

A Simple Statistical Method for Recognition of Hand-Written Numerals

Arka P. Ghosh
Statistics Department
University of North Carolina
Chapel Hill, NC 27599-3260,
apghosh@email.unc.edu *

October 12, 2002

Abstract

We explore a simple statistical method for classifying handwritten numerals. We work with numerals only in segmented form. We defined a pseudo-distance function between handwritten numerals and classify each numeral based on its distance from the those in the reference set. This method, and its obvious variants, are studied on the basis of their performance on a data-set of moderate size (95 observations). The results suggest that this method may be considered in recognition of handwritten numeric characters.

1 Introduction

Optical Character Recognition (OCR), more specifically handwriting recognition, is a complex subject that has received much attention over the past 35 years. In the hand printed character recognition field, significant research efforts have been made [1, 2, 3, 4, 5]. Broadly these classifiers are of two types : Statistical Classifiers and Neural Network Classifier. These two classifiers are closely related [9]. Statistical techniques are based on the idea of estimating classconditional likelihoods and using Bayes rule to convert these to posterior class probabilities [10, 11, 8]whereas neural techniques estimate directly the posteriors[6, 7]. Of the two types of classifiers, the most popular ones are *Principal Component Analysis (PCA) based methods* (statistical classifier) and *different variants of Neural Network (NN) methods* . PCA methods are based on the following principle: all handwritten numbers will form a distinct distribution in the feature-space and that within that distribution, particular number classes will have their own distributions. PCA seeks the directions in the input space along which most of the image variation lies. Images are then projected into this reduced-dimensional space. Training images are used to compute cluster centroids, or averages, in this space; running the system on unseen images involves projecting and grouping the data to the nearest centroid. NN classifiers are work in somewhat similar fashion, but does not have good geometric interpretation like PCA methods. In a simple neural network classifier, each input pixel value contributes to a weighted sum of each output units. The output unit with the highest sum indicates the class of the input character. If each input data has N pixels, these methods have $10N$ weights and 10 biases.A good survey of commonly used methods and their performances can be found in [12].

However, in spite of many years of research, optical readers that can recognize handwritten materials at a satisfactory rate are rare. In this paper, an simple intuitive method has been described, and its results are studied on a small data-set, collected by the author.

The main idea behind the method is that if two characters are similar, after suitable scaling and rotation, they should be very close to each other. To make this more concrete, a pseudo-distance function between two handwritten characters has been defined. This function is then used to find the distance of a new character(from the test-set) from the 10 sets of characters (corresponding to the 10 digits 0,1,...9)in the reference set. The new character is then classified on the basis of minimum distance from the characters in the reference set (or training set). The definition and interpretation of this pseudo-distance function is given in next section. The Classification method is then described in Section 3. Results are shown in section 4, and conclusions are made in the last section of the paper.

*This work was done under the guidance of Prof. Probal Chaudhuri, (Stat-Math Unit, Indian Statistical Institute,203, B.T. Road Kolkata 700035 INDIA). I did this work as my Masters project an M-Stat student in Indian Statistical Institute, Kolkata.

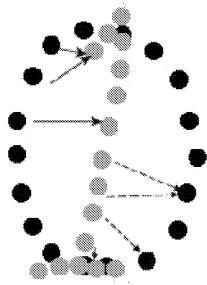


Figure 1: dissimilar figures

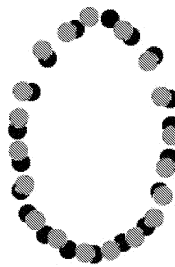


Figure 2: similar figures

2 The Pseudo-Distance Function

Let $\mathbf{B} = \{(x_i^{(B)}, y_i^{(B)}), 1 \leq i \leq n_B\}$ and $\mathbf{C} = \{(x_j^{(C)}, y_j^{(C)}), 1 \leq j \leq n_C\}$ be two set of bivariate data. Define :

$$\mathbf{d}(\mathbf{B}, \mathbf{C}) = \frac{1}{n_B + n_C} \sum_{p=1}^{n_B + n_C} d_p^*(\mathbf{B}, \mathbf{C})$$

where $d_p^*(\mathbf{B}, \mathbf{C})$ is defined as :

$$d_p^*(\mathbf{B}, \mathbf{C}) = \begin{cases} \min_{j=1,2,\dots,n_C} \|(x_p^{(B)}, y_p^{(B)}) - (x_j^{(C)}, y_j^{(C)})\| & , \text{ if } 1 \leq p \leq n_B, \\ \min_{i=1,2,\dots,n_B} \|(x_i^{(B)}, y_i^{(B)}) - (x_p^{(C)}, y_p^{(C)})\| & , \text{ if } n_B < p \leq n_B + n_C \end{cases}$$

($\|(x, y)\| = \sqrt{x^2 + y^2}$) being the usual *norm* in \mathbb{R}^2 .

In the context of character recognition, each bivariate data point represents dots (colored pixels) describing the curve(handwritten character). If two such characters are suitably normalized (scaled, centered, and rotated appropriately to bring to comparable orientation), then this function \mathbf{d} measures amount of *overlap* between them when superimposed [see fig.1, fig.2].

\mathbf{d} is zero if and only if the two characters are identical (after scaling, centering and rotation). This is symmetric also, i.e $\mathbf{d}(\mathbf{B}, \mathbf{C}) = \mathbf{d}(\mathbf{C}, \mathbf{B})$. The only reason why we call it a pseudo-distance is that it does not satisfy the triangle inequality.

But if two characters overlap, except for a few points and none of these non-overlapping points are very far from the overlapping points (as in fig.2), then \mathbf{d} takes small values. Intuitively, in this case we expect the two characters to be *similar*. If there are many non-overlapping points (compared to total number of points in two characters) at a moderate distance, or if there are few non-overlapping points far away from the overlapping parts (as in fig.1)then only \mathbf{d} takes large values. In this case we expect the two characters to be *dissimilar*. In short, this function \mathbf{d} takes large values for a pair of dissimilar characters and small value for similar characters. So we expect that this function should be able to distinguish different characters successfully.

3 The Classification method

Let $\mathbf{A}_m^{(k)}$ = m -th observation of the k -th digit = m -th bivariate data set (of size $n_m^{(k)}$) of the k -th digit, $k = 0, 1, 2, \dots, 9; m = 1, 2, \dots, N$. Use first M of the observations as the *reference set*, or, *training set* and the rest of the $N - M$ observations as the *test set* (for each of the 10 digits). To use this method, one can start with a reference set, and treat each of the new characters as an observation from the test set.

Take any observation from the *test set*: $\mathbf{A} \in \{\mathbf{A}_m^{(k)} : k = 0, 1, 2, \dots, 9; N - M < m \leq N\}$

Define distance of \mathbf{A} from the k -th digit's *reference set* as :

Method 1:

$$\mathbf{D}(\mathbf{A}, k) = \frac{1}{M} \sum_{m=1}^M [\mathbf{d}(\mathbf{A}, \mathbf{A}_m^{(k)})]$$

Method 2:

$$\mathbf{D}(\mathbf{A}, k) = \text{Median}_{m=1,\dots,M} [\mathbf{d}(\mathbf{A}, \mathbf{A}_m^{(k)})]$$

Method 3:

$$\mathbf{D}(\mathbf{A}, k) = \frac{1}{M} \sum_{m=1}^M \min_{\theta \in [-30^\circ, 30^\circ; 10^\circ]} [\mathbf{d}(\mathbf{R}_\theta \mathbf{A}, \mathbf{A}_m^{(k)})]$$

Method 4:

$$\mathbf{D}(\mathbf{A}, k) = \text{Median}_{m=1, \dots, M} \min_{\theta \in [-30^\circ, 30^\circ; 10^\circ]} [\mathbf{d}(\mathbf{R}_\theta \mathbf{A}, \mathbf{A}_m^{(k)})]$$

where \mathbf{R}_θ is the 2×2 orthogonal matrix that rotates each bivariate point by an angle θ and $\mathbf{R}_\theta \mathbf{A}$ is the set of all bivariate points in \mathbf{A} after rotation. Here, minimum is taken over all $\theta \in [-30^\circ, 30^\circ]$ with increments of 10° . The test digit \mathbf{A} (in the form of a set of bivariate data) is then classified as a member of the k_0 -th class (i.e identified as the k_0 -th digit) if

$$\mathbf{D}(\mathbf{A}, k_0) = \min_{k \in \{0, 1, \dots, 9\}} \mathbf{D}(\mathbf{A}, k)$$

That means, the test digit is identified as the k_0 -th digit ($k \in \{0, 1, \dots, 9\}$) if its average or median (with or without rotation, depending on the method used) of all the M distances ($\mathbf{d}(\mathbf{A}, \mathbf{A}_m^{(k_0)}), 1 \leq M$) from M members of the k_0 th *reference set*, is least among 10 such distances (for the 10 different digit's reference set).

4 Results

To see the performance of the 4 methods, data are collected from 95 individuals. Each individual wrote the digits 0, 1, ..., 9 in a sheet of paper. So, for each digit, we have $N = 95$ observations. $1 \leq M (= 70)$ are used as the *reference set* and rest of the $N - M = 25$ of the observations as the *test set*.

Each observation in the *reference set* goes through the following set of operations before classification starts [see figure above]:

- (1): Data sheet is scanned and stored as greyscale .bmp image.
- (2): Using mathematical software (e.g *Matlab*), the image is converted into a 2 dimensional array (dimension = pixel-dimension of the images) with each entry from 0, ..., 255 denoting the greyscale intensity of the corresponding pixel.
- (3): These matrix was converted into matrices with entries 0 or 1 (Monochrome image). Here, the cut-off used was 127, i.e any grayscale value less than 127 (darker pixels) are converted to the value 0 (denoting a black dot) and converted to 1 (no black dot) otherwise.
- (4): The 0-1 matrices are then cleaned (using *majority function* etc.) to get rid of stray dots.
- (5): The observation (now, a 0-1 matrix) is then converted to a set of bivariate data points denoting the coordinates of the black dots (positions of 0's in the 0-1 matrix).
- (6): The set of bivariate data-points are then standardized (coordinate-wise) with respect to the coordinate-wise means and variances of that data set.

After doing this, the observation (in the form of a set of bivariate points) are now comparable with any other observation which goes through the same set of operations above.

Any observations from the *test set* also go through the same set of operations before it gets classified.

Let $\mathbf{A} \in \{\mathbf{A}_m^{(k)} : k = 0, 1, 2, \dots, 9; 71 \leq m \leq 95\}$ be any of the test observations (after going through the above 6 steps). By the method described earlier, $\mathbf{D}(\mathbf{A}, k), k = 0, \dots, 9$ are computed, which denote the distance of \mathbf{A} from the *reference set* of 10 digits. \mathbf{A} is classified as the k_0 -th digit if $\mathbf{D}(\mathbf{A}, k_0)$ is the least. Note: in 4 different methods, $\mathbf{D}(\mathbf{A}, k)$ are computed differently. The results of the 4 methods are summarized in the table below.

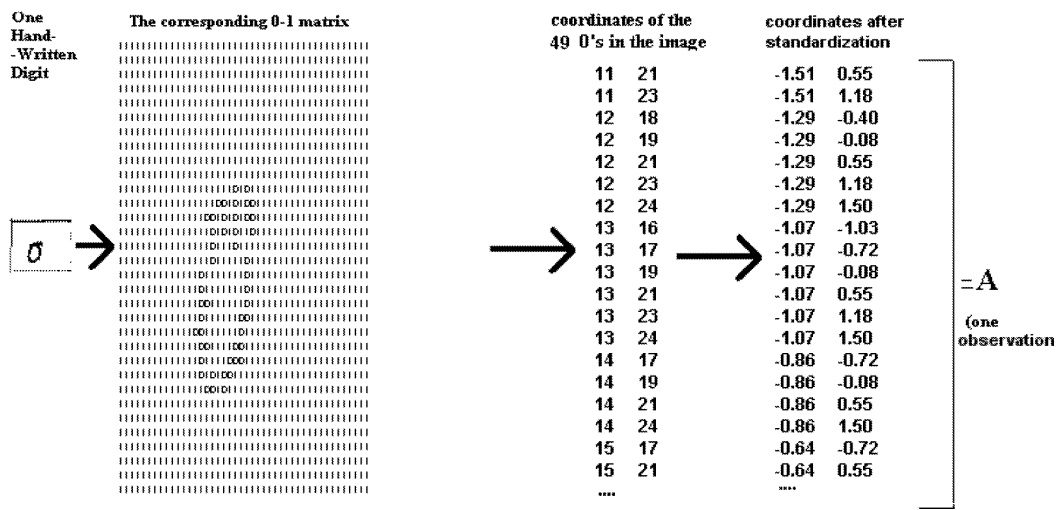


Figure 3: data processing

Performance of the four methods:

Method 1: Mean without Rotation

True digit	Identified as										Total
	0	1	2	3	4	5	6	7	8	9	
0	20	0	0	0	1	0	0	0	4	0	25
1	0	22	0	0	0	0	0	0	2	1	25
2	2	0	16	0	0	0	1	1	5	0	25
3	0	0	0	14	0	0	2	1	8	0	25
4	1	0	0	0	19	0	1	0	3	1	25
5	4	0	0	1	1	12	2	0	3	2	25
6	0	0	0	0	0	0	25	0	0	0	25
7	1	0	0	0	2	0	0	19	1	2	25
8	0	0	0	0	0	0	1	0	23	1	25
9	0	0	0	0	3	0	0	1	3	18	25
Total	28	22	16	15	26	12	32	22	52	25	250

Method 2: Median without Rotation

True digit	Identified as										Total
	0	1	2	3	4	5	6	7	8	9	
0	24	0	0	0	0	0	0	0	1	0	25
1	0	23	0	0	0	0	0	0	0	2	25
2	1	0	23	1	0	0	0	0	0	0	25
3	1	0	0	18	0	0	0	0	4	2	25
4	0	0	0	0	21	0	2	0	1	1	25
5	0	0	0	4	1	16	3	0	1	0	25
6	0	0	0	0	0	1	24	0	0	0	25
7	0	0	0	0	0	0	0	25	0	0	25
8	0	0	1	1	0	1	1	0	20	1	25
9	0	0	0	0	1	0	0	1	1	22	25
Total	26	23	24	24	23	18	30	26	28	28	250

Method 3: Mean with Rotation

True digit	Identified as										Total
	0	1	2	3	4	5	6	7	8	9	
0	20	0	0	0	1	0	0	0	4	0	25
1	0	22	0	0	0	0	0	0	2	1	25
2	2	0	18	0	0	0	0	0	4	1	25
3	0	0	1	15	0	0	2	1	6	0	25
4	0	0	0	0	21	0	2	0	1	1	25
5	2	0	0	1	2	16	3	0	1	0	25
6	0	0	0	0	1	0	24	0	0	0	25
7	0	0	0	0	2	0	0	20	2	1	25
8	0	0	0	0	1	1	1	0	22	0	25
9	0	0	0	0	4	0	0	1	1	19	25
Total	24	22	19	16	32	17	32	22	43	23	250

Method 4: Median with Rotation

True digit	Identified as										Total
	0	1	2	3	4	5	6	7	8	9	
0	24	0	0	0	0	0	0	0	1	0	25
1	0	25	0	0	0	0	0	0	0	0	25
2	1	0	24	0	0	0	0	0	0	0	25
3	0	0	0	20	0	0	0	0	4	1	25
4	0	0	0	0	23	0	2	0	0	0	25
5	1	0	0	2	0	21	1	0	0	0	25
6	0	0	0	0	0	0	25	0	0	0	25
7	0	0	0	0	0	0	0	24	0	1	25
8	1	0	1	1	0	1	0	0	20	1	25
9	0	0	0	0	1	0	0	1	0	23	25
Total	27	25	25	23	24	22	28	25	25	26	250

The percentages of misclassification can be easily obtained by multiplying all the figures in the table by 4. For example, from table 4, 96% of all the 0's in the test set have been classified correctly. Alternatively, the error percentage for 0's is 4%. The corresponding error percentages for 1's is : 0% , 2's : 4%, 3's : 20%, 4's : 8%, 5's : 16%, 6's : 0%, 7's : 4%, 8's : 20%, 9's : 8%.

5 Conclusion

A simple algorithm for digit recognition is presented in this paper. Experiments on a small data set show that the 4th method (the final variant, with median, and rotation-adjustment) does the classification with quite a satisfactory rate of accuracy.

This method, like most of the digit recognition methods, depends on fast-computing, since the complexity of the algorithm is high. The computational aspect of the method may be improved further. But this method, unlike most of the other methods in practice, is easy to interpret visually and can be easily implemented.

References

- [1] S. Mori et al. "Historical Review of OCR research and development", *Proc. of the IEEE*,80(7):1029-1058,1992
- [2] S.Impedovo, L. Ottaviano & S.Occhiegro, "Optical Character Recognition – A Survey", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 5, 1991, pp. 1-24.
- [3] J. Mantas. "An Overview of Character Recognition Methodologies." *Pattern Recognition*, 19(6):425–430, 1986.
- [4] V.K. Govindan and A.P. Shivaprasad, "Character recognition – A review," *Pattern Recognition*, vol. 23, no. 7, pp. 671-683, 1990.
- [5] R.H. Davis and J. Lyall, "Recognition of handwritten characters-a review", *Image and Vision Computing*, vol. 4, 1986, 208-218
- [6] Fukushima, K. (1988) "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition," *Neural Networks*, 1, 119–130.

- [7] Alpaydin, E., Aratma, S., Yagci, M. (1994) "Recognition of Handwritten Digits using Neural Networks," *ELEKTRİK, Turkish Journal of Electrical Engineering and Computer Sciences*, 2(1), 20–31.
- [8] S.B. Gelfand and E.J. Delp, "On Tree Structured Classifiers," *Artificial neural networks and statistical pattern recognition*, ed. I.K. Sethi and A.K. Jain, pp. 51 - 70, North-Holland, Amsterdam, 1991.
- [9] Ethem Alpaydin, Fikret Gurgen . "Comparison of Statistical and Neural Classifiers and Their Applications to Optical Character Recognition and Speech Classification " *Neural Network Systems Techniques and Applications* (in print).
- [10] McLachlan, G. J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*, Wiley.
- [11] Ripley, B. D. (1994) "Neural networks and related methods for classification," *Journal of Royal Statistical Society B*, 56, 409–456.
- [12] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Mller, E. Sckinger, P. Simard, and V. Vapnik, "Comparison of Learning Algorithms for Handwritten Digit Recognition", in F. Fogelman and P. Gallinari (eds), International Conference on Artificial Neural Networks, pp 53-60, Paris, (1995).