A MONTE CARLO METHOD FOR IMPLEMENTING MODEL-BASED DIAGNOSTIC PROGRAMS

by

Bryan S. Todd

Technical Monograph PRG-83 May 1990

Oxford University Computing Laboratory
Programming Research Group
8–11 Keble Road
Oxford OX1 3QD
England

Oxford University Computing Laboratory
Programming Research Group
8–11 Keble Road
Oxford OX1 3QD
England

Electronic mail: todd@uk.ac.oxford.prg (JANET)

A Monte Carlo Method for Implementing Model-Based Diagnostic Programs

Bryan S. Todd

Summary

The statistical analysis of collections of previous case records has proved a useful way of giving diagnostic assistance to the clinician. In certain applications, 'simulation models' of disease processes provide a way of supplementing the available numerical data with the causal relationships that are known to exist. However, the diagnosis of new patients by reference to such simulation models tends to be computationally hard. In these circumstances a possible solution is to use the model to generate randomly a database of hypothetical cases which is sufficiently large to enable a more effective form of statistical classification than was previously possible. In this paper, several classifiers are considered for this purpose. A method is described for comparing the diagnostic accuracy of the classifiers in a way which is independent of the medical correctness of the simulation model itself. The method is illustrated by an example.

Contents

1 Introduction 1

2 A Monte Carlo Method 2

3 Classification 2
3.1 Bayes' Theorem 3
3.2 A Non-Parametric Method 4
3.3 Linear Discriminant 5
3.4 Two-layer Perceptron 5

4 Comparing Diagnostic Accuracy 7
4.1 An Empirical Method 7
4.2 An Example 7
4.3 Methods 8
4.4 Results 9
4.5 Discussion 12

5 General Conclusions 13

Acknowledgements 13

References 14

1 Introduction

If the purpose of a medical diagnostic program is to assist the clinician in diagnosing patients with greater accuracy, then a particularly successful type of program is one which compares the patient to a database of previous cases. It is not even necessary to resort to complicated statistical methods. Using a database of fewer than 1000 cases to estimate the required probabilities, and applying Bayes' theorem with the assumption of conditional independence, de Dombal *et al* [Dom72, Dom74] showed that effective support can be provided in the diagnosis of acute abdominal pain. A comparative study [Ser86] by others has since suggested that diagnostic performance can be improved slightly by taking into account pairwise interactions, but a much larger database is needed in order to estimate the many conditional probabilities involved. It is not always practicable to collect vast amounts of objective data for this purpose, particularly if the relevant disorders are rare.

When statistical dependencies exist, they are often predictable from an understanding of the way diseases progress, and from an appreciation of the relevant anatomy and physiology. At least in some applications, this knowledge can complement the available objective data in the construction of a simulation model. This means a computer model which simulates probabilistically the occurrence of one or more disorders and their consequences, and whose structure reflects the relevant first principles.

A representation that has been studied recently is the 'Bayesian network' [Pearl86, Pearl87a, Laur88]. Here a directed graph is used to group together the dependent findings explicitly. Each node represents a random variable describing some medical condition (disease, symptom, sign *etc.*). Each arc represents direct dependence of one variable on the state of another. Associated with each node is a table which specifies the conditional probability of that node taking any of its possible values given every combination of states of its parents. A Bayesian network amounts to a simulation model of the disease process because, in conjunction with a random number generator, the network can be used to generate descriptions of random hypothetical cases [Hen88]. Depending on the particular application, other representations may be preferred: for example, probabilistic causal graphs [Lud83, Peng87], or even direct mathematical specification [Roh75, Hains88].

Unfortunately, the task of drawing diagnostic conclusions does tend to be computationally hard [Coop89]. Nevertheless, it may be possible to deal with special cases. For example, algorithms for Bayesian networks have been developed which are efficient provided that the graphs are suitably

sparse [Laur88]. In other cases it is often possible to devise some specific means of approximating the required solution [Pearl87b, Todd88, Kault89]. But, when no such algorithm can be found, a more generally applicable (if less powerful) method is appropriate. The following is one such technique.

2 A Monte Carlo Method

If a simulation model generates a random sample of hypothetical cases, and if the model is a full and accurate description of the mechanism of disease in the intended population, then the sample of hypothetical cases cannot be distinguished from a sample of actnal cases drawn randomly from the same population. Therefore the sample can be used in the same way as an actual training sample to parameterize a statistical classifier. However, since the sample is generated rapidly by a computer rather than collected by hand it can be very much larger, sufficiently large to parameterize a more complex classifier than was previously possible.

When using very large training databases, the computational complexity of the chosen classification method is especially important. It is clearly an advantage if the classifier can be trained just once, and then used to classify new cases without the need for further reference to the database. This means that the generated database can be much larger: significantly more computing effort can be expended during an initial training phase than would be reasonable for the diagnosis of just one new case. In either event, methods which involve a procedure whose complexity is quadratic or worse with respect to the size of the database would appear to be unsuitable for this application.

In this paper several classification methods are considered which are all computationally feasible for this task, and which represent a diversity of approaches from Bayes' theorem to neural networks. However, the author would not wish to give the impression that this is an exhaustive collection of possible classifiers; additional suggestions would be most welcome.

3 Classification

For simplicity, we regard diseases and symptoms bere as being merely present or absent, and unassociated with any other parameter. A case is thus described by a pair (D, S) of binary vectors. Component D_d takes value 1 or 0 according to whether disease d is present or absent, respectively. Likewise, S, takes value 1 or 0 according to whether symptom s is present or absent. We use Greek letters (δ, σ) to denote a patient drawn randomly from

the population. The classification task given such a patient's symptoms σ is to determine δ as reliably as possible.

When reconstructing the vector δ, fewest errors are expected if, for each d independently, δ_d is assumed to have the most probable value 1 or 0 given σ. Let us now consider some alternative ways of estimating each conditional probability $p(\delta_d = 1 \mid \sigma = S)$.

3.1 Bayes' Theorem

Let $r_d(\sigma = S)$ be the ratio of the probability that disease d is present, to the probability that it is absent.

$$r_d(\sigma = S) = \frac{p(\delta_d = 1 \mid \sigma = S)}{p(\delta_d = 0 \mid \sigma = S)} \tag{1}$$

$$= \frac{p(\delta_d = 1)}{p(\delta_d = 0)} \times \frac{p(\sigma = S \mid \delta_d = 1)}{p(\sigma = S \mid \delta_d = 0)} \tag{2}$$

The first of the two terms in the product above (Equation 2) can be estimated directly from a reasonably-sized random sample of the population. In the present context, the random sample is the training database generated using the simulation model. However, it is not usually feasible to estimate the second term directly because no examples of cases with symptoms identical to those of the patient in question can be found in a sample of practicable size. We can proceed only by making some assumptions about the underlying distribution. The simplest and strongest assumption that we might wish to make is that symptoms occur independently of one another both in the presence and absence of any disease. This enables the following.

$$\frac{p(\sigma = S \mid \delta_d = 1)}{p(\sigma = S \mid \delta_d = 0)} = \prod_s \frac{p(\sigma_s = S_s \mid \delta_d = 1)}{p(\sigma_s = S_s \mid \delta_d = 0)} \tag{3}$$

A weaker alternative permits pairwise dependencies but still no higher order interactions [Ser86, Zent75]. The right-band expression above (Equation 3) is then modified by a correction factor $q(1)/q(0)$, where the function q is defined as follows.

$$q(b) = 1 + \sum_{t,u \mid t<u} \left(\frac{p(\sigma_t = S_t \wedge \sigma_u = S_u \mid \delta_d = b)}{p(\sigma_t = S_t \mid \delta_d = b) p(\sigma_u = S_u \mid \delta_d = b)} - 1 \right) \tag{4}$$

We shall refer to the two methods as the Lancaster model ($s = 1$) and the Lancaster model ($s = 2$) respectively. Higher order models also exist but

they require the estimation of more conditional probabilities than is feasible for our present study.

Whichever model is used to compute the ratio $r_d(\sigma = S)$, the conditional probability we require is easily recovered.

$$p(\delta_d = 1 \mid \sigma = S) = \frac{r_d(\sigma = S)}{1 + r_d(\sigma = S)} \tag{5}$$

Only a single pass through the training database is required to estimate the necessary parameters. No further reference to the database is then necessary for the classification of any new case. We can express this as follows, where T stands for the size of the training database, N stands for the number of new cases to classify, and O denotes 'order of computational complexity'.

$$\text{Lancaster\_Model}(T, N) = O(T + N) \tag{6}$$

If a very large database is required in order to obtain reasonable estimates of the conditional probabilities, then the task can be distributed between an unlimited number of processors, each counting the frequencies of every event in a different part of the database. No inter-process communication is necessary until the entire database has been examined, following which, the separate counts are summed.

3.2 A Non-Parametric Method

An alternative method which avoids making specific assumptions about the distribution of diseases and symptoms, is to extract from the database the first k cases we find which most closely resemble (least Hamming distance between the symptom vectors) the case we are attempting to classify [Croft74]. We estimate conditional probabilities of each disease d being present by

$$p(\delta_d \mid \sigma = S) = h_d/k \tag{7}$$

where h_d is the number of closest neighbours who have d. Unfortunately though, the entire database must be searched for each new case we diagnose. There is no distinct training phase as for the previous method.

$$\text{Nearest\_Neighbours}(T, N) = O(T \times N) \tag{8}$$

The search task is readily distributed among different processors by assigning each the task of extracting the closest k neighbours from a different part of the database. The results are then merged to form a smaller database before repeating the process [Stan86]. If the number of processors is unrestricted then a logarithmic reduction in the search time is possible.

3.3 Linear Discriminant

An alternative is to assume that $p(\delta_d = 1 \mid \sigma = S)$ depends linearly on the components of S.

$$p(\delta_d = 1 \mid \sigma = S) = a + \sum_s b_s S_s \tag{9}$$

The coefficients a and b are determined by fitting a least-squares regression line to the diseases and symptoms recorded in the database. This in turn entails the solution of a system of linear equations derived by counting the frequencies of occurrence of all symptom-symptom pairs, and all symptom-disease pairs. We therefore have distinct training and diagnosis phases when using this method.

$$\text{Linear\_Discriminant}(T, N) = O(T + N) \tag{10}$$

If the database is very large, then the counting task can be distributed between different processors, each counting the frequencies of all the relevant events in a separate part of the database.

3.4 Two-layer Perceptron

A more flexible model is provided by the two-layer perceptron [Lipp87] (Figure 1). More complex configurations are possible, but here we consider only two-layer devices with single outputs: each disease is recognized independently by a separate perceptron.

Each unit computes the weighted sum of its inputs, and then applies the sigmoid function ϕ.

$$\phi(x) = 1/(1 + e^{-x}) \tag{11}$$

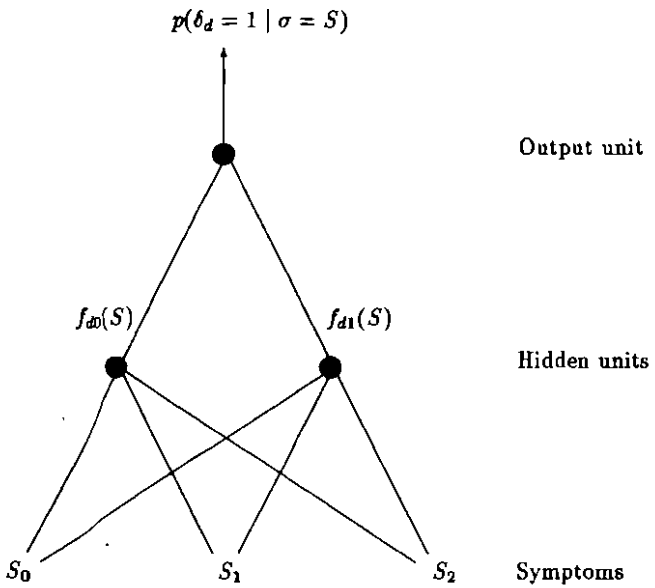
Thus each unit h in the hidden layer of the perceptron discriminating for or against disease d, implements the function f_{dh} (Equation 12) where A and B are respectively a vector and a matrix of real-valued coefficients.

$$f_{dh}(S) = \phi(A_h + \sum_s B_{hs} S_s) \tag{12}$$

The values computed by the first layer are presented as inputs to the single output unit. The entire perceptron therefore estimates the required conditional probability according to

$$p(\delta_d = 1 \mid \sigma = S) = \phi(a + \sum_h b_h f_{dh}(S)) \tag{13}$$

Figure 1: Two-layer perceptron with single output recognizing disease d. (For clarity only two hidden units and three symptoms are shown.)



The 'back-propagation' algorithm [Rum86] provides an effective iterative method for optimizing the various coefficients so as to minimize the square error. On the assumption that each iteration of this procedure is as expensive computationally as generating a new random case from the simulation model, there is little to be gained by storing generated cases. Training thus entails a single pass through the generated database, which is as large as computing resources permit.

$$\text{Perceptron}(T, N) = O(T + N) \tag{14}$$

The training of each disease's perceptron can proceed in parallel entirely independently of that of any other. However, further useful distribution of the training task appears progressively more difficult.

4 Comparing Diagnostic Accuracy

4.1 An Empirical Method

Although a variety of classification methods are available, some with a more favourable computational complexity than others, our final choice must ultimately depend on the nature of the underlying joint distribution of the training data, and hence of the simulation model itself. Fortunately, it is a simple matter to compare the different methods empirically. Having generated a large training database, the same model is used to generate a further small set of test cases. Each of the classification methods in turn is used to reconstruct the disease vectors from the just the symptom vectors of the test cases. Classification methods are ranked according to how well they succeed in reconstructing the disease vectors correctly.

The circularity in testing the classifiers on further generated data rather than on real patient data is deliberate. This ranks the classifiers according to how well they invert the descriptive information in the model. If the model used for training contains errors, and real patient data are used for testing, then a paradoxical effect may be observed: a complex, more highly parameterized classifier, which adapts more closely to the generated data, is worse when applied to real data. Clearly, if we wish to identify and correct errors in a model, rather than simply obscure them, then we should choose the method which most accurately inverts the model.

4.2 An Example

We have applied this method to a simulation model being developed for the diagnosis of acute abdominal pain. The model [Todd88] consists of a probabilistic causal graph. This differs from a Bayesian network in that instead of associating a table of probabilities with each node, it associates a single conditional probability with each arc: the probability that, if the source node is present, it *causes* the target node to occur also. Peng and Reggia refer to this as the 'conditional causal probability' [Peng87]. Furthermore, in order to avoid impossible combinations of events occurring, a 'prevents' relation is defined on the nodes to indicate that when certain nodes are present the attempt to cause certain others necessarily fails. This was also suggested in the paper by Peng and Reggia [Peng87]. Directed cycles are permitted provided that none include any 'prevents' arcs [Todd88].

The graph was compiled from textbook accounts of many of the common conditions. It contains 297 nodes, 775 causative arcs and 30 preventative arcs. Although the model has not been refined, its size and structure appear typical. One way to identify deficiencies of such a model is to attempt to

use it to diagnose real cases whose actual diagnosis is already known. But for this, an inference algorithm is necessary.

Since the longest directed cycle in the graph involved only six nodes, it was possible, by liberal introduction of auxiliary nodes, to convert the model to a Bayesian network in which no node had more than six parents. The methods described by Lauritzen and Spiegelhalter [Laur88] were tried, but both 'maximum cardinality' and 'lexicographic' searches led to filled-in graphs with unmanageably large cliques (more than 50 nodes). The Monte Carlo method described by Pearl [Pearl87b] was also tried, but no useful convergence was obtained even after 10^4 iterations for each case. Therefore, the technique presented in this paper was tried, using the original probabilistic causal graph, from which random cases could he generated quickly.

4.3 Methods

All programs were written in Pascal, and run on a Sun 3/50 Workstation under a Unix (Registered Trademark) version 4.2bsd operating system. An algorithm described elsewhere [Todd88] was used to generate random cases from the model. A multiplicative congruential generator [Fish86] with multiplier 742938285 and modulus 2147483647 provided a source of pseudo-random numbers. Taking the seed as the case's index number a virtual database was implemented: this avoided storing large amounts of data. The training set and the test set consisted of 10^6 and 100 cases, respectively (larger test sets were not feasible for the 'nearest neighbours' method). For test purposes, 12 nodes representing the principal disorders were labelled 'disease' and 150 nodes representing clinical findings were labelled 'symptom'. The disease vector of each case D therefore had 12 components, and the symptom vector S had 150 components.

Two of the classifiers required estimation of application-specific parameters. Before applying the 'nearest neighbours' technique it had first to be decided how many of the closest neighbours to use for estimation purposes. Similarly, it was necessary to choose how many hidden units to employ and which gain factor to use before implementing and training an array of perceptrons. Initial experiments were carried out using training sets of 10^5 cases in order to determine the most effective values ($k = 10$, 5 hidden units and a gain factor of 0.3, respectively).

The initial weights in the perceptrons were drawn randomly from the uniform distribution between -0.5 and 0.5. Only the simplest version of the back-propagation algorithm [Rum86] was used to optimize the weights, back-propagating errors after each case, and using only first derivatives of the error gradient.

Although the relative performance of the various classifiers can be determined empirically, since no exact method appears applicable, the absolute performance of the classifiers in this application is unknown. To remedy this partially, the author attempted his own diagnosis of the 100 test cases, making free reference to the probabilistic causal graph that had been used to generate them.

4.4 Results

The results obtained with the five classifiers are summarized in Table 1, together with a similar assessment of the author's own performance. For each classifier the total number of true and false, positive and negative diagnoses are shown, with respect to all 12 diseases in all 100 cases. The scores therefore represent the accumulated outcomes in a total of 1200 diagnostic decisions. A total of 184 diseases were actually present, representing an average of 1.84 diseases per case.

To offset possible bias introduced through incorrect assumptions about the structure of the underlying joint distribution, the performance of each classifier was assessed with various decision thresholds. Classifiers are compared according to their optimum performance; the scores correspond to different decision thresholds for different classification methods, the optimal threshold being chosen as one which minimizes the total number of errors (false positives + false negatives).

Table 1: Error rates for all methods. Training database: Cases 0 to 999999 inclusive. Test set: Cases 1000100 to 1000199 inclusive. 'OT' = 'Optimal Threshold', 'TP' = 'True Positives', 'FP' = 'False Positives', 'TN' = 'True Negatives', 'FN' = 'False Negatives'.

METHOD	OT	TP	FP	TN	FN
Lancaster ($s = 1$)	0.99	130	446	570	54
Lancaster ($s = 2$)	0.99	131	485	531	53
Nearest Neighbours ($k = 10$)	0.51	135	10	1006	49
Linear Discriminant	0.49	151	12	1004	33
Perceptron (5 hidden units)	0.57	157	10	1006	27
Author + Graph		145	14	1002	39

Repeating the comparison with a larger test set of 1000 cases, the perceptrons were found to produce fewer classification errors than the linear discriminants at the 1% significance level (Wilcoxon paired rank test).

Tables 2 and 3 show the numerical outputs obtained with the linear discriminants and the perceptrons respectively, for the first ten cases in the test set. Values outside the interval [0,1] computed by the first method show the assumption of linearity is invalid (Equation 9), although the deviations are small.

Table 2: Discriminant values obtained from linear discriminants.

CASE	DISEASE											
	1	2	3	4	5	6	7	8	9	10	11	12
1000100	0.03	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.01	0.00	0.00
1000101	0.03	0.01	0.02	-0.02	0.04	0.19	0.13	0.06	0.58	0.01	-0.03	0.01
1000102	0.03	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.01	0.00	0.00
1000103	0.18	0.00	0.08	0.10	0.15	-0.04	0.08	0.84	-0.13	0.01	0.77	0.47
1000104	1.10	0.10	0.00	0.02	0.06	0.12	0.19	0.09	0.99	0.01	-0.01	-0.01
1000105	0.71	-0.09	0.03	0.00	-0.01	0.36	0.08	-0.01	-0.02	1.14	0.06	0.01
1000106	0.99	0.18	0.02	0.02	0.00	0.00	0.74	0.11	-0.02	0.01	-0.05	-0.01
1000107	0.88	0.05	0.01	0.21	0.15	0.29	0.03	1.02	0.10	0.00	0.88	0.43
1000108	1.01	1.28	0.17	0.00	0.22	0.18	-0.02	1.05	0.03	0.00	0.03	-0.13
1000109	1.09	-0.04	0.00	0.01	0.00	0.05	0.01	0.00	0.00	0.01	0.00	0.00

Table 3: Discriminant values obtained from perceptrons.

CASE	DISEASE											
	1	2	3	4	5	6	7	8	9	10	11	12
1000100	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
1000101	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	1.00	0.00	0.00	0.00
1000102	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
1000103	0.00	0.00	0.01	0.02	0.07	0.09	0.08	1.00	0.00	0.00	0.90	0.26
1000104	1.00	0.00	0.00	0.00	0.01	0.07	0.00	0.00	1.00	0.01	0.00	0.00
1000105	1.00	0.00	0.01	0.00	0.00	0.31	0.00	0.00	0.00	1.00	0.00	0.00
1000106	1.00	0.00	0.00	0.00	0.00	0.01	1.00	0.00	0.00	0.01	0.00	0.00
1000107	0.98	0.01	0.01	0.04	0.08	0.10	0.08	1.00	0.01	0.00	0.96	0.07
1000108	1.00	1.00	0.03	0.01	0.08	0.08	0.07	1.00	0.03	0.00	0.00	0.00
1000109	1.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00

Table 4 shows the author's diagnoses for the first ten cases in the test set. In two cases there was considerable difficulty in distinguishing between

disease 11 and disease 12. Table 5 lists the actual diagnoses for the same ten cases. Two of the cases (number 1000100 and number 1000102 in the virtual database) had none of the diseases: such cases are included because it is also important to be able to detect normality as well as disorder.

Table 4: Author's diagnoses. '+' = 'present', '-' = 'absent', '?' = 'uncertain'.

CASE	DISEASE											
	1	2	3	4	5	6	7	8	9	10	11	12
1000100	-	-	-	-	-	-	-	-	-	-	-	-
1000101	-	-	-	-	-	-	-	-	+	-	-	-
1000102	-	-	-	-	-	-	-	-	-	-	-	-
1000103	-	-	-	-	-	-	-	+	-	-	+?	-?
1000104	+	-	-	-	-	-	-	-	+	-	-	-
1000105	+	-	-	-	-	-	-	-	-	+	-	-
1000106	+	-	-	-	-	-	+	-	-	-	-	-
1000107	+	-	-	-	-	-	-	+	-	-	+?	-?
1000108	+	+	-	-	-	-	-	+	-	-	-	-
1000109	+	-	-	-	-	-	-	-	-	-	-	-

Table 5: Actual diagnoses.

CASE	DISEASE											
	1	2	3	4	5	6	7	8	9	10	11	12
1000100	-	-	-	-	-	-	-	-	-	-	-	-
1000101	-	-	-	-	-	-	-	-	+	-	-	-
1000102	-	-	-	-	-	-	-	-	-	-	-	-
1000103	-	-	-	-	-	-	-	+	-	-	+	-
1000104	+	-	-	-	-	-	-	-	+	-	-	-
1000105	+	-	-	-	-	-	-	-	-	+	-	-
1000106	+	-	-	-	-	-	+	-	-	-	-	-
1000107	+	-	-	-	+	-	-	+	-	-	-	+
1000108	+	+	-	-	-	-	-	+	-	-	-	-
1000109	+	-	-	-	-	-	-	-	-	-	-	-

4.5 Discussion

In this study, Bayesian methods have been the least successful. Clearly the invalid assumptions of conditional independence have made estimation of the joint probabilities very unreliable. The optimum threshold for the first two models is at the extreme upper end of the range, and reflects their poor suitability and calibration.

The k-nearest neighbours technique fared much better, and could no doubt be considerably improved by the use of weights of importance. Nevertheless, it is handicapped by high computational complexity. In other comparative studies, the k-nearest neighbours method has been found to be inferior to the independence Bayes model [Croft74, Ser85]. This would appear to be due to the use of much smaller training samples (1991 and 5916 cases respectively) than in the present study.

The linear discriminant method has proved better than unweighted k-nearest neighbours for the size of training database employed here. However, with increasing size of training database, the k-nearest neighbours method will approach ever closer the optimum possible performance while the linear discriminant method will not continue to improve significantly. But, it is unclear whether or not the cross-over point occurs with a database of feasible size.

The neural network appears the best of those studied. The probabilities (Table 3) computed by the 2-layer perceptrons correlate well with the actual diagnoses (Table 5), and do so more precisely than the linear discriminant values (Table 2). Where the author (Table 4) felt uncertain, the probability computed by the perceptron also deviated from 0 or 1. Reviewing the other cases in which the perceptron's output was significantly different from 0 or 1, the author felt that in retrospect he had been too confident himself.

The training technique employed in this study differs from the conventional in that each training case is presented once only. There is an obvious danger when training a classifier such as a neural network with many degrees of freedom, of overfitting to past data leading to poor prediction. Presenting cases only once obviates this danger: the expected diagnostic performance can only improve if a larger training database is used.

In this particular instance it was not possible, but in future studies, models will be used that are sufficiently simple that available exact algorithms are applicable. In this way it will be possible to assess more accurately the extent to which the model's information is transferred to the neural network by massive simulation.

5 General Conclusions

When specific algorithms cannot be found for drawing diagnostic inferences from a simulation model, massive simulation combined with the training of a statistical classifier appears to be a useful computational technique. Alternative classifiers can be compared in the way described in this paper (using the simulation model to generate random cases both for training and testing), and the best selected. For models with a size and structure similar to the causal graph employed here, our results suggest that a form of neural network is likely to be effective, and we recommend that neural networks are considered.

Acknowledgements

I am most grateful to Tony Hoare for his advice and comments on this work, and to the Oxford University Programming Research Group for permission to use their computing facilities.

References

[Coop89] Cooper, G. F., *Current Research Directions in the Development of Expert Systems Based on Belief Networks, Applied Stochastic Models and Data Analysis*, 1989 **5** 39-52.

[Croft74] Croft, D. J. and Machol, R. E., *Mathematical Methods in Medical Diagnosis, Annals of Biomedical Engineering*, 1974 **2** 69-89.

[Dom72] De Dombal, F. T., Leaper, D. J., Staniland, J. R., McCann A. P and Horrocks J. C., *Computer-Aided Diagnosis of Acute Abdominal Pain, British Medical Journal*, 1972 **2** 9-13.

[Dom74] De Dombal, F. T., Leaper, D. J., Horrocks, J. C., Staniland, J. R. and McCann, A. P., *Human and Computer-Aided Diagnosis of Abdominal Pain: Further Report with Emphasis on Performance of Clinicians, British Medical Journal*, 1974 **2** 376-80.

[Fish86] Fishman, G. S. and Moore III, L. R., *An Exhaustive Analysis of Multiplicative Congruential Random Number Generators with Modulus* $2^{31}-1$, *SIAM Journal of Scientific and Statistical Computing*, 1986 **7** 24-45.

[Hains88] Hains, G. and Todd, B. S., *A Parallel Implementation of a Medical Diagnostic Program*, Proc. 3rd Int. Conference, *Supercomputing*, 1988 1 222-9.

[Hen88] Henrion, M., *Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling, Uncertainty in Artificial Intelligence 2* ed J F Lemmer and L N Kanal (Amsterdam: North-Holland), 1988.

[Kault89] Kault, D., Stark, D. and Stark, K., *An Automated System of Strabismus Management, Investigative Ophthalmology*, 1989 **30** 276-87.

[Laur88] Lauritzen, S. L. and Spiegelhalter, D. J., *Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems, Journal of the Royal Statistical Society B*, 1988 **50** 157-224.

[Lipp87] Lippmann, R. P., *An Introduction to Computing with Neural Nets, IEEE Acoustics, Speech & Signal Processing Magazine*, 1987 **4** 4-22.

[Lud83] Ludwig, D. and Heilbronn, D., *The Design and Testing of a New Approach to Computer-Aided Differential Diagnosis, Methods of Information in Medicine*, 1983 **22** 156-66.

[Pearl86] Pearl, J., *Fusion, Propagation, and Structuring in Belief Networks, Artificial Intelligence*, 1986 **29** 241-88.

[Pearl87a] Pearl, J., *Distributed Revision of Composite Beliefs, Artificial Intelligence*, 1987 **33** 173-215.

[Pearl87b] Pearl, J., *Evidential Reasoning using Stochastic Simulation of Causal Models, Artificial Intelligence*, 1987 **32** 245-57.

[Peng87] Peng, Y. and Reggia, J. A., *A Probabilistic Causal Model for Diagnostic Problem Solving - Part 1: Integrating Symbolic Causal Inference with Numeric Probabilistic Inference, IEEE Transactions on Systems, Man, and Cybernetics*, 1987 **SMC-17** 146-62.

[Rob75] Robinson, D. A., *A Quantitative Analysis of Extraocular Muscle Cooperation and Squint, Investigative Ophthalmology*, 1975 **14** 801-25.

[Rum86] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., *Learning Representations by Back-Propagating Errors, Nature*, 1986 **323** 533-6.

[Ser85] Séroussi, B., *Comparison of Several Discrimination Methods: Application to the Acute Abdominal Pain Diagnosis, Lecture Notes in Medical Informatics*, 1985 **28** 12-8.

[Ser86] Séroussi, B. and the ARC and AURC Cooperative Group, *Computer-Aided Diagnosis of Acute Abdominal Pain when Taking into Account Interactions, Methods of Information in Medicine*, 1986 **25** 194-8.

[Stan86] Stanfill, C. and Waltz, D., *Toward Memory-Based Reasoning, Communications of the ACM*, 1986 **29** 1213-28.

[Todd88] Todd, B. S., *A formal approach to the design of medical diagnostic programs*, D.Phil. thesis, Oxford, 1988.

[Zent75] Zentgraf, R., *A Note on Lancaster's Definition of Higher-Order Interactions, Biometrika*, 1975 **62** 375-8.