the Availability Digest

Dr. Bill's Doctoral Thesis

During my tenure at Bell Telephone Laboratories in the early 1960s, the Labs was kind enough to send me for my Doctorate in Electrical Engineering (DEE) at Brooklyn Polytechnic Institute (now the New York Institute of Technology). My work at Bell Labs involved an effort to build a system that would read handwritten numbers. At the time, long distance calls were recorded by an operator on a handwritten ticket and then were transcribed manually onto punch cards for processing by a billing program. This required a great deal of labor, and Bell Labs wanted to automate the process. My assignment was to build a system that could automatically read the operators' tickets and create the punch cards.

This was an ideal task for a doctoral thesis and led to my 211-page "Linear Decision Functions, With Application to Pattern Recognition." Recognize that back then, there were no PCs. The only computer the Labs had was a monstrous IBM 7090, IBM's first transistorized scientific computer (the IBM 7070 was its first transistorized computer for business functions). Therefore, the system that I had to implement could not use a computer.

As a reference, Hewlett-Packard in 1960 reported net sales of USD $60.2 million; and the company was first listed on the New York Stock Exchange in 1961.[1] HP entered the computer market in 1966 with the HP 2100 / HP 1000 series of minicomputers. HP Labs was formed in 1967.[2]

A pattern recognition machine consists of two principal parts – a receptor and a categorizer. The receptor makes certain measurements on the unknown pattern to be recognized. The categorizer determines from these measurements the particular allowable pattern class to which the unknown pattern belongs.

Building the Receptor

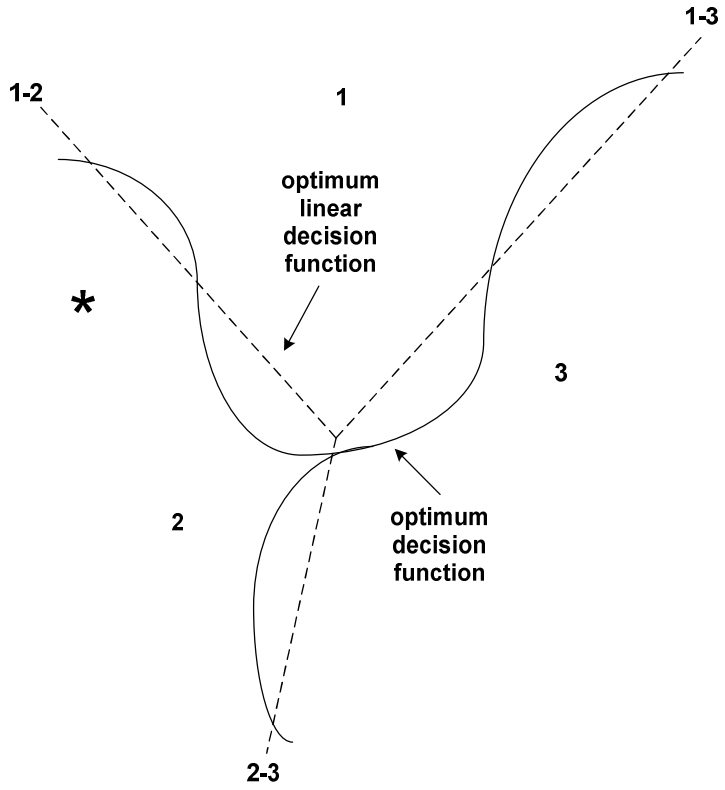
In my attempt to read hand-printed numbers, the first step was to build a receptor to convert a number to a machine-readable format. I built a simple scanner that could scan a hand-printed number and convert it to a 12 x 12 matrix of ones and zeros. Next, I recorded the matrix on a punch card (this was the reason that I used a 12 x 12 matrix – a punch card has twelve rows).

Each number then could be considered a point in a 144-dimensional space. Presumably, points representing the same number were batched together in this space. Thus, there would be ten clusters of points in the 144-dimensional space, each representing one of the numbers 0 to 9.

Implementing the Categorizer

The concept of linear decision functions is to define within the 144-dimensional space a set of hyperplanes that separate the clusters of numbers. Then, given an unknown sample, its position relative to each hyperplane can be calculated. Based on which side of each hyperplane the sample lies, the sample can be determined to be a particular digit, as shown in Figure 1.

[1] A Decade of Steady Growth, HP Memory Project; http://hpmemoryproject.org/wb_pages/wall_b_page_00.htm
[2] Hewlett-Packard, Wikipedia; https://en.wikipedia.org/wiki/Hewlett-Packard#1960s



Domains of Three Pattern Classes in Measurement Space
Figure 1

For instance, the point represented by the asterisk in Figure 1 is on the "2" side of hyperplane 1-2. Therefore, it cannot be a "1." It is on the "1" side of hyperplane 1-3 and therefore cannot be a "3." It is on the "2" of hyperplane 2-3 and therefore must be a "2."

Given that I undertook this research in the early 1960s when no compute time was available, it was important that the linear decision functions be implemented with independent circuitry. This turned out to be quite straightforward. An n-dimensional hyperplane is given by the equation

$$\sum_{i=1}^{n} a_i x_i + a_0$$

In order to classify a point m, it is only necessary to determine on which side of each hyperplane the point m lies. This is determined by the sign of the quantity

$$\sum_{i=1}^{n} a_i m_i + a_0$$

In fact, the magnitude of this quantity is proportional to the distance of the point m from the hyperplane.

This calculation can be accomplished with an inexpensive resistive adder, as shown in Figure 2. The voltage and resistor values are chosen to represent the terms in the above equation.

My thesis proved several theorems relating to linear decision functions:

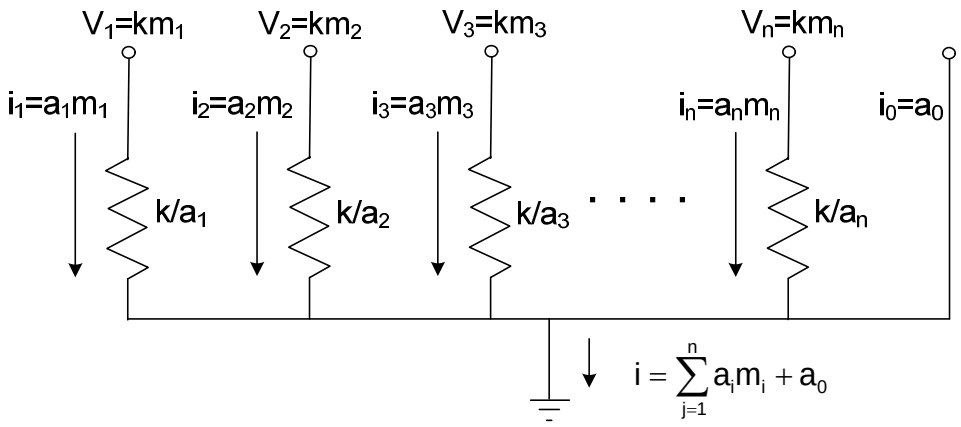
Theorem 1: For any categorizer based upon minimizing a Euclidian distance to a set of reference points, there exists a categorizer based on a linear decision function which is at least as good.

Theorem 2: For n pattern classes, a linear decision function comprises n(n-1)/2 hyperplanes.

Theorem 3: A complete linear decision function will classify any measurement into no more than one allowable pattern class.

Theorem 4: In a measurement space, the points that are identified with a particular class by a linear decision function comprise a convex set (i.e., a line joining any two points in the set is contained within the set).

Theorem 5: The categorization defined by a linear decision function is invariant under a nonsingular affine transformation on the measurement space. (My wife asked me what this meant. I told her I had no idea because I wrote my thesis fifty-five years ago. Sorry.)



Implementation of a Hyperplane
Figure 2

The next step was to determine the set of hyperplanes that would optimize the recognition of the hand-printed digits. To determine this, I obtained a set of hand-printed digits from fifty different people. Half of the samples were used to determine an optimum set of hyperplanes, and the other half were used to test the result.

Given ten digits, the number of hyperplanes that had to be determined was n(n-1)/2 = 45. A hyperplane was chosen for each pair of digits by incrementally adjusting the coefficients a_i of the hyperplane to minimize the number of digits that were incorrectly identified.

Testing the Pattern Recognition Machine

When all hyperplanes had been implemented using this procedure, a test of the resulting categorizer was made with the other half of the samples. The result was a 73% accuracy of identification.

This result clearly was not acceptable for commercial use. Therefore, Bell Labs terminated the project with the determination that linear decision functions were not appropriate for the recognition of hand-printed numbers.

A Second Attempt

In conjunction with a co-worker, Lou Kaminsky, we implemented a flying-spot scanner that broke handwritten numbers into closures, cusps, and lines. For instance, a ‘six’ is a closure on the bottom and a cusp to the right on top. A ‘three’ is two cusps to the left on top of each other. This improved the recognition results, but they still were not up to being commercially acceptable.

Fast Forward Six Decades

It is now almost six decades later. Computers are readily available, including those on microchips that can be embedded into special equipment. A great deal of research has gone into character recognition using the power of these computers. Now, not only can handwritten numbers be reliably recognized, but so can handwritten alphanumerics and even script. What a difference six decades can make.

Hewlett-Packard has been active in optical character recognition (OCR) for many years. Back in 1985, HP developed an OCR package named Tesseract. Tesseract outputs analyzed text into plain text, PDF, and HTML formats. In 2005, Tesseract was open-sourced by HP. Since 2006, Tesseract has been supported by Google.

HP also once offered OCR under HP IDOL (Intelligent Data Operating Layer) via IDOL’s Worksite OCR Module. This offering was part of the Autonomy software suite. Autonomy was a company purchased by HP in 2011. Autonomy recently was acquired by Micro Focus as part of a larger software spinoff by HPE to Micro Focus.

Summary

My thesis was one of the early attempts at recognizing hand-printed text. Although it was accepted as legitimate research by Brooklyn Polytechnic Institute, the thesis did not achieve its goals because of a lack of computer resources and reliable categorization algorithms. As time has gone on and after a great deal of further research, these limitations have been overcome. Now, even machine reading of handwritten alphanumeric script is common.