Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus

Jack W. Smith, BS[†], JE Everhart, MD, MPH[*], WC Dickson[†],
WC Knowler, MD, DrPH[**], RS Johannes, MD,MS[*††]
From the Logistics Management Institute[†],
the National Institute of Diabetes Digestive and Kidney Diseases:
Epidemiology and Data Systems Program[*] and
the Diabetes and Arthritis Epidemiology Section[**], and
The Johns Hopkins University School of Medicine[††]

ABSTRACT

Neural networks or connectionist models for parallel processing are not new. However, a resurgence of interest in the past half decade has occurred. In part, this is related to a better understanding of what are now referred to as *hidden nodes*. These algorithms are considered to be of marked value in pattern recognition problems. Because of that, we tested the ability of an early neural network model, ADAP, to forecast the onset of diabetes mellitus in a high risk population of Pima Indians. The algorithm's performance was analyzed using standard measures for clinical tests: sensitivity, specificity, and a receiver operating characteristic curve. The crossover point for sensitivity and specificity is 0.76. We are currently further examining these methods by comparing the ADAP results with those obtained from logistic regression and linear perceptron models using precisely the same training and forecasting sets. A description of the algorithm is included.

1. INTRODUCTION

The ability to forecast and discriminate is central in many medical situations. Standard statistical techniques such as discriminate analysis, regression analysis, and factor analysis have been used to provide this ability. However, even with the existence of hidden functional relationships that can provide forecasting ability, standard statistical techniques may be unsuccessful. The standard statistical methods may provide disappointing results when:

1. The sample size is small.
2. The form of the underlying functional relationship is not known.
3. The underlying functional relationships involve complex interactions and intercorrelations among a number of variables.

These conditions are not unusual in medical problems. In such situations, some of the neural network approaches offer promise.[1]

Neural network models are a class of learning algorithms that were initially described in the late 1950's.[2] Those initial algorithms have been extended recently, and interest in them has expanded significantly.[3] The algorithms are capable of using a training data set to discover patterns in data. One such algorithm, known as ADAP was developed by two of the authors [Smith, Dickson] in 1961.[4,5] We chose to examine the ADAP algorithm and test its use in forecasting the onset of non-insulin-dependent diabetes mellitus (DM) within a five-year period. The data used in this study were from the Pima Indian population near Phoenix, Arizona. Once the algorithm had been trained using 576 cases, ADAP was used to forecast whether another 192 test cases would develop diabetes within five years. Forcing ADAP to conclude on all test cases produced a sensitivity and specificity of 76 percent. A receiver operating characteristic (ROC) curve was determined. Efforts are underway to test the same data set using a logistic regression model and a linear perceptron.

2. MATERIALS AND METHODS

2.1 Study Population

The population for this study was the Pima Indian population near Phoenix, Arizona. That population has been under continuous study since 1965 by the National Institute of Diabetes and Digestive and Kidney Diseases because of its high incidence rate of diabetes.[6,7,8] Each community resident over 5 years of age was asked to undergo a standardized examination every two years, which included an oral glucose tolerance test. Diabetes was diagnosed according to World Health Organization Criteria[10]; that is, if the 2 hour post-load plasma glucose was at least 200 mg/dl (11.1 mmol/l) at any survey examination or if the Indian Health Service Hospital serving the community found a glucose concentration of at least 200 mg/dl during the course of routine medical care [7]. In addition to being a familiar database to the investigators, this data set provided a well validated data resource in which to explore prediction of the date of onset of diabetes in a longitudinal manner.

2.2 Selection of the Variables

When ADAP was applied to the diabetes problem, eight variables were chosen to form the basis for forecasting the onset of diabetes within five years in Pima Indian women. Those variables were chosen because they have been found to be significant risk factors for diabetes among Pimas or other populations.

0195-4210/88/0000/0261$01.00

Those input variables were:

1. Number of times pregnant
2. Plasma Glucose Concentration at 2 Hours in an Oral Glucose Tolerance Test (GTT)
3. Diastolic Blood Pressure (mm Hg)
4. Triceps Skin Fold Thickness (mm)
5. 2-Hour Serum Insulin (μU/ml)
6. Body Mass Index (Weight in kg / (Height in m)^2)
7. Diabetes Pedigree Function
8. Age (years)

We developed the (not yet validated) Diabetes Pedigree Function (DPF) to provide a synthesis of the diabetes mellitus history in relatives and the genetic relationship of those relatives to the subject. The DPF uses information from parents, grandparents, full and half siblings, full and half aunts and uncles, and first cousins. It provides a measure of the expected genetic influence of affected and unaffected relatives on the subject's eventual diabetes risk. The diabetes pedigree function

$$DPF = \frac{\sum_i K_i(88-ADM_i)+20}{\sum_j K_j(ALC_j-14)+50}$$

is computed for each examination using only data that were available on the date of that examination and where:

i ranges over all $relatives_i$ who had developed diabetes by the subject's examination date;

j ranges over all $relatives_j$ who had NOT developed diabetes by the subject's examination date;

K_x is the percent of genes shared by the $relative_x$ and

- equals 0.500 when the $relative_x$ is a parent or full sibling,
- equals 0.250 when the $relative_x$ is a half sibling, grandparent, aunt or uncle, and
- equals 0.125 when the $relative_x$ is a half aunt, half uncle or first cousin;

ADM_i is the age in years of $relative_i$ when diabetes was diagnosed;

ACL_j is the age in years of $relative_j$ at the last non-diabetic examination (prior to the subject's examination date);

Constants

- The constants 88 and 14 represent, with rare exception, the maximum and minimum ages at which relatives of the subjects in this study developed DM.
- The constants 20 and 50 were chosen such that:
  1. A subject with no relatives would have a DPF value slightly lower than average
  2. The DPF value would decrease relatively slowly as young relatives free of DM joined the database
  3. The DPF value would increase relatively quickly as known relatives developed DM.

Note that the value of the DPF increases as the number of relatives who developed DM increases, as the age at which those relatives developed DM decreases, and as the percentage of genes that they share with the subject increases. Also notice that the value of the DPF decreases as the number of relatives who never developed DM increases, as their ages at their last examination increase, and as the percent of genes that they share with the subject increases.

2.3 Case Selection

Diabetes was defined as a plasma glucose concentration greater the 200 mg/dl two hours following the ingestion of 75 gm of a carbohydrate solution. Cases were drawn from the pool of examinations which met the following criteria:

i. The subject was female.

ii. The subject was ≥ 21 year of age at the time of the index examination. An index examination refers to the study that was chosen for use in this model. It does not necessarily correspond to the chronologically first examination for this subject.

iii. Only one examination was selected per subject. That examination was one that revealed a nondiabetic GTT and met one of the following two criteria:

   a. Diabetes was diagnosed within five years of the examination, **OR**

   b. A GTT performed five or more years later failed to reveal diabetes mellitus.

iv. If diabetes occurred within one year of an examination, that examination was excluded from the study to remove from the forecasting model those cases that were potentially easier to forecast. In 75% of the excluded examinations, DM was diagnosed within six months.

Using these criteria, 768 examinations were selected. From those, 576 were selected randomly to be used in the training or learning set and the remaining 192 cases became the forecasting set. Our hypothesis was that ADAP could learn to forecast whether a given individual would develop diabetes mellitus within five years given the value of the eight input variables.

3. DESCRIPTION OF THE ALGORITHM

ADAP is an adaptive learning routine that generates and executes digital analogs of perceptron-like devices[9]. It can generate a variety of such analogs. However, it was designed expressly to generate special-purpose "partitioned" perceptron-like analogs tailored for the specific problem at hand.

ADAP analogs learn by making internal adjustments when their forecasts turn out to be incorrect. Assume that some phenomenon is expressed as a value or a state and you assume that it can be forecast some unknown function of other variables. The ADAP learning cycle starts by reading in the values of variables of a case (the index examination in this study). ADAP then generates a forecast of the value or state of the phenomenon which may be a function of the input variables. Then, if it is in the learning mode, it reads in the actual state or value that it has just tried to forecast. It then compares the forecast with the true value and makes internal adjustments based on the direction and magnitude of the error. [FIGURE 1] In this respect ADAP is like other neural network learning devices.

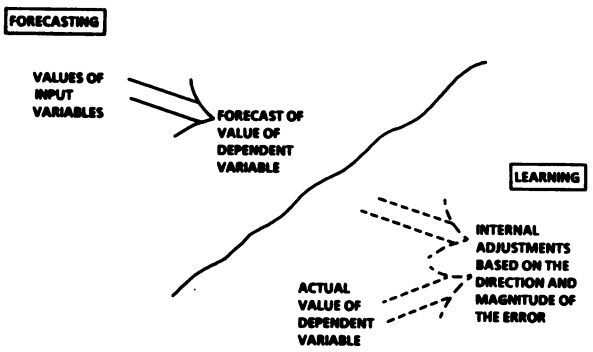

Figure 1

One major difference between ADAP and the more traditional neural networks is the way sensor units can be organized and "connected." Figure 2 is a schematic of an ADAP analog. Input cases excite sensors which are organized into partitions. In turn, those sensors are connected to association units that are connected through adjustable weights to a responder unit. A sensor unit represents a discrete value or range of values which may be assumed by a given input variable. The set of all sensors for a given input variable comprises the partition for that variable. While each association unit is connected to each sensor, it is usual in ADAP to restrict this pattern by requiring that there be one and only one nonzero connection within each partition. All nonzero connections are normally set to a weight of 1, and those weights are not varied during the learning process. This nonzero connection within a partition is selected randomly. ADAP typically uses a large number of such association units. (In this study it used 100,000) Each association unit has one and only one output connection to a responder. The rule of activation for an association unit given a particular set of input variable values is a threshold function. All of the values of a case that match a nonzero connection are said to *excite* those connections. If the total number of excited connections for a given association unit is greater than a predetermined value - the threshold - then that association unit is said to be *activated*. The threshold value is fixed such that somewhere between 2% and 10% of all association units will be activated for any input case. Finally, the responder values that are connected to each activated association unit are summed and this sum constitutes the forecast for the given input case. If ADAP is in the learning mode, the known value is compared with the forecast that has just been made and the difference is computed and allocated among the activated units for that training case. The entire forecasting/learning process may be repeated for additional cases. With ADAP, an output response can be interpreted as a forecast of either a continuous value (65.4, for example) or a state (*present or not present*).

Since the ADAP algorithm predates the resurgence of interest in connectionist models, some of its features differ from contemporary models. However, using the terminology of Rumelhart, Hinton and McClelland [3], it is most like an interactive associative learning model using the Hebbian learning rule.

3.1 Initialization of ADAP

To prepare for input into ADAP, one must divide or partition the values of the input variables into categories. For the diabetes forecasting problem, categories for each of the input variables were selected as shown in Table 1.

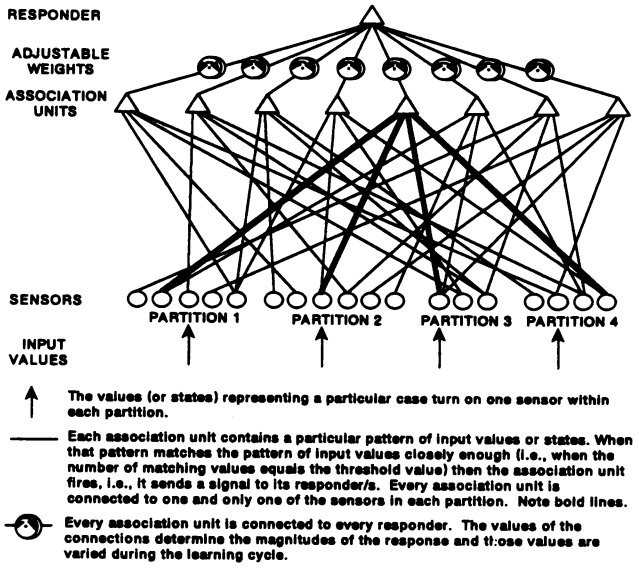

Figure 2

Next ADAP was instructed to generate an analog of a neural network specifically tailored for forecasting diabetes mellitus onset in five years given the above input variables and categories. The network consisted of a fixed matrix (the network of sensors and association units) and a variable array (the variable strength connections between association units and responders).

INPUT VARIABLES	# OF CATEGORIES	CATEGORIES
Number of Times Pregnant	3	{0,1,2} {3,4,5,6} {>7}
2 Hr Glucose Tolerance	6	{0-89.1} {89.2-107.1} {blank†, 107.2-123.1} {123.2-143.1} {143.2-165.1} {>165.2}
Diastolic BP	4	{blank} {1-76.1} {76.2-98.1} {>98.2}
Triceps Skin Fold	4	{blank} {1-25} {26-32} {>33}
2 Hr Serum Insulin	5	{blank} {1-110} {111-150} {151-240} {>241}
Body Mass Index	5	{1-22.814} {22.815-26.84} {blank, 26.841-33.55} {33.551-35.563} {>35.564}
Diabetes Pedigree Function	5	{0-.244} {.245-.525} {.526-805} {.806-1.11} {>1.11}
Age	5	{21-24} {25-30} {31-40} {41-55} {>55}

† blank = unknown

Table 1

3.2 The Fixed Matrix

The fixed matrix contained eight partitions corresponding to the eight input variables. Each partition was subdivided into rows, each of which corresponded to a range of values for the input variable. For instance, the first partition representing *number of times pregnant* contained three rows corresponding to the three categories of Table 1. Figure 3 is a diagram showing the partitions and categories used in this study. The columns are the analogs of association units. The Fixed Matrix contained 100,000 columns.

The intersection of a row and a column is the analog for the presence or lack of a connection between a sensor (partition value) and an association unit. An empty intersection indicates no connection. A plus sign (+) indicates a positive connection, commonly referred to as an exciter.

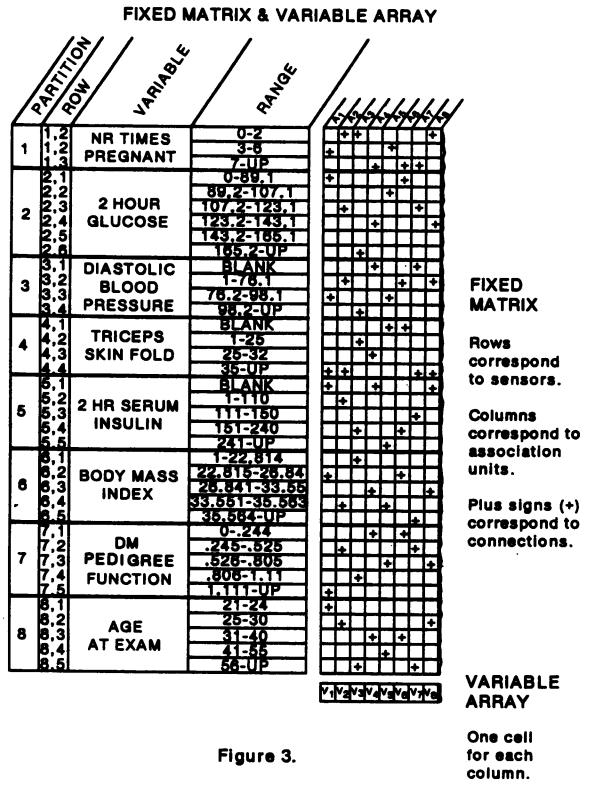

Figure 3.

3.3 The Variable Array

ADAP next generated a variable array which consisted of a single row containing one cell for each of the columns in the fixed matrix. This row is the analog of the connections between association units and one responder. It contains one cell for each column, and each cell contains a value corresponding to the strength of the connection between the corresponding column (association unit) and the responder (row). All values in this row were initialized (set equal) to zero. The word "initialized" is used here to emphasize that, although the fixed matrix remains unchanged during the learning process, the values in the variable array row do change as ADAP learns. Note the following:

- ADAP allows generation of fixed matrices with both positive connections (exciters) and negative connections (inhibitors). Only positive connections were used in this study.
- ADAP can generate variable arrays with many rows corresponding to neural networks with many responders. This allows more than one prediction to be made. In this study only one row was used since the effort was to forecast only one dependent variable.

In a perceptron, when some of the sensors are activated, they in turn activate some of the association units. Similarly, in this ADAP matrix when a set of rows is activated, certain columns will be activated. A case activates a set of rows. If a column contains enough positive connections in the activated rows (i.e., if the number of such connections is greater than or equal to a threshold value specified as an input parameter), then that column is activated and the corresponding value in the variable array is flagged. In the diabetes project, the threshold was four. If a column contained four or more + signs in rows that had been excited, then the column was activated. This threshold was chosen so that approximately 8% of the columns would be activated by an arbitrary case.

3.4 The Forecasting/Learning Process

A "*case*" triggers either a forecast cycle or a forecast cycle followed by a learning cycle. A case identifies a subset of rows corresponding to some occurrence.

Subject #1111 Exam Date 12/12/82			
INPUT VARIABLES			
Data Before Categorization		Categories as entered into ADAP	
VARIABLE	VALUE	PARTITION	ROW
Number of Times Pregnant	4	1	2
2 Hr Glucose	97	2	2
Diastolic BP	90	3	3
Triceps Skin Fold	31	4	3
2 Hr Serum Insulin	blank†	5	1
Body Mass Index	34.6	6	4
DM Pedigree Fn	0.46	7	2
Age	37	8	3

† blank = unknown

Table 2

For example, assume a case in our diabetes learning set was Subject 1111, Exam date 12/12/82 (Table 2). The values of the eight input variables, as shown, represent measurements taken at that exam or values computed from information that had been gathered and was in the Pima Indian data base as of 12/12/82. The input to ADAP is the partition numbers and row numbers corresponding to those measurements or calculations.

3.5 The Execution Cycle

1. ADAP reads in a case consisting of a value for each input variable.
2. Those rows that match to the input variable values (categories) for the case at hand are activated.
3. ADAP counts the number of + signs in the first column that are in an activated rows. If that count is greater than the threshold (in this case the threshold was four), the cell in the variable array corresponding to the column is flagged. Otherwise it is not flagged.
4. The next column is then compared and the algorithm iterates until all columns have been processed. Approximately 8000 columns were activated by each input case.
5. The values of the flagged cells in the variable array are summed. That sum is the FORECAST for the case.
6. If ADAP is not in the learning mode the flags in all cells are erased and a new case is read in.
7. If ADAP is in the learning mode the actual state or value that ADAP had just tried to forecast is read in. For this study, a value of 0 was used for the dependent variable if the subject had a diabetes-free examination more than five years from the index examination and a value of 1 was used if the subject developed diabetes within one to five years of the index examination.
8. ADAP now makes appropriate internal corrections. It generates an error estimate as

$$\overline{error} = \frac{Acutal\ value - Forecast\ value}{Number\ of\ flagged\ cells}$$

   This value is added to each of the flagged cells.
9. All flags are now removed and a new case is read in.

4. RESULTS AND CONCLUSIONS

After learning using all 576 training cases, the ADAP algorithm was put into a nonlearning mode. That is, the values in the forecasting row could be used for prediction but those values would not be updated following each case. Since the value returned by ADAP is a real number and since the prediction being made is discrete, the value chosen to discriminate whether or not diabetes will occur within five years will influence the sensitivity and specificity of the forecast. (See Figure 4.) When a value of 0.448 was used (those with lower values were forecast to remain diabetes free while those with values greater than or equal were forecast to develop diabetes within five years), the sensitivity and specificity were both equal at 76%. A ROC curve [10], which plots the sensitivity (true positive rate) against the false positive rate (1 - specificity), is shown to display the influence of altering the value of the discrimination point. (See Figure 5).

We are currently testing the same data set divided identically into the same training and forecasting sets using both logistic regression and a linear perceptron model. We plan to have data to compare the ROC curves for this prediction using all three methods - ADAP, logistic regression and a linear perceptron.

REFERENCES

[1] Simmons, G.J., A Mathematical Model for an Associative Memory. Sandia Corporation Monograph, SCR-641, April 1963.

[2] Rosenblatt, F., The Perceptron, A Theory of Statistical Separability in Cognitive Systems. Cornell Aeronautical Laboratory Report Nr. VG-1196-G1, January 1958.

[3] Rumelhart DE, Hinton GE, McClelland JL, A General Framework for Parallel Distributed Processing in Parallel Distributed Processing, Volume 1, pp 45-75, MIT Press (Cambridge), 1986. MIT Press, 1986.

[4] Smith, J.W., ADAP, An Adaptive Algorithm, Logistics Management Institute, Technical Report 1R 705, November 1986.

[5] Smith, J.W. ADAP II, An Adaptive Routine for the LARC Computer, Navy Management Office, Sept 1962. (Available through the Logistics Management Institute Library)

[6] Bennett, P.H., T.A. Burch, and M. Miller. 1971. Diabetes mellitus in American (Pima) Indians. Lancet 2:125-128.

[7] Knowler, W.C., P.H. Bennett, R.F. Hamman, and M. Milier. 1978. Diabetes incidence and prevalence in Pima Indians: a 19-fold greater incidence than in Rochester, Minnesota. Am J Epidemiol 108:497-505.

[8] Knowler, W.C., D.J. Pettitt, P.J. Savage, and P.H. Bennett. 1981. Diabetes incidence in Pima Indians: contributions of obesity and parental diabetes. Am J Epidemiol 113:144-156.

[9] Feigenbaum E, Barr A, Learning in Control and Pattern Recognition Systems in The Handbook of Artificial Intelligence, Volume III William Kaufmann Inc, pp 373-382 (Los Altos), 1982.

[10] World Health Organization, Report of a Study Group: Diabetes Mellitus. World Health Organization Technical Report Series. Geneva, 727, 1985.

[11] McNeil BJ, Hanley JA: Statistical approaches to the analysis of receiver operating characteristic (ROC) curves. Medical Decision Making. 1984. 4(2). pp. 137-50.

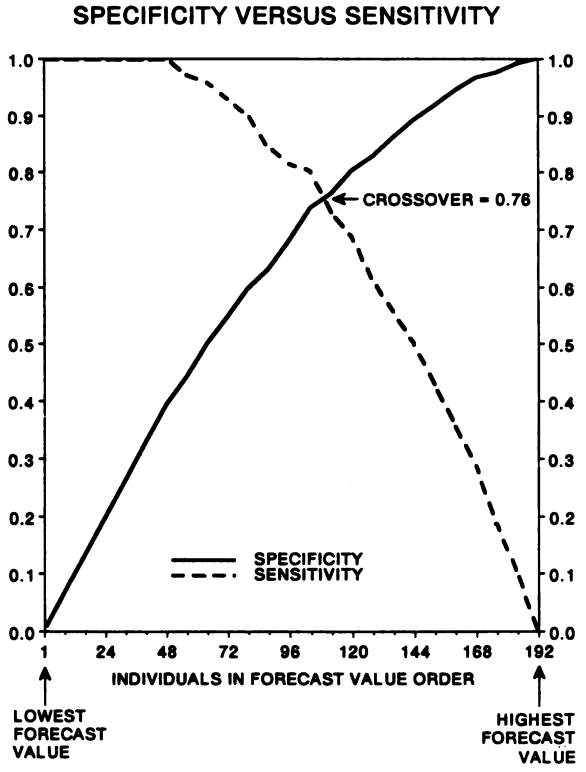

Figure 4

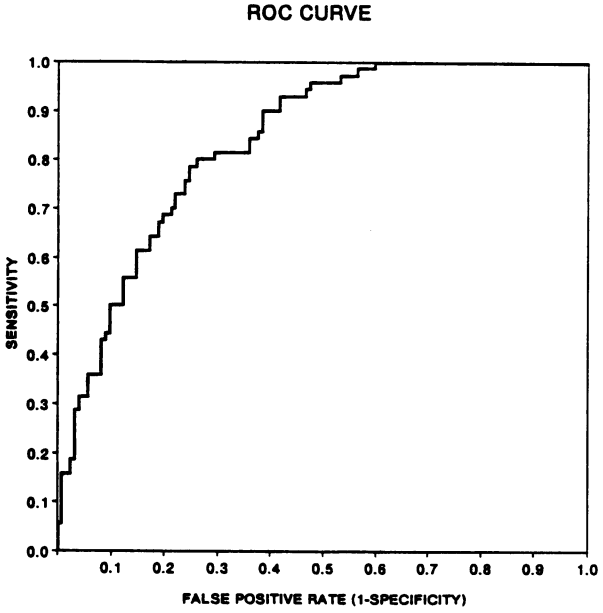

Figure 5