NEURAL METHODS FOR ANALYSIS OF PSYCHOMETRIC DATA.

Włodzisław Duch, Rafał Adamczak, Krzysztof Grąbczewski

Department of Computer Methods, Nicholas Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland.
E-mail: duch,raad,kgrabcze@phys.uni.torun.pl.

Abstract: Neural networks were used to extract crisp logic rules from the MMPI psychometric questionnaires. These rules are used in an expert system that is capable of assigning probabilities of different diagnoses and verbal interpretations for each case. Similar techniques may be used for analysis of any questionnaire-based data.

I. INTRODUCTION: THE PROBLEM

Psychometric tests are performed quite often to support clinical decisions, evaluate suitability of applicants for jobs requiring special skills, or even to determine whether permission for carrying a gun should be granted. Computerized versions of many tests were developed, but assessment of their results still requires an expert psychologist. We have been approached by psychologists working in the Academic Psychological Clinic of Nicholas Copernicus University who use the Minnesota Multiphasic Personality Inventory (MMPI) test (hundreds of books and papers about this test exist, cf. review in [1]), consisting of 550 questions with yes/no/don't know answers. Questions range from general health problems and specific neurological symptoms to personal, political and moral opinions.

Existing computerized versions of the MMPI test assist only in information acquisition. Our goal is to provide automatic psychological diagnosis. Rule based system is most desirable because a detailed interpretation, including description of personality type, may be assigned to each diagnosis. Unfortunately existing rules for interpretation of such tests are based on intuition, not on data mining, and seem to be rather inaccurate. To find the rules a larger collection of diagnosed cases had to be accumulated and analyzed. There are many machine learning methods for rule induction that could be used for this task instead of neural networks. There is a strong competition from decision trees [2], which are fast, accurate and can easily be converted to sets of logical rules. Some inductive methods of machine learning [3], [4] have been used for quite a long time for extraction of rules from the data. Fuzzy logic systems [5] are also oriented towards rule-based description.

Despite this competition neural networks seem to have important advantages, especially for problems with continuous-valued inputs. Good linguistic variables (collection of symbols, or intervals, representing linguistic concepts appearing as conditions in the rule antecedent) may be determined simultaneously with logical rules, selection of features and aggregation of features into smaller number of more useful features is possible, adaptation mechanisms are build into the method, wide-margin classification provided by neural networks make logical rules more robust. We have developed a complete methodology for logical rule extraction, optimization and application [6]-[7]. The rule extraction step is based on several neural methods, including the multilayer perceptron (MLP) and probability density estimation methods. These methods may be used either directly on the data, or to find an approximately equivalent logical description for a trained neural network, i.e. to understand what neural networks really do. Although neural systems may perform useful classification in many applications humans do require a comprehensible description of the data, or knowledge instead of black-box recommendations. Results obtained with our rule extraction methods were compared on many benchmark datasets with results obtained by other systems. In most cases our methods gave much simpler and more accurate logical rules. These methods are described very briefly in the next section, and their application to the psychometric data is in the third section. Problems specific to the questionnaire-based data analysis are presented in the final section.

II. METHODOLOGY OF LOGICAL RULE EXTRACTION

Logical rules require symbolic inputs (linguistic variables), therefore the input data have to be quantized first, i.e. features defining the problem should be identified and their values (sets of symbolic or integer values, or continuos intervals) labeled.

Linguistic (logical) input variables s_k, for integer or symbolic variables x_i, taking values from a finite set of elements $\aleph_i = \{X_i^{(j)}\}$, are defined as true if x_i belongs to a specific subset $\aleph_{ik} \subseteq \aleph_i$. For continuos input features x_i linguistic variable s_k is obtained by specifying an open or closed interval $x_i \in [X_k, X'_k]$ or some other predicate function $P_k(x_i)$. Each continuos input feature x_i is thus replaced by two or more linguistic values for which predicates P_k are true or false. For neural networks a convenient representation of these linguistic values is obtained using vectors of predicates, for example V_{s1} = (+1,−1, −1...) for linguistic variable s_1 or V_{s2} = (−1,+1, −1...) for s_2 etc.

Initial values of the intervals for continuos linguistic variables may be determined by the analysis of histograms, dendrograms, decision trees or by clusterization methods. FSM, our probability density estimation neurofuzzy network, is initialized using simple clusterization methods [7], for example dendrogram analysis of the input data vectors (for very large datasets rounding of continuous values to lower precision is applied first, and then duplicate data vectors are sorted out to reduce the number of input vectors). Rectangular functions are used in FSM for a direct extraction of logical rules, or soft trapezoidal functions described below are used.

MLP (multilayered perceptron) and other neural models may find linguistic variables by training **a special neural layer** [6] of L-units (**linguistic units**). Each *L*-unit is a combination of two sigmoidal functions realizing the function $L(x_i;b_i,b_i',\beta) = \sigma(\beta(x_i-b_i)) - \sigma(\beta(x_i- b_i'))$, parameterized by two biases *b, b'* determining the interval in which this function has non-zero value. The slope β of sigmoidal functions $\sigma(\beta(x-b))$ is slowly increased during the learning process, transforming the fuzzy membership function ("soft trapezoid") into a window-type rectangular function. Similar smooth transformation is used in the FSM network using bicentral transfer functions, which are combinations of products of $L(x_i;b_i,b_i',\beta)$ functions [7] with some additional parameters.

Construction and training of neural network follows the initial definition of linguistic variables. To facilitate extraction of logical rules from an MLP network one should transform it smoothly into something resembling a network performing logical operations (Logical Network, LN). This transformation gave the name MLP2LN [6] to the method. Skeletonization of a large MLP network (i.e. leaving only a few most important connections) is the method of choice if our goal is to find logical rules for an already trained network. Otherwise it is simpler to start from a single neuron and construct the logical network using the training data. A smooth transition from MLP to a logical-type of network performing similar mappings is advocated. This is achieved by:

a) gradually increasing the slope β of sigmoidal functions to obtain crisp decision regions;
b) simplifying the network structure by inducing the weight decay through a penalty term;
c) enforcing the integer weight values 0 and ± 1, interpreted as 0 = irrelevant input, +1 = positive and −1 = negative evidence.

These objectives are achieved by two additional terms added to the mean square error function $E_0(W)$:

$$E(W) = E_0(W) + \frac{\lambda_1}{2}\sum_{ij} W_{ij}^2 + \frac{\lambda_2}{2}\sum_{ij} W_{ij}^2 (W_{ij}-1)^2 (W_{ij}+1)^2 \qquad (1)$$

The first term, scaled by the λ_1 hyperparameter, encourages weight decay, leading to skeletonization of the network and elimination of irrelevant features. The second term, scaled by λ_2, forces the remaining weights to approach ± 1 or 0, facilitating easy logical interpretation of the network function: features are

either necessary (+1), should not appear (−1) or are irrelevant (0). Relatively large λ_1 value is used at the beginning, followed by a large λ_2 value near the end of the training process. These parameters determine the simplicity/accuracy tradeoff of the generated network and extracted rules. A few experiments are sufficient to find the largest λ_1 value for which the MSE is still acceptable, and does not decrease quickly when λ_1 is decreased. Smaller values of λ_1 should be used to obtain more accurate networks (rules).

Logical rule extraction: rules $\Re_k$ implemented by trained networks are obtained in the form of logical conditions by considering contributions of inputs for each linguistic variable *s*, represented by a vector V_s. Contribution of variable *s* to the activation is equal to the dot product $V_s \cdot W_s$ of the subset W_s of the weight vector corresponding to the V_s inputs. A combination of linguistic variables activating the hidden neuron above the threshold is a logical rule of the form:

$$\Re = (s_1 \text{ AND } \neg\, s_2 \text{ AND } ... \text{ AND } s_k). \qquad (2)$$

In the **constructive version** of the MLP2LN approach (called C-MLP2LN) usually one hidden neuron per output class is created at a time and the training proceeds until the modified cost function reaches minimum. The input data that are correctly handled by the first group of neurons will not contribute to the error function, therefore weights of these neurons are kept frozen during further training. This is equivalent to training one neuron (per class) at a time on the remaining data, although sometimes training two or more neurons may lead to faster convergence. This procedure is repeated until all the data are correctly classified, weights analyzed and a set of rules $\Re_1 \vee \Re_2 ... \vee \Re_n$ is found for each output class or until the number of rules starts to grow rapidly. The network repeatedly grows when new neurons are added, and then shrinks when connections are deleted. Since the first neuron for a given class is trained on all data for that class the rules it learns are most general, covering largest number of instances.

Simplification of rules: the final solution may be presented as a set of logical rules or as a network of nodes performing logical functions. However, some rules obtained from analysis of the network may involve spurious conditions, more specific rules may be contained in general rules or logical expressions may be simplified if written in another form. Therefore after extraction rules are carefully analyzed and whenever possible simplified (we use a Prolog program for this step).

Optimization of rules. Gradient-based backpropagation methods have problems with finding the optimal values of adaptive parameters, and they break down when the slopes of sigmoidal functions are very high. Optimal linguistic variables (intervals) and other adaptive parameters may be found by maximization of predictive power of a rule-based classifier. Let $\wp(C_i,C_j|M)$ be the confusion matrix, i.e. the number of instances in which class C_j is predicted when the true class was C_i, given some parameters *M*. Then for *n* samples $p(C_i,C_j|M) = \wp(C_i,C_j|M)/n$ is the probability of (mis)classification. The best parameters of the model *M* are selected by maximizing the number (or probability) of correct predictions (the "predictive power" of rules): $\max_M \text{Tr}\ \wp(C_i,C_j|M)$, over all parameters in *M*, or minimizing the number of wrong predictions (possibly with some risk matrix $\boldsymbol{R}(C_i,C_j)$: $\min_M \sum_{i\neq j} \boldsymbol{R}(C_i,C_j)\ \wp(C_i,C_j|M)$.
Weighted combination of these two terms:

$$E(M) = \lambda \sum_{i\neq j} \boldsymbol{R}(C_i,C_j)\ \wp(C_i,C_j|M) - \text{Tr}\ \wp(C_i,C_j|M) \geq -n \qquad (3)$$

is bounded by $-n$ and should be minimized over parameters in *M* without constraints. For this minimization we have used simulated annealing and multisimplex global minimization methods. If λ is large the number of errors after minimization may become zero, but some instances may be rejected (i.e. rules will not cover the whole input space).

The input data is usually not quite precise, with true values given by Gaussian distribution $G_x=G(y;x,s_x)$ around the measured values *x*. In such a case one can estimate how well each rule is satisfied by the input

vector, or estimate the probability of different classes by performing Monte Carlo simulations drawing the vectors from G_x distribution. An analytical formula is given by a convolution of the step function representing crisp rule, and the Gaussian function representing the data, giving an error function $\mathrm{erf}(a-x) = \int_a^\infty G(y;x,s_x)dy$. The error function is approximated quite accurately (within 2%) by a sigmoidal function used in MLPs. Taking instead of the **erf** function a sigmoidal function corresponds to an assumption that the error distribution of x measurements is given by $\sigma(x)(1-\sigma(x))$, approximating Gaussian distribution with $s_x^2=1.7$ within 3.5%. If the rule involves closed interval $[a,b]$, $a \leq b$ the probability that it is fulfilled by a sample from the Gaussian distribution G_x representing the data centered at x is:

$$p(R_{a,b}(G_x=T)) \sim \sigma(x-a)-\sigma(x-b) = L(x;a,b,\beta) \quad (4)$$

Thus the probability that a given condition is fulfilled is proportional to the value of a soft trapezoid function realized by L-units. Crisp logical rules with assumption that data has been measured with finite precision lead to soft L-functions that allow to compute classification probabilities that are no longer binary. In fuzzy logic confidence factors that are not probabilities are frequently used. For complex, non-orthogonal rules with many conditions related to different attributes, it is indeed difficult to estimate probabilities that would agree with Monte Carlo simulations. For independent features x_i present in logical rule probabilities given by the equation above should be multiplied and if more conditions are present for the same feature x probabilities should be summed over all contributions.

III – APPLICATION TO THE PSYCHOMETRIC DATA

The psychometric data was collected in the Academic Psychological Clinic of Nicholas Copernicus University and in several psychiatric hospitals around Poland. We have worked with two datasets, one for woman, with 1027 cases belonging to 27 classes (normal, neurotic, drug addicts, schizophrenic, psychopaths, organic problems, malingerers, persons with criminal tendencies etc.) determined by expert psychologists, and the second for man, with 1167 cases and 28 classes. The answers to 550 questions were converted to 14 continuos scales using standard tables. These coefficients are displayed in a histogram, called a "psychogram", allowing skilled psychologists to diagnose specific problems. The first four coefficients are used for control, measuring consistency of answers or the number of "don't know" answers, allowing to find malingerers. The next 10 coefficients form clinical scales, developed to measure tendencies towards hypochondria, depression, hysteria, psychopathy, paranoia, schizophrenia etc. For example values between 70 and 80 in the hypochondria scale may be interpreted as "very strong worries about own health, leading to psychosomatic reactions". A large number of simplification schemes has been developed to make the interpretation of psychograms easier. They may range from rule-based systems derived from observations of characteristic shapes of psychograms, statistical discrimination functions, or systems using smaller number of aggregated coefficients. It should be very interesting to evaluate the accuracy of this conversion process, based on experience of experts in psychometry, but it is not possible since all diagnosis are made on the basis of reduced information.

Rule based system is most desirable because a detailed interpretation, including description of personality type, may be assigned to each diagnosis. Rules were generated using C4.5 classification tree [2], a very good classification system which may also generate logical rules, and the FSM neural network [8]. These two systems are the easiest to use; we have also used several other networks to generate rules for a few classes. For the first dataset C4.5 created 55 rules, achieving 93.0% of correct responses. Assuming about 1% inaccuracy of measurements increases accuracy to 93.7%. FSM (with rectangular membership functions) generated 69 rules, agreeing in 95.4% with diagnosis by human experts. Gaussian fuzzification at the level of 1.1-1.5% increases accuracy to 97.6%. For the second dataset C4.5 created 61 rules giving 92.5% accuracy, while FSM generated 98 rules giving 95.9% accuracy. After fuzzification 93.1% and 96.9% were obtained. Some rules cover only few cases from the database, therefore further pruning and reoptimization is desirable. These results are for the reclassification accuracy only using generated sets of

rules. Statistical estimation of generalization by 10-fold crossvalidation gave 82-85% correct answers with FSM (crisp unoptimized rules) and 79-84% correct answers with C4.5. Fuzzification improves FSM crossvalidation results to 90-92%.

In general C4.5 gives slightly lower accuracy with most rules based on open intervals, and therefore harder to interpret. For most classes there were only a few errors and it is quite probable that they are due to the psychologists interpreting the psychogram data or due to the complex problems that cannot be uniquely classified. Two classes, organic problems and schizophrenia, are difficult since their symptoms are easily confused with symptoms belonging to other classes. On average 2.5-3 logical rules per class were derived, involving between 2 and 9 attributes. These rules are most accurate on the available data if about 1% of the uncertainty of measurement in each of the scales is assumed, corresponding to a Gaussian dispersion centered around measured values. Larger uncertainties, on the order of 5%, lead to about the same number of classification errors as the original crisp rules, but provide softer evaluation of possible diagnoses, assigning non-zero probabilities to classes that were not covered by slightly fuzzified rules. Since our training set is relatively small (considering the number of classes) such fuzzy evaluations are frequently useful to clinical psychologist.

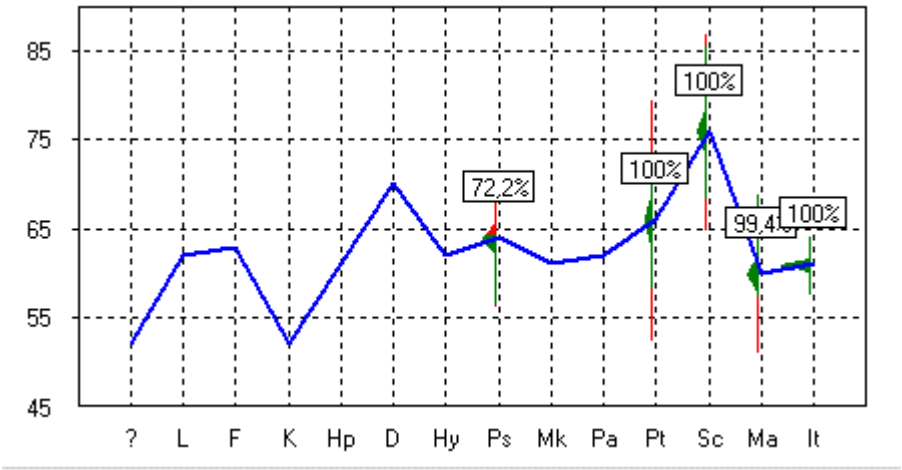
An example of a psychogram with rule conditions shown as vertical bars is shown below. The rule:

IF $Ps(X) \in [56.5,64.5] \wedge Pt(X) \in [58.5,73.5] \wedge Sc(X) \in [68.5,85.5] \wedge$
$Ma(X) \in [57.5,69.5] \wedge It(X) \in [57.5,66.5]$ Then Schizofrenia-men

has 5 conditions and the actual case X characterized by the values of 14 scales

52 62 63 52 61 70 62 64 61 62 66 76 60 61 men

is accepted by that rule with 71.8% probability, calculated with assumption of Gaussian uncertainties shown on the vertical bars for each condition. The rule condition for the Ps (psychostenia) scale fits with only 72.2% to the measured value. This means that this value is close to the interval boundary.



Each rule has detailed interpretation associated with it by psychologists. Fuzzification leads to additional adjectives in verbal interpretation, like “strong tendencies”, or “typical”. A book has been published (in Polish), containing detailed description of the rule generation methodology and interpretation of all the rules [9]. An expert system using these rules and associated interpretation is already used by a few clinical psychologist and should be commercially available in the near future.

IV - CONCLUSIONS

Automatization of questionnaire evaluation may have wide applications. We have presented an approach that tries to combine the ease of interpretation of a rule-based logical expert system, with high accuracy and probabilistic interpretation provided by fuzzification of inputs. Unfortunately the system can work only as well as the data it gets and in case of psychometric data evaluations are always to some degree subjective. Perhaps one should be satisfied with a system that makes diagnoses at the human level. The only way to test it would be to ask a few experts for diagnosis of many cases and check the mean and the variance of such ensemble. Assuming that a group of experts improves the reliability of the diagnosis one could easily evaluate performance of individual experts as well as the automatic system. Unfortunately there are no databases of such kind that we could use.

Dealing with a large number of classes and thus large number of rules means that one has to use an efficient system for rule generation. Our system has not been yet completely automated, therefore FSM system was used for initial rule generation, although on simpler benchmark problems MLP2LN has always been more accurate. Since the number of cases that we have is only about 1000 many different sets of logical rules may give the same accuracy. Although quite different rules are produced a small variance in crossvalidation tests shows that they are quite reliable. It may be interesting for the psychometric experts to analyze many different sets of rules for each class, showing equivalent logical descriptions. IncNet, another neural network model used in our group [10], obtained 93-95% accuracy in crossvalidation tests (thanks are due to Norbert Jankowski for providing the results). Any reliable neural network or statistical model is suitable as a generator of a large number of additional training data vectors for our rule extraction system. In this way we may obtain sets of rules which are more accurate and stable.

The same methodology may be used for data mining of many other types of questionnaire-based data.

Acknowledgement: Support by the Polish Committee for Scientific Research, grant 8 T11F 014 14, is gratefully acknowledged.

REFERENCES

[1] J.N. Butcher, S.V. Rouse, *Personality: individual differences and clinical assessment.* Annual Review of Psychology 47, 87 (1996)
[2] J.R. Quinlan, *C4.5: Programs for machine learning*. San Mateo, Morgan Kaufman 1993
[3] D. Michie, D.J. Spiegelhalter and C.C. Taylor, *Machine learning, neural and statistical classification*. Elis Horwood, London 1994
[4] T. Mitchell, *Machine learning*. McGraw Hill 1997
[5] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, MIT Press (1996)
[6] W. Duch, R. Adamczak and K. Grąbczewski, *Extraction of logical rules from backpropagation networks.* Neural Processing Letters 7, 1-9 (1998)
[7] W. Duch, R. Adamczak and K. Grąbczewski, *Methodology of extraction, optimization and application of crisp and fuzzy logical rules* (submitted to the IEEE Transactions on Neural Networks)
[8] W. Duch, G.H.F. Diercksen, *Feature Space Mapping as a universal adaptive system*, Computer Physics Communication, 87: 341-371, 1995
[9] W. Duch, T. Kucharski, J. Gomuła, R. Adamczak, Metody uczenia maszynowego w analizie danych psychometrycznych. Zastosowanie do wielowymiarowego kwestionariusza osobowości MMPI-WISKAD (Toruń, March 1999; 650 pp.)
[10] N. Jankowski, V. Kadirkamanathan. *Statistical control of RBF-like networks for classification*. 7th Int. Conf. on Artificial Neural Networks, Lausanne, Switzerland 1997, pp 385–390. Springer Verlag.