ECS 130 Final Project

Study on the Classic Pattern Recognition Problem using Centroid Method and Principal Component Analysis Method

Shelly Hsu, Alex Kong

March 16, 2017

Abstract

The mathematical problem of this report is the comparison between the centroid method and the principal component analysis method when used to solve a pattern recognition problem. Our chosen solution is the principal component analysis because it performed better than the centroid method, and this is because computing a solution based on the variation determined by the orthogonal basis vectors was much better than simply taking the Euclidean distance between the mean of a trained digit and the given digit. The report below will go into more detail of our findings.

Introduction

Pattern recognition technology has been implemented into many components in our daily lives, such as signature capture, facial recognitions on Snapchat, and more [4]. Given a training set, it can build a representation of the data, and determine a generalization to identify more in the future. In this report, we will be exploring which of the two methods, centroid and principal component analysis, is the best when pattern recognizing handwritten found, for example, on mail envelopes, bank deposits, and more.

Related Definition, Concepts, and Theory

This report explores the concept of using matrix methods to use for pattern recognition of handwritten numerical values. According to a study done in University of Buffalo, pattern recognition is defined to be the study of methods and algorithms for putting certain data objects into categories.[1] In this report, these algorithms will be putting each

handwritten numerical value in the given data set into its respective numerical category. For example, if the algorithms define this handwritten number x to be '0', then it will be placed in the category '0', and so forth. The two algorithms used for this report are the centroid method and the principal component analysis method, and later on, a comparison between the two algorithms and their performance will be in the results section later on.

First of all, each of the data in the training set is given in the same format, a 16-by-16 gray scale images, a function s that returns (x, y) coordinates, and vectors in $\mathbb{R}^{256}$. The only format used for this method are the vectors in $\mathbb{R}^{256}$ subspace. [2] This is because we will use the Euclidean distance formula between a given data digit's vectors and the average of a digit's training vectors, which will be discussed in the algorithm section.

In contrast, the principal component analysis method, according to Victor Powell, is a "technique is used to emphasize variation and bring out the strong patterns in a dataset."[3] This method calculates the variation based on the orthogonal basis vectors, using the singular value decomposition.[2] In our report, the principal component analysis method identifies the training digit's characteristics, then uses those specific characteristics to each digit to develop the classifications for each digit.

The principal component analysis computes and singular value decomposition of all the training data for a specific digit contained in a matrix A, where each column of A is a 784-by-1 vector describing an image in the training set. An n, if $n > 0$, number of singular vectors in u, each representing a singular image, is chosen. Using linear least squares, each singular image corresponds to an equation represented with 784 coefficients for the unknown beta values that, when solved for a random b test input, produces an approximate image based on the selected singular vectors. Since there are 10 digits total, the process must be repeated independently for each digit. The solution with the minimal error is then chosen to be the identification for the unknown test case.

Algorithms

For the centroid method the algorithm runs through each of the average trained digits from 0-9, and computes the Euclidean distance between the average trained digitâĂŹs 784 elements and the given test case of 784 elements. After computing all the distances between the ten digits, the algorithm then picks the digit with the minimum distance as the identified digit. The algorithm is described as below and centroid_test.m makes calls to centroid.m for several test cases.

```
set min to infinity
set min_index = -1
for i in 1:10:
     take the calculated euclidean distances from each train digit i
     if distance < min
          min = distance
          min_index = i
return min_index
```

For the principal component analysis method, the algorithm first looks into each individual training set and picks the first 5 singular vectors from the singular value decomposition U matrix. After that it solves for the distance using the following formula [2]. This is found in pca.m and is called in pca_test.m.

```
set min to infinity
set min_index = -1
for i in 1:10:
     take the calculated euclidean distances from each train digit i
     if distance < min
          min = distance
          min_index = i
return min_index
```

Discussion on Implementation Issues

One implementation issue found was that principal component analysis is a very costly method to run. This cost is reduced in the initial test since only the first 5 singular vectors are used for computation, but to determine a more accurate result, the algorithm could potentially take more singular vectors. In fact, the algorithm can use all 28 singular vectors. Below is a table to demonstrate the times taken as we increase the number of singular vectors computed on.

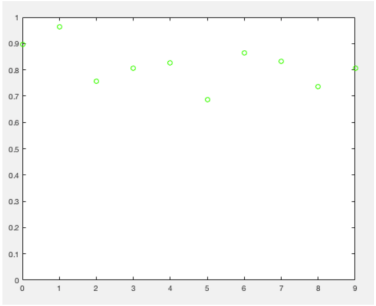
n singular vectors	5	8	28
Time (seconds)	3.9477	4.6074	8.6535

From the table above, it's shown whether or not it is worth the trade-off of more computation time for a higher accuracy in identifying digits. Later on, a residual graph is shown so that a programmer using this method can pick an appropriate n singular vectors.

Experimental Results

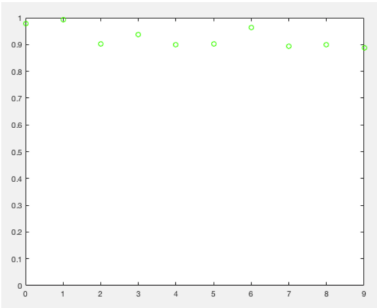
The centroid method's and principal component method's experimental results are described with the following graphs and tables.

Centroid results:



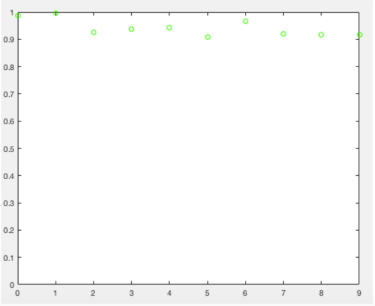
Digit	Success Rates
0	0.895918
1	0.962115
2	0.756783
3	0.805941
4	0.825866
5	0.686099
6	0.863257
7	0.832685
8	0.737166
9	0.806739

Principal Component Analysis: (basic length = 5) results:



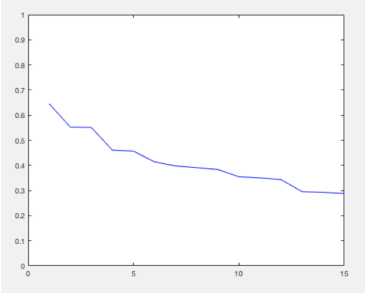
Digit	Success Rates
0	0.978571
1	0.992070
2	0.902132
3	0.937624
4	0.898167
5	0.901345
6	0.962422
7	0.892996
8	0.900411
9	0.888999

Principal Component Analysis: (basic length = 8) results:



Digit	Success Rates
0	0.985714
1	0.994714
2	0.924419
3	0.936634
4	0.943992
5	0.908072
6	0.965553
7	0.921206
8	0.916838
9	0.915758

Principal Component Analysis residuals for the Test Case 2 of Digit 0:



Concluding Remarks

From the data gathered, it is determined that principal component analysis of identifying unknown digits produces more accurate results than centroid calculation. A problem with the centroid method calculation is that the foundation of the method does not take into consideration variations, making it too simplistic in its calculation. The method assumes that minimizing the difference between individual components of the inputted digit versus the trained digit is the most important factor when it comes to identification. That is an incorrect assumption because handwritten digits could vary greatly from the trained digit. Simply computing the difference between respective individual components of the inputted digit and the trained digit may result in a higher error.

For example, the centroid method may misidentify a 5 as a 6 if the bottom of the 5 comes close to touching the top, since taking a component wise difference, and subsequently the 2-norm of that vector, would essentially show that the inputted 5 appears similar to a 6. Indeed, 5 had the worst success rates for the centroid method. Another potential source of error has to do with using the average of the training data as a comparison benchmark. Again, since handwritten numbers vary so greatly; an average of the number 1, for instance, may fail to account for those who hook the top of their 1's or add a base to their 1's. While 1 ended up having a high success rate, this factor may have played a role in the low success rates of other numbers.

For this reason, a more accurate method relying on more information must be used. Here lies the advantage of the principal component analysis method. The principal component analysis does not rely simply on the trained data of a digit itself, but rather on its most dominant characteristics. By computing an singular value decomposition decomposition and picking the n^{th} most dominant singular vectors, the most dominant characteristics of each digit is selected. Thus, the program has a better idea about the characteristic of a digit because it accounts for variation between the training data when computing a digit's singular vectors [2]. As an example, the 2^{nd} and 3^{rd} singular images, which are represented by the 2^{nd} and 3^{rd} column vectors in U after the singular value decomposition is computed, of the digit 3 are very different [2]. This is most likely attributed to the fact that handwritten 3 can appear unexpectedly different. The linear least squares method thus relies on data from a variety of sources, like the n^{th} most dominant singular vectors; it will thus produce a solution that better fits with the variety of trained data for a digit.

As is evidenced from the above graphs, taking more vectors from U to form our approximate basis leads to more accurate results. Taking one particular example, test 2 of the digit 0's residuals, or the difference between the observed and the estimated values computed as a quotient of 2-norms, decrease as more singular vectors from U are included. It can be expected that, with more data, the linear least squares fit will be better since there are more singular vectors, which are equations, to fit the solution with. However, the benefits of including more and more vectors in U become marginal, as evidenced

by the decreasing difference between residuals of consecutive numbers of U chosen. For example, choosing between 1 and 2 singular vectors from U would make large difference, but not as much for 13 and 14. Thus, it is expected that taking 14 singular vectors from U to produce more or less similar results as would taking 13 singular vectors. When more vectors from U's are picked, more time and space must be allocated for the computations.

Another consideration is that with too many images, the system may become overdetermined, which may actually lead to a less accurate solution being found. For very large numbers of U vectors, the benefits seem marginal. For any particular case, it seems best to take a number of U singular vectors reasonably large enough to produce a small residual, yet reasonably small enough to not take too much time or space or overdetermined the system.

Success rate table = best rate = mid rate = worst rate

Digit	0	1	2	3	4	5	6	7	8	9
Centroid	0.895918	0.962115	0.756783	0.805941	0.825866	0.686099	0.863257	0.832685	0.737166	0.806739
PCA 5	0.978571	0.992070	0.902132	0.937624	0.898167	0.901345	0.962422	0.892996	0.900411	0.888999
PCA 8	0.985714	0.994714	0.924419	0.936634	0.943992	0.908072	0.965553	0.921206	0.916838	0.915758

References

1. Buffalo, University At. "UB - University at Buffalo, The State University of New York." Pattern Recognition, Machine Learning, and Data Mining - UB Computer Science and Engineering. University of Buffalo, n.d. Web. <http://www.cse.buffalo.edu/research/areas/pattern.php>

2. Elden, L. Matrix Methods in Data Mining and Pattern Recognition. Philadelphia,PA: Society for Industrial and Applied Mathematics, 2007. Print.

3. Powell, Victor. "Principal Component Analysis Explained Visually." Explained Visually. N.p., n.d. Web.<http://setosa.io/ev/principal-component-analysis/>

4. "Pattern Recognition." Pattern Recognition | Pattern Recognition Laboratory. Delt University of Technology, n.d. Web.<http://prlab.tudelft.nl/content/pattern-recognition>