Scandinavian Journal of Clinical and Laboratory Investigation

ISSN: 0036-5513 (Print) 1502-7686 (Online) Journal homepage: https://www.tandfonline.com/loi/iclb20

Using data preprocessing and single layer perceptron to analyze laboratory data

J. J. Forsström, K. Irjala, G. Selén, M. Nyström & P. Eiuund

To cite this article: J. J. Forsström, K. Irjala, G. Selén, M. Nyström & P. Eiuund (1995) Using data preprocessing and single layer perceptron to analyze laboratory data, Scandinavian Journal of Clinical and Laboratory Investigation, 55:sup222, 75-81, DOI: 10.3109/00365519509088453

To link to this article: https://doi.org/10.3109/00365519509088453

Published online: 08 Jul 2009.

Using data preprocessing and single layer perceptron to analyze laboratory data

J. J. FORSSTRÖM[1], K. IRJALA[2], G. SELÉN[3], M. NYSTRÖM[3] & P. EKLUND[3]
[1]Department of Medicine, University of Turku, [2]Central Laboratory, Turku University Central Hospital, [3]Department of Computer Science, Åbo Akademi University, Turku, Finland

Forsström JJ, Irjala K, Selén G, Nyström M, Eklund P. Using data preprocessing and single layer perceptron to analyze laboratory data. Scand J Clin Lab Invest 1995; 55, Suppl. 222: 75-81.

During daily work in hospitals a large amount of clinical data is produced each day. Totally computerized patient records are not yet widely used but a large part of essential information is already stored on computer files. These include laboratory test results, diagnoses, codes for operations, codes of histopathological diagnoses and maybe even the patient's medication. Accordingly, these databases include much clinical knowledge that would be useful for clinicians.

Laboratories try to support clinicians by producing reference values for laboratory tests. It is, of course, necessary information but, however, it does not give very much information about the weight of evidence that an abnormal laboratory test will give in special clinical settings.

We have developed a software package - DiagaiD - in order to build a smart link between patient databases and clinicians. It utilizes neural network-based machine learning techniques and can produce decision support which meets the special needs of clinicians. From example cases it can learn clinically relevant transformations from original numeric values to logical values. By using data transformation together with a single layer perceptron it is possible to build non-linear models from a set of preclassified example cases.

In this paper, we use two small datasets to show how this scheme works in the diagnosis of acute appendicitis and in the diagnosis of myocardial infarction. Results are compared with those obtained using logistic regression or backpropagation neural networks. The performance of our neuro-fuzzy tool seemed to be slightly better in these two materials but the differences did not reach statistical significance.

Key words: acute appendicitis, backpropagation, diagnosis, medical decision support, neuro-fuzzy system

Jari Forsström, Department of Medicine, University of Turku, Kiinamyllynkatu 4-8, FIN-20520 Turku, Finland. Fax Int + 358-21-2613920; E-mail: jari.forsstrom@utu.fi

Neural computing has been studied already for several decades. A single layer perceptron was one of the first developments in neural networks. However, it had limitations which Minsky and Papert [1] described in their critical review of the single-layer perceptrons. This decreased the interest in neural networks remarkably. It was shown that a multi-layered perceptron is needed to be able to learn non-linear problems. In those days, lack of a proper learning algorithm was the major problem. After Rumelhart and his colleagues [2] introduced an effective learning algorithm for multi-layer neural networks, the backpropagation algorithm, the use of neural networks in a variety of different tasks increased rapidly.

However, multi-layer networks still have their own drawbacks:

1. The knowledge that has been learned is included in the weights of the net and cannot be interpreted in rules.
2. The amount of cases needed to ensure reliable learning results has to be large.
3. The learning, especially in large networks, takes a very long time.
4. No domain knowledge can be included in the learning phases.

How harmful these facts turn out to be, depends on the problem itself. Obviously, in medicine, we often meet all these difficulties. Let us take the problem described in this paper, the laboratory diagnosis of acute appendicitis. Clinicians undoubtedly insist on knowing the founding of the suggestion given by a computer.

Multi-layer neural networks are not the only methods available for building non-linear models. Another powerful non-linear method is fuzzy logic. Traditionally, in this method the original attribute values are changed to logical ones using triangular membership values. The determination of these membership values is most often based on the knowledge of an expert. However, the combination of fuzzy logic with neural network learning methods makes it possible to use preclassified example cases to learn membership functions. Several examples of these neuro-fuzzy methods can be found in medical literature [3-9]. We have developed a neuro-fuzzy tool that utilizes the delta rule adopted from the backpropagation algorithm and a network structure which allows us to visualize the knowledge and transform it to fuzzy rules in a logic program.

MATERIALS

Acute appendicitis

Altogether 186 patients operated on for a clinical diagnosis of acute appendicitis in the University Central Hospital of Turku were included in the present study. Laboratory measurements were performed on blood samples obtained on admission to hospital. White blood cell count (WBC), C-reactive protein (CRP) and phospholipase A2 (PLA2) were used to predict the presence of acute appendicitis.

We divided the material into two groups: 120 cases (95 with appendicitis) were used for learning (learning set) and the remaining 66 (50 with appendicitis) for testing (testing set). For DiagaiD and neural network methods, the prevalence of appendicitis and non-appendicitis was made equal by taking non-appendicitis cases into the learning set several times. Thus we had in those sets 95 cases with appendicitis and 95 examples without appendicitis.

Myocardial infarction

Altogether 233 patients admitted to the Turku University Central Hospital because of chest pain were included in the present study. To build the decision support for interpreting the laboratory tests we used five measurements. Creatinine kinase at admission (CK1), after 6 hours (CK2) and after 12 hours from admission (CK3) and lactate hydrogenase after 6 hours (LD1) and 12 hours from admission (LD2).

We divided the material into two groups. 96 cases (48 with MI, 48 non-MI) were used for learning (learning set) and the rest, 137 (67 MI, 70 non-MI) for testing (testing set). Because the laboratory diagnosis of MI is based much upon the trends of the enzyme levels we decided to calculate the differences between successive measurements. In the net we used the five attributes: CK1, LD1, differences between CK1 and CK2, CK2 and CK3, and between LD2 and LD1. The diagnoses of the patients were based on other laboratory tests and ECG as well. However, we were interested in studying the performance of our method and made the task more difficult by using only CK and LD measurements in the model.

METHODS

DiagaiD

The DiagaiD architecture consists of three submodules that are closely linked together:

1. DiagaiD@Extract (DE)
2. DiagaiD@PreProcess (DP)
3. DiagaiD@CreateSystem (DC)

DiagaiD@Extract. DE is used for extracting the clinical data from the hospital information system. In this module, the user describes the problem in terms of attributes, *e.g.*, laboratory test results and outcomes, *e.g.*, diagnoses. The system can search data from the following databases:

- laboratory test results
- diagnoses (ICD-9-code)
- medication (ATC-code)
- codes for histopathological diagnoses
- operations (international classification of operations).

Attributes can be selected from a list which includes the standard nomenclature of all the laboratory tests used in the hospital as well as all the codes mentioned above. The search can be restricted by giving conditions on the range of laboratory values, sex, age, medications, diagnoses, etc. In this case study the search would be as follows:

- *search laboratory tests:*
  C-reactive protein (CRP)
  White blood cell count (WBC)
  Phospholipase A2 (PLA2)
- *and histopathological diagnoses:*
  code
- *on those patients who:*
  have had appendectomy
  have had above mentioned tests made

With a Windows user interface the search is easy to formulate by clicking the desired option and selections on the screen. By a code generator this search is translated automatically into MUMPS-code, which is the operating system in our hospital information system. The MUMPS program is sent to the hospital information system and as a result a datafile is sent on to the clinician's PC.

DiagaiD@PreProcess. This module includes the connectionist learning procedure. Because clinicians have already some kind of insight on how to interpret the laboratory test which would help the computer to reach the solution more rapidly, they only give their guess on the shape of membership function. In our tool, we have included four different curve models: positive gaussian, negative gaussian, positive sigmoid and negative sigmoid curve. Both functions have only two parameters that can be adjusted to find out the best-fitting curve for the attribute used. For a more detailed description of this module see also the article by Eklund and Forsström in this same issue [12].

The network structure is determined by the data to be learned. It is a single-layer perceptron, where attributes and outcomes are represented by input nodes and output nodes, respectively. In a single-layer perceptron the output is calculated as:

$$o_p = f\left(\sum_{i=1}^{n} w_i s_i\right) \tag{1}$$

and the learning rule is:

$$\Delta_p w_i = \eta(o_p)(t_p - o_p) s_i \tag{2}$$

This single-layer perceptron with sigmoid transfer function is only a linear method which calculates logistic regression [10], where the output of the network is:

$$o_p = \frac{1}{1 + e^{-(\Sigma s_i x_i)}} \tag{3}$$

In order to enhance the performance of the net we will replace the original attribute value with a function including adjustable parameters. The original attribute values are first transferred between 0 and 1 so that the maximum value becomes 1 and minimum value becomes 0. For each attribute we can select a desired type of function based on the clinical knowledge of the expert. In this study, the values of all three attributes increase in infection and clinically it is feasible to use a positive sigmoid function for all of the attributes. In a sigmoidal function, we use two adjustable parameters: α which determines the steepest part of the curve (in clinical terms the cut-off point), and β, which determines the slope of the curve. When tuning sigmoidal functions

$$g[\alpha\beta](r) = \frac{1}{1 + e^{-\beta(r-\alpha)}}, \beta > 0 \tag{4}$$

with

$$o_p = f(\sum_{i=1}^{n} w_i \, g[\alpha_i,\beta_i](r_i)) \tag{5}$$

the corresponding delta rules become (6) and (7), respectively (see below).

With this method we get a non-linear model where the output of the net is:

$$o_p = f(\sum \frac{w_i}{1 + e^{-\beta_i (r_i - \alpha_i)}}) \tag{8}$$

Learning takes place stepwise. Adjustable parameters are the weights, cut-off points, and the slope for each attribute. Before starting the learning phase the parameters can be adjusted according to the clinician's estimation. For example, the cut-off level can be adjusted to match with the upper border of the reference value of the parameter. Also some good guesses about the weights can be given. If no background information is given, the initial parameter values for each attribute will be: α = 0.5, β = 10, weight = 1. With these values the cut-off point is located at the mean of the largest and the smallest original value for each attribute.

For each iteration, we adjust only one of the three parameters. To find out the fuzzy membership function we start adapting only the αs and βs. We use off-line learning which means that the corrections for the parameters are made only after each iteration, not after each case. We used a constant learning factor $\eta' = 0.01$. As a result, we get transformation curves for the three attributes which are the membership functions for the rules. In the last phase, we adjust the weights. Each phase can be repeated several times, while the learning factor, order adjustments of parameters and number of iterations in each phase can be selected by the user. As a result, we will get the weights for each attribute showing the influence of the attribute in the diagnosis, as well as cut-off points and slopes which can be interpreted as fuzzy transformations from original values to logical values between 0 and 1.

DiagaiD@CreateSystem. This module transforms the knowledge to a fuzzy logic program. The details of this module are described elsewhere [11].

Backpropagation neural network

As a conventional neural network model we used a backpropagation network with one hidden layer, 3 input nodes and one output node. We tested networks with 2, 3 and 4 hidden nodes. Two different datasets were tested as inputs: 1) the original laboratory test values; 2) logical values which were transformed from the original values exploiting the membership functions found by the DiagaiD approach. Each network and dataset was tested three times. The networks were started every time with random weights, learning factor h was 0.02, momentum term was 0.7 and the number of iterations was 10000.

$$\Delta_p\alpha_i = -\frac{\partial E_p}{\partial \alpha_i} = -\eta(t_p - o_p)o_p(1 - o_p)w_i g[\alpha_i,\beta_i](r_i)(1 - g[\alpha_i,\beta_i](r_i))\beta_i \tag{6}$$

$$\Delta_p\beta_i = -\frac{\partial E_p}{\partial \beta_i} = \eta(t_p - o_p)o_p(1 - o_p)w_i g[\alpha_i,\beta_i](r_i)(1 - g[\alpha_i,\beta_i](r_i))(r_i - \alpha_i) \tag{7}$$

Analysis of the results

Statistical analysis was performed using receiver operating characteristic (ROC) curves [13].

RESULTS

Acute appendicitis

The parameters found using DiagaiD are presented in Table I. With logistic regression we obtained a formula for probability of the appendicitis: P(appendicitis) = 1/(1+exp(3.0897 + 0.0012 x PLA2 + 0.2338 x WBC + 0.029 x CRP)).

The performance of classification of the test set measured using the area under ROC-curve is presented in Table II. The differences between groups did not reach statistical significance.

Myocardial infarction

The parameters found using DiagaiD are presented in Table III. With logistic regression we got a formula for probability of the MI: P(MI) = 1/(1+exp-(-3.2452 - 0.001 x CK1 + 0.0001 x (CK2-CK1) + 0.0027 x (CK3-CK2) + 0.0065 x LD + 0.0016 x (LD2-LD1))).

The performance of classification of the test set measured using the area under ROC-curve is presented in Table IV. The p-values from the comparisons of ROC-curves between the methods in myocardial infarction data are shown in table V.

DISCUSSION

The neuro-fuzzy method used here turned out to be an effective tool for formalizing and extracting clinical knowledge from example cases. Because the model to be learned is quite strictly defined by the user, the risk of sticking into a poor local minimum is smaller than in backpropagation neural networks. In backpropagation neural networks, the learning results varied greatly when original data was

TABLE I. The parameters after tuning.

Attribute	Alpha	Beta	Weight	Actual cut-off
PLA2	0.30	25.29	0.41	160.0 (mg/l)
WBC	0.38	48.69	1.43	13.1 (10^9/l)
CRP	0.24	27.40	0.46	64.0 (mg/l)

TABLE II. The performance of different methods on classification of the test set. For neural network approach the range of performance of three different tests is also shown (AUC = area under ROC curve, SEM = standard error of the mean).

METHOD	AUC	SEM	RANGE
Logistic regression	0.6775	0.0710	
DiagaiD	0.6825	0.0728	
Backpropagation (original data)			
2 hidden nodes	0.6363	0.0813	0.5373-0.6363
3 hidden nodes	0.5537	0.0819	0.5309-0.5537
4 hidden nodes	0.6469	0.0747	0.5606-0.6469
Backpropagation (transformed data)			
2 hidden nodes	0.6219	0.0763	0.5400-0.6219
3 hidden nodes	0.6069	0.0756	0.5869-0.6069
4 hidden nodes	0.6075	0.0732	0.5838-0.6075

TABLE III. The parameters after tuning.

Attribute	Alpha	Beta	Weight	Actual cut-off
CK1	0.41	21.20	0.68	896(IU//l)
CK2-CK1	0.35	23.27	1.02	3300(IU/l)
CK3-CK2	0.60	46.39	1.01	339(IU/l)
LD1	0.09	100.95	1.32	291(IU/l)
LD2-LD1	0.12	83.05	1.15	-160(IU/l)

TABLE IV. The performance of different methods on classification of the test set. (AUC = area under ROC curve, SEM = standard error of the mean).

METHOD	AUC	SEM
Logistic regression	0.7038	0.0453
DiagaiD	0.7789	0.0393
Backpropagation (original data)		
2 hidden nodes	0.6320	0.0473
3 hidden nodes	0.6424	0.0472
4 hidden nodes	0.7270	0.0441
5 hidden nodes	0.7052	0.0449
Backpropagation (transformed data)		
2 hidden nodes	0.7614	0.0405
3 hidden nodes	0.7467	0.0424
4 hidden nodes	0.7505	0.0417
5 hidden nodes	0.7414	0.0429

TABLE V. Significance levels (p-values) between the comparisons in myocardial infarction. For comparisons only the best net from backpropagation (BP) using original data (4 hidden nodes) and the best from backpropagation using transformed data (2 hidden nodes) are used.

	BP(ori)	BP(trans)	DiagaiD
Logistic regression	0.68	0.18	0.063
BP (original data)		0.52	0.39
BP (transformed data)			0.67

used. When transformed data was used the results were not significantly better than with original data. However, the risk of local minima seemed to be smaller and, accordingly, the results in networks with 3 or more hidden nodes seemed to be more consistent than those obtained using original data. We would have expected that a backpropagation network with transformed data would have given better results than DiagaiD alone because using a multi-layer perceptron it is possible to learn more complex models than with our single layer perceptron with data transformation. However, the results for backpropagation were poorer. This gives support to our vision that we can build non-linear models with good performance in a logically understandable way. The fact that single layer perceptron with data transformation is not able to find as complex models as multi layer perceptron, may even be an advantage when the number of example cases is quite small. With sigmoid transformation functions (as in this study) the DiagaiD method is a kind of non-linear extension of logistic regression. Accordingly, it is not surprising that DiagaiD is as good or better than conventional logistic regression.

Generalization is an essential part of machine learning. It has been shown that neural nets that learn the learning set too well have poorer performance on testing cases. This overlearning is more common if large networks are used. If the size of a network is limited it is forced to generalize and, accordingly, the performance on unseen cases will be better. We think that the way we combined clinical knowledge and learning from examples in DiagaiD keeps the size of the network small enough to avoid the risk of overlearning.

When we look at the adjusted values from DiagaiD it can be seen that the betas are quite large values, showing that the system uses very sharp cut-off values. This may be due to the order the parameters are taught. There is a complex interrelation among the three adjustable parameters. For example, if weights are adjusted earlier they become more prominent and the betas might remain smaller. More experiments are needed to find out which is the most feasible order in which to adjust the parameters.

In fuzzy logic programming, triangular-shaped membership functions are most often used. This seems to be efficient in most cases. To learn linear membership functions from examples we also need to use adjustable variables. Accordingly, we decided to select sigmoid function instead, because we think that this function is closer to reality in clinical decision making. Gaussian curve is needed if the relation between original values and logical values is a J-shaped curve. Many other possibilities exist, but we think that functions with only two adjustable parameters for data transformation are the most feasible.

In this method, much emphasis is put into processing of the original data. By applying this parameter tuning we have been able to overcome some of the drawbacks of multi-layer networks. The logical values derived from this parameter tuning can be utilized not only by including the rules to a fuzzy program but also as an intelligent transformation of the original values before they are fed to the classical multi-layer networks. This parameter tuning may enhance the learning result and speed up the learning. As a rule of thumb it has been issued that for a neural network to give repeatable results the number of cases should be ten times the number of parameters adjusted. In traditional multi-layer networks the number of cases needed limits the use of this method with small data sets. In this neuro-fuzzy tool we have only three adjustable variables for each attribute which makes it suitable for quite a small dataset as well.

With the two example problems we have shown that single layer perceptron with relevant data preprocessing can be successfully applied to laboratory medicine. The results with the method are equal or even slightly better than those obtained using multilayer perceptron and backpropagation. Even more important is that the knowledge learned with this system can be easily understood by clinicians.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of The Finnish Academy, the Technology Development Center in Finland, and The Turku University Foundation.

REFERENCES

1 Minsky ML, Papert S. Perceptrons. MIT Press, Cambridge MA: 1969.

2 Rumelhart DE, Hinton GE and Williams RJ. Learning internal representations by back-propagating errors. Nature 1986; 323: 533-536.

3 Walczak B, Wegschneider W. Calibration of non-linear analytical system by neuro fuzzy approach. Chemometrics and Intelligent Laboratory Systems 1994; 22: 199-207

4 Barreto JM, DeAzevedo FM. Connectionist expert systems as medical decision aid. Artificial Intelligence in Medicine 1993; 5: 515-523.

5 Kuncheva L. An aggregation of PRO and CON evidence for medical decision support systems. Comput Biol Med 1993; 23: 17-424.

6 De B, Harrington P, Pack BW. FLIN: Fuzzy linear interpolating network. Analytica Chimica Acta 1993; 277: 189-197.

7 Molnar B, Szentirmay Z, Bodo M, Sugar J, Feher J. Application of multivariate fuzzy set and neural network analysis in quantitative cytological examinations. Anal Cellular Pathol 1993; 5: 161-175.

8 Harrington PB. Minimal neural networks: concerted optimization of multiple decision planes. Chemometrics and Intelligent Laboratory Systems 1993; 18: 157-170.

9 Simutis R, Havlik I, Lubbert A. Fuzzy-aided network for real-time state estimation and process prediction in the alcohol formation step of production-scale beer brewing. J Biotechn 1992; 27: 203-215.

10 Hripcsak G. Using connectionist modules for decision support. Meth Inform Med 1990; 29: 167-181.

11 Eklund P, Forsström J, Holm A, Nyström M, Selén G. Rule generation as an alternative to knowledge acquisition. A system architecture for medical informatics. Fuzzy Sets and Systems 1994; 66: 195-205.

12 Eklund P, Forsström JJ. Computational intelligence for laboratory information systems. Scand J Clin Lab Invest 1995; 55 (222): 21-30.

13 Zweig MH, Campbell G. Receiver operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine. Clin Chem 1993; 39: 561-577.