

    tic;
    [classified2, success2]=pca_digits(2,test2,Us);
    t2=toc;
    o1(3)=success2;
    o2(3)=t2+timetrain;

    tic;
    [classified3, success3]=pca_digits(3,test3,Us);
    t3=toc;
    o1(4)=success3;
    o2(4)=t3+timetrain;

    tic;
    [classified4, success4]=pca_digits(4,test4,Us);
    t4=toc;
    o1(5)=success4;
    o2(5)=t4+timetrain;

    tic;
    [classified5, success5]=pca_digits(5,test5,Us);
    t5=toc;
    o1(6)=success5;
    o2(6)=t5+timetrain;

    tic;
    [classified6, success6]=pca_digits(6,test6,Us);
```

```
t6=toc;
o1(7)=success6;
o2(7)=t6+timetrain;

tic;
[classified7, success7]=pca_digits(7,test7,Us);
t7=toc;
o1(8)=success7;
o2(8)=t7+timetrain;

tic;
[classified8, success8]=pca_digits(8,test8,Us);
t8=toc;
o1(9)=success8;
o2(9)=t8+timetrain;

tic;
[classified9, success9]=pca_digits(9,test9,Us);
t9=toc;
o1(10)=success9;
o2(10)=t9+timetrain;

tic;
[outvec0, SR0] = centroid(0,test0,T);
tt0=toc;
o3(1)=tt0;
o4(1)=SR0;

tic;
[outvec1, SR1] = centroid(1,test1,T);
tt1=toc;
o3(2)=tt1;
o4(2)=SR1;

tic;
[outvec2, SR2] = centroid(2,test2,T);
tt2=toc;
o3(3)=tt2;
o4(3)=SR2;

tic;
[outvec3, SR3] = centroid(3,test3,T);
tt3=toc;
o3(4)=tt3;
o4(4)=SR3;
```

```
tic;
[outvec4, SR4] = centroid(4,test4,T);
tt4=toc;
o3(5)=tt4;
o4(5)=SR4;

tic;
[outvec5, SR5] = centroid(5,test5,T);
tt5=toc;
o3(6)=tt5;
o4(6)=SR5;

tic;
[outvec6, SR6] = centroid(6,test6,T);
tt6=toc;
o3(7)=tt6;
o4(7)=SR6;

tic;
[outvec7, SR7] = centroid(7,test7,T);
tt7=toc;
o3(8)=tt7;
o4(8)=SR7;

tic;
[outvec8, SR8] = centroid(8,test8,T);
tt8=toc;
o3(9)=tt8;
o4(9)=SR8;

tic;
[outvec9, SR9] = centroid(9,test9,T);
tt9=toc;
o3(10)=tt9;
o4(10)=SR9;
```