PERFORMANCE OF THE PERCEPTRON ALGORITHM FOR THE CLASSIFICATION OF COMPUTER USERS

S.A. Bleha, J. Knopp and M.S. Obaidat
Computer Engineering Research Laboratory
Department of Electrical and Computer Engineering
University of Missouri - Columbia

Abstract

The perceptron algorithm was used to classify computer users. Test data was collected interactively from 5 users over a 5 week period. The times between keystrokes entered in a password formed the measurement vector. Decision functions were derived using part of the data (training data) to compute the weight vectors. The decision functions were applied to the remaining data (testing data) to classify the users. Four users were classified correctly with no error, and one user was misclassified with a 10% error resulting in an overall misclassification error of 2%.

Key Words

Perceptron, Classify Computer Users.

Introduction

The progress of computer technology has provided us with various types of computers and workstations that are usually connected together in different techniques. Due to the growth of computer users and the dependence of people on computers to store and process information, the issue of automating the process of computer access verification is a vital one. There are critical data stored in the computers of some organization that accessing them by invalid user may be catastrophic. Accessing other types of data may entail loss of money or confidential information. Object recognition can easily be achieved by applying pattern recognition techniques on a set of feature vectors that represent it [1].

Although handwriting and typing are distinct manual skills they both have measurable characteristics that are unique to who performs the task. Pattern recognition techniques applied in handwriting analysis may also be appropriate to typing recognition problems [2]-[4]. Such techniques have been applied in previous work using feature vectors derived from keystroke intervals [5]-[7]. Pattern recognition techniques such as minimum distance, Bayes classification, and Fuzzy logic have been used successfully. This work is continued here. The objective is to classify typists using the perceptron algorithm to derive decision functions from keystroke interval feature vectors. This approach is easy to implement on most computer systems with simple software.

Description of the Experiment

Experiments were conducted with 5 users, using time intervals between keystrokes in a fixed message. We refer to the message as the "password." Time periods between keystrokes were collected using an IBM personal computer. An 8086/8088 assembly language program used software keyboard interrupts to measure time intervals between typed characters in the password. For example, if the password "LIST" is entered then the assembly language program computed the time durations between the character pairs (L,I), (I,S), and (S,T). The protocol adopted in collecting the data relies on having a significant period of time available between each trial. Each trial consisted of typing the password twice. The participants then waited at least one day before another trial. All participants used one keyboard. This reduced effects of uncorrelated noise and other human factors.

The phrase 'UNIVERSITY OF MISSOURI

COLUMBIA' was used as the password. This produced 30 vector components; however, only the first fifteen are used here. The data were collected interactively during the testing period of on-line recognition systems [6]. Total number of measurements vectors collected per user was forty. It took five weeks to collect the data.

Part of the data (i.e., training data) were used to calculate the weight vectors via the perceptron algorithm. The rest of the data (i.e., test data) were used for classification testing.

Linear Decision Functions

Details of linear decision functions are well described in the literature [8]-[11]. Only a brief discussion is given here to clarify notation.

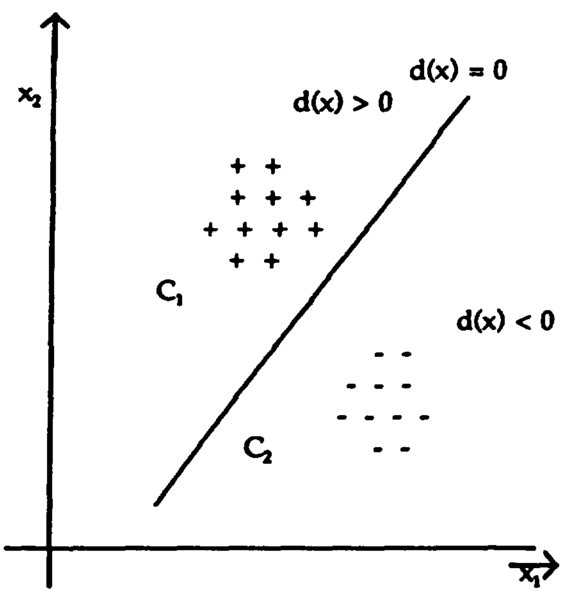

Figure 1: Two Class Case.

Consider Figure 1 (the two dimensional case), the linear decision function is a line, d(x), that separates the two classes, C_1 and C_2. Any pattern vector X in class C_1 is positive definite when substituted into d(x) and negative definite if it belongs to C_2. Therefore d(x) can be used to decide membership in class C_1 and class C_2. In order to be useful, the pattern vectors must be linearly separable.

In higher dimension spaces, d(x) is generally a hyperplane with functional form

$$\begin{aligned} d(x) &= w_1x_1 + w_2x_2 + \dots + w_nx_n + w_{n+1} \\ &= W'X \end{aligned}$$

here the vector $W = (w_1, w_2, \dots, w_{n+1})'$ is the weight vector, and $X = (x_1, x_2, \dots, x_{n+1})'$ is the augmented pattern vector. In the two-class case a decision function d(x) is assumed to have the property

$$d(x) = W'X \begin{cases} >0 \ \textit{if}\ X \in C_1 \\ <0 \ \textit{if}\ X \in C_2 \end{cases}$$

For the M-class case, we have M decision functions with the property

$$d_i(X) = W_i'X \begin{cases} >0 \ \textit{if}\ X \in C_i \\ <0 \ \textit{otherwise} \end{cases}$$

where $W_i = (w_{i1}, w_{i2}, \dots, w_{in+1})'$ is the weight vector associated with the ith decision function, and X is the pattern classified.

The Perceptron Algorithm

Once the type of the decision function is specified, the problem is finding the components of the weight vector. The perceptron learns from the training set, provided the patterns are linearly separable.

Consider the two class problem, it is desired to find a solution vector with the property that W'X>0 for all patterns of class one and W'X<0 for all patterns of class two. If the patterns of class two are multiplied by -1, we get the equivalent condition W'X>0 for all patterns. The problem is to find a weight vector W such that the system of inequalities

$$\bar{X}W > \bar{0}$$

is satisfied, where the matrix

$$\bar{X} = \begin{pmatrix} X_1' \\ X_2' \\ \cdot \\ \cdot \\ \cdot \\ X_N' \end{pmatrix}$$

and N represents the total number of augmented sample patterns.

If there exists a solution vector W which satisfies the system of inequalities given above, then the classes are linearly separable and the perceptron algorithm will converge to it in a finite number of steps. If the classes are not linearly separable, then the perceptron algorithm doesn't converge.

For the M-class case, the two-class case can be used repeatedly to derive in-class, out-of-class pairs. This means each of the classes are individually separable from the remaining classes. To determine the decision function for the ith class, we consider the two-class problem C_i and $\bar{C}_i$, where $\bar{C}_i$ denotes all classes except C_i.

Training Procedure

The perceptron algorithm finds a decision weight vector, W, iteratively. Given two training sets belonging to classes C_i and $\bar{C}_i$, respectively, choose W(1) arbitrarily to represent an initial guess at the weight vector. At the kth training step:

If $X \in C_i$ *and* $W'(k)X \leq 0$, *replace* $W(k)$ *by*

$$W(k+1) = W(k)+X$$

If $X \in \bar{C}_i$ *and* $W'(k)X \geq 0$, *replace* $W(k)$ *by*

$$W(k+1) = W(k)-X$$

Otherwise, leave $W(k)$ *unchanged, that is,*

$$W(k+1) = W(k)$$

Note W(k+1) changes if and only if the pattern being considered is misclassified by W(k).

Equivalently, the algorithm can be expressed in a simpler form if the augmented patterns of class $\bar{C}_i$ are multiplied by -1, that is,

$$W(k+1) = \begin{cases} W(k) & \text{if } W'(k)X>0 \\ W(k)+X & \text{if } W'(k)X \leq 0 \end{cases}$$

This latter form is used in our algorithm.

Results

Ten vectors out of the forty vectors for each class (or user) were used for training, (i.e., determine the weight vector W). The thirty remaining vectors were used for testing. Once the weight vector was determined, then the decision function d(X)=W'X was used to classify the test vectors using the criteria

$$\text{If } d(X) = W'X>0 \text{ then } X \in C_i$$
$$\text{otherwise } X \in \bar{C}_i$$

The case where X is classified to class $\bar{C}_i$ represents a classification error. The results for the 5 classes are in Table 1. The initial weight vector was chosen as $\boldsymbol{W} = \vec{1}$. (Other values of the weight vector were also tried with similar results, only the rate of convergence changed). The only classification error is in class one. We tried increasing the training data to twenty vectors, and then thirty vectors but the error was unchanged at 10% in class one, with a constant overall error of 2%. The tale also shows number of iterations needed to obtain the weight vector. Notice it is class dependent.

Class #	Number of Iterations	Percent Misclassification
1	167	3/30 = 10%
2	180	0/30 = 0%
3	331	0/30 = 0%
4	2022	0/30 = 0%
5	107	0/30 = 0%

Table 1: Classification results.

Conclusions

In this paper, we addressed two major questions. First, how can we characterize the difference in the way people type on a computer keyboard, in a manner suitable for classification? Second, how can we design and parametrize appropriate and effective classification system?

The use of the perceptron algorithm to provide linear decision functions to classify computer users worked surprisingly well. The algorithm was robust with respect to the choice of the initial weight vector. The use of keystroke intervals as feature vectors appears to work well in determining linear decision functions for classification. An overall misclassification error of 2% was achieved. The error was due to one user only.

The results presented here are comparable to the previous work and handwriting recognition systems.

References

[1] M.S. Obaidat, and J.W. Ekis, "An Automated System for Characterizing Ultrasonic Transducers Using Pattern Recognition," IEEE Trans. on Instrumentation and Measurement, Vol. 40, No. 5, pp. 847-850, October 1991.

[2] H. Almuallim, and S. Yamaguchi, "A Method of Recognition of Arabic Cursive Handwriting," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 5, September 1987.

[3] R. Plamondon, and G. Lorette, "Automatic Signature Verification and Writer Identification -- the State of the Art," Pattern Recognition, Vol. 22, No. 2, 1989.

[4] C.C. Tappert, C.Y. Suen, and T. Wakahara, "On-line Handwriting Recognition: A Survey," Proc. 9th Int. Conf. on Pattern Recognition, 1988.

[5] S. Bleha, and M. Obaidat, "Dimensionality Reduction and Feature Extraction Applications in Identifying Computer Users," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 21, No. 2, March/April 1991.

[6] S. Bleha, C. Slivinsky, and B. Hussien, "Computer-Access Security Systems Using Keystroke Dynamics," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 12, December 1990.

[7] B. Hussien, R. McClaren, and S. Bleha, "An application of Fuzzy Algorithms in a Computer Access Security System," Pattern Recognition Letters, Vol. 9, No. 1, January 1989.

[8] J. Tou, and R. Gonzalez, Pattern Recognition Principles, Reading, Addison Wesley, 1981.

[9] Y. Pao, Adaptive Pattern Recognition and Neural Networks, Reading, Addison Wesley, 1989.

[10] B. Soucek, Neural and Concurrent Real-Time Systems, John Wiley & Sons, 1989.

[11] D. Rumelhart, J. McClelland, and PDP Research Group, Parallel Distributed Processing, The MIT Press, 1988.