Neuro-Wavelet Classifiers for EEG Signals Based on Rough Set Methods

Marcin Szczuka Piotr Wojdyłło

Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
{szczuka,pwoj}@mimuw.edu.pl

1 Introduction

The issue of analysing data that is presented to us in the form of signal measured over the period of time using analog devices may be complicated one. Especially in case of multichannel source like electroencephalogram (EEG) we have to overcome several difficulties. The amount of data, its internal complexity, possibility of errors and disturbances forces us to search effective methods for extraction of relevant and valid information in the way that is computationally affordable and intuitively understandable.

Numerous attempts to tackle EEG were performed. Since EEG is one of the most important sources of information in therapy of epilepsy, several researchers tried to address this topic. They used approaches coming from area of signal processing (e.g. wavelet analysis) as well as approximate methods originating in Artificial Intelligence. Some results have been reported in [22], [23] and recently in [20].

In our work we trying to establish the tool for classification of EEG signals. Particular data we are dealing with is connected to dissemination of different kinds of epilepsy. By identifying features in the signal and finding their importance we want to provide automatic system that will be able to support physician in diagnosing process by providing suggestions and focusing his attention on the most important elements of the EEG.

Initial set of data comprises of 44 cases, each represented as the vector 5736 real-valued measurements. Those values correspond to 2,5 second sample with frequency 102,4 Hz taken by 21 electrodes placed on the scalp. The patients are classified to one of two groups depending of the nature of epilepsy they suffer. First of those groups (group B) corresponds to cases of posttraumatic epilepsy, second (group A) gathers patients suffering epilepsy caused by other facts (e.g. endogenic). Initial set of data contains information that some cases definitely belong to one of the classes and some others probably belong to class A and B respectively. For the purpose of our research we unified groups A and B by treating sure and probable cases in the same way.

This data was already studied in [25] with some interesting outcome. That approach is referred in the paper as the WaRS method. It was a start point to the extensions that are core of this study.

Since original data contained large amount of information as compared to the number of cases we decided to enrich it by adding new cases produced on the basis of existing ones using wavelet *noisification* technique. This and other wavelet techniques allowed further construction of automatic classification system. Application of wavelet analysis benefited to our study by reducing size of data, identifying important features and improving stability of solution.

In order to have better flexibility we employed a simple neural network at the final step of classifier construction. The overall performance of our system was improved in that way.

It is important to stress here that the entire process of data preparation and construction of classification support was directed towards simplicity, compactness, generality and intuitive understanding of derived solution. Several times when the trade-off between quality (accuracy) and transparency happened we chose rather to preserve clarity than perfect fitting.

The paper starts with very short introduction to basic notions and language used in three major fields that contributed to this study, namely, rough sets,scaling and neural networks. Having basic notions handy in the next section we introduce initial WaRS model of classification and discuss some of its features. Next part describes modifications to the initial concept of classifier and their consequences for both classification quality and classifier construction. Proposed methods undergone experimental verification and results of those experiments close this part of paper. At the end we present our conclusion regarding results obtained.

2 Basic notions

2.1 Rough sets and rules

The structure of data that is subject of our study is represented in the form of *information system* [19] or, more precisely, the special case of information system called *decision table.*

Information system is a pair of the form $\mathbf{A} = (U, A)$ where U is a *universe* of *objects* and $A = (a_1, ..., a_m)$ is a set of *attributes* i.e. mappings of the form $a_i : U \rightarrow V_a$, where V_a is called *value set* of the attribute a_i. The decision table is also a pair of the form $\mathbf{A} = (U, A \cup \{d\})$ where the major feature that is different form the information system is the distinguished attribute d. In case of decision table the attributes belonging to A are called *conditional attributes* or simply *conditions* while d is called *decision* (sometimes *decision attribute*). We will further assume that the set of decision values is binary i.e. $V_d = \{d_1, d_2\} = \{\text{A,B}\}$. The $i-$th *decision class* is a set of objects $C_i = \{o \in U : d(o) = d_i\}$, where d_i is the $i-$th decision value taken from decision value set $V_d = \{\text{A,B}\}$. In our particular case the set of conditional attribute values will be either finite (in case of scaled or rule-based tables) or contained in some interval of real domain (in case of original and wavelet-processed data).

For any subset of attributes $B \subset A$ *indiscernibility relation* IND(B) is defined as follows:

$$xIND(B)y \Leftrightarrow \forall_{a \in B} a(x) = a(y) \tag{1}$$

where $x, y \in U$.

Having indiscernibility relation we may define the notion of reduct. $B \subset A$ is a *reduct* of information system if $IND(B) = IND(A)$ and no proper subset of B has this property. Intuitively, reduct is the minimal set of attributes that allows us to preserve ability to distinguish objects at the same level as for original decision table.

Decision rule is a formula φ of the form

$$(a_{i_1} = v_1) \wedge ... \wedge (a_{i_k} = v_k) \Rightarrow d = v_d \tag{2}$$

where $1 \leq i_1 < ... < i_k \leq m$, $v_i \in V_{a_i}$. The set of all rules for a particular decision table $\mathbf{B} \subset \mathbf{A}$ we denote by $RUL(\mathbf{B})$. Atomic subformulae $(a_{i_1} = v_1)$ are called *conditions.* We say that rule r is *applicable* to object, or alternatively, the object *matches* rule, if its attribute values satisfy the premise of the rule. With the rule we can connect some characteristics. *Support* denoted as $Supp_{\mathbf{A}}(r)$ is equal to the number of objects from $\mathbf{A}$ for which rule r applies correctly i.e. premise of rule is satisfied and the decision given by rule is similar to the one preset in decision table. $Match_{\mathbf{A}}(r)$ is the number of objects in $\mathbf{A}$ for which rule r applies in general. Analogously the notion of matching set for a collection of rules may be introduced. By $Match_{\mathbf{A}}(R, o)$ we denote the subset M of rule set R such that rules in M are applicable to the object $o \in U$. The rule is said to be *optimal* if removal of any of its conditions causes decrease of its support. Support and matching are also used to define coefficient of consistency $\mu_{\mathbf{A}}(r)$ for a rule, being equal to $\mu_{\mathbf{A}}(r) = \frac{Supp_{\mathbf{A}}(r)}{Match_{\mathbf{A}}(r)}$.

In our study we refer only to the rules that are derived using knowledge contained in data. In that view we may introduce the *meaning* of the premise of rule r in the decision table $\mathbf{A}$. The meaning of $Pred(r)$ will be denoted by $|\Pr red(r)|_{\mathbf{A}}$ and defined inductively in the following way:

(1) if $Pred(r)$ is of the form $a = v$ then $|Pred(r)|_{\mathbf{A}} = \{o \in U : a(o) = v\}$
(2) $|Pred(r) \wedge Pred(r')|_{\mathbf{A}} = |Pred(r)|_{\mathbf{A}} \cap |Pred(r)|_{\mathbf{A}}$;
$|Pred(r) \vee Pred(r')|_{\mathbf{A}} = |\Pr red(r)|_{\mathbf{A}} \cup |Pred(r)|_{\mathbf{A}}$;
$|\neg Pred(r)|_{\mathbf{A}} = U - |Pred(r)|_{\mathbf{A}}$

One more key notion in our study is dynamic rule. If we consider a family $\mathbf{F}$ of the subsets (subtables) of $\mathbf{A}$ ($\mathbf{F} \subset \mathbf{P}(\mathbf{A})$) then we will call rule $r \in \bigcup\limits_{\mathbf{B}\in\mathbf{F}} RUL(\mathbf{B})$ $\mathbf{F}$-*dynamic* (usually simply *dynamic*) if and only if:

$$|Pred(r)|_{\mathbf{B}} \neq \emptyset \Rightarrow r \in RUL(\mathbf{B}), \text{ for any } \mathbf{B} \in \mathbf{F} \tag{3}$$

In our further study we will rely on certain numerical characteristics of dynamic rules. One important is *stability coefficient* for the dynamic rule r relatively to $\mathbf{F}$ denoted by $SC_{\mathbf{A}}^{\mathbf{F}}$ and defined as follows:

$$SC_{\mathbf{A}}^{\mathbf{F}} = \frac{card(\{\mathbf{B} \in \mathbf{F} : r \in RUL(\mathbf{B})\})}{card(\{\mathbf{B} \in \mathbf{F} : |Pred(r)|_{\mathbf{B}} \neq \emptyset\})} \tag{4}$$

This coefficient reflects the frequency of occurrence of particular rule in the set of rules generated by subsequent steps of rule generation algorithm. The more frequent the rule (the higher $SC_{\mathbf{A}}^{\mathbf{F}}$) the better its reliability. For further details consult [4] and [3].
$SC_{\mathbf{A}}^{\mathbf{F}}$ in our study is used for resolving conflicts among rules. If for some object $o \in U$ exist two (or more) rules r_i, $r_j \in R$ such that r_i, $r_j \in Match_{\mathbf{A}}(R, o)$, where R is a set of rules, and $r_i$points at different decision class than r_j then *conflict* occurs. To make a decision we have to choose which rules should be trusted more in particular situation. To grade rules in view of conflict resolving we connect weights with groups of them. Those weights are given by:

$$W(B_i, o) = \frac{\sum\limits_{r \in Match(B_i, o)} Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P}(\mathbf{A})}(r)}{\sum\limits_{r \in B_i} Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P}(\mathbf{A})}(r)} \tag{5}$$

where B_i is the set of rules connected with i-th decision class ($i = 1, 2$), o is the object to be classified. In case when denominator in the above equals 0 the weight is set to 0 too. Instead of family $\mathbf{F}$ we use the entire $\mathbf{P}(\mathbf{A})$. Clearly, in the above formula, the value of weight depends on number of rules that match object to be classified and their stability given by the stability coefficient. The final decision is taken by comparing summarised weights for both decision classes ($W(B_1, o)$ and $W(B_2, o)$) and choosing the one that has higher value. We will identify B_1 as corresponding to class A (decision value 0) and B_2 as corresponding to class B (decision value 1).Of course this procedure is applicable in the case of more than two decision classes.

2.2 Attribute scaling

In case our attributes are mapping objects into potentially infinite (in our case real) domain we have to use some methods in order to avoid time-consuming computation. For that purpose several methods known as *discretisation, quantisation* or *scaling* were proposed by many researchers (see [1],[14]). In our work we rely on methods proposed in [3] and [14]. To give a reader some idea we are bringing here basic keywords with some explanation.

Let us consider an attribute $a_i : U \to V_a$, where V_a is an interval on the real axis. By *cut* c_i for the attribute a_i we mean any real number belonging to V_a . With the use of c_i we can exchange a_i with new binary attribute $\overline{a_i}$ defined by:

$$\overline{a_i}(o) = \begin{cases} 0 \text{ iff } a_i(o) < c_i \\ 1 \text{ otherwise} \end{cases} \tag{6}$$

The potential problem is how to search for cuts that will really help us in our task. The method we utilise is taking into account discernibility. The cut is selected if the attribute obtained by applying this cut allows to discern the highest number possible of not yet discerned object in the decision table. At each step one cut is chosen. By repeating this procedure until full discernibility is achieved, we select set of new binary attributes for further computations. In this way, we have binary decision table with the number of attributes usually smaller than in the original one. This method, although elegant, is usually too sensitive by the means of noise in data. Therefore, in our approach, we usually apply above algorithm with small modification allowing selection of several best cuts instead of one at each step. In our particular application it turned out that this number should be around 100.

Yet another method we use is dynamic scaling as proposed in [4]. The idea of this method has a lot in common with dynamic reduct and rule calculation.

Let us consider a gigantic decision table that is constructed from initial one by taking as attributes all possible cuts. Of course in the non-trivial case such a table is practically unmanageable since it has a number of elements proportional to the square of that of the initial table. But, interesting for us is the fact, that set of cuts constructed in a way presented in the previous paragraphs is a reduct in this huge table. This reduct, however, is not always the best for our needs. Therefore we extend the set of cuts by adding several, possibly redundant cuts (unnecessary with regard to discernibility) for the sake of better flexibility. The optimal criteria for adding particular cut are not identified yet. There may be several heuristics for that task, for details review [4], [14]. Having the widened set of cuts we perform dynamic reduct calculation using the attributes generated by them. The best dynamic reduct is chosen and the final scaling of decision table is determined by cuts belonging to this reduct. All this process in the terms of time complexity and result quality strongly rely on the number of redundant cuts we use. Therefore, it is sometimes necessary to make several trials in order to establish the extension of cut set that best suits our demands.

2.3 Neural Networks

It would be pointless to bring here all the basic definitions and algorithm descriptions concerning neural networks we use. Therefore, in this subsection we will only enlist major notions to be used and bring some conventions applied when speaking of artificial neural network models. For further reference consult [2],[15],[7]. Traditionally by $\vec{x} = (x_1, ..., x_n)$ we denote vector of inputs to our network. As we deal with binary decisions it is enough to consider only one output from our networks and we will refer it as y. Since our network has only one output, the error function in learning phase also has simple form. Depending on needs we have error of the form $(y - d)^2$ or $|y - d|$ where d is the required value of decision (network output) taken from decision table used for training (testing). We will also refer to the vector of neuron weights by using $\vec{w} = (w_0, ..., w_n)$, where w_0 represents the weight of bias (if exists).

The networks we use are, in most of the cases, very simple in layout. Majority of them is reduced to a single layer of neurons or even a single neuron. Neurons we use have either logistic sigmoid or hyperbolic tangent as their activation functions. The learning is performed with gradient based methods (modification of backpropagation). To improve learning abilities we use biased networks and train them using gradient descent method with regularisation and momentum factor. We also apply adaptive tuning of learning rate as described in [8]. In some experiments we also use network learning algorithms based on Lavenberg-Marquardt method (see [7]).

3 The WaRS method

The objective of this section is to present in detail a new and reliable method of preprocessing real-word signal data based on rapidly growing wavelets methods in combination with rough set approach founded by Z. Pawlak [19] and developed in the direction of approximate reasoning by A. Skowron (for survey see e.g. [16]). WaRS method (Wavelets+Rough Sets) gives a significant reduction of dimensionality of the problem, which is with biomedical signals very large, while keeping all the essential information allowing good classification. Rough set based classifiers built on the data yield only a few (about 10) rules which are easy to interpret and, what is particularly interesting, of very simple form of one-term expression e.g.

$$(a_7 = 0) \Rightarrow (d = 0)$$

where a_i is a scaled wavelet coefficient (see further text), $d = 0$ (decision) refers to endogenic case.

Then there is a question of the appropriate conflict solving when rules give contradictory results. Then we used with the best result the weight based on stability coefficient (4) see [4]. The experimental results of this paper showed however that application of adoptively learning ANN to solve this conflict is a very good solution showing both transparency of rough set methods and efficiency of classification tuned to the problem we consider.

Let us describe shortly details of WaRS method referring the interested reader to [25] for variants and extensions.

We construct a wavelet in a way presented in [17]. For a sequence of coefficients (c_n) we find a function satisfying scaling equation

$$\varphi\left(x\right) = \sum_{n \in Z} c_n \varphi\left(2x - n\right). \tag{7}$$

The coefficients (c_n) are chosen so as to get the condition

$$\sum_n c_n \overline{c_{n+2k}} = \delta_{k0} \tag{8}$$

This is equivalent to orthogonality of integer translates of φ: $\left\{\varphi\left(x - n\right)\right\}_{n \in Z}$. I. Daubechies in [5] proposed a method of (c_n) - sequence selection to guarantee that φ is of $C^{\alpha N}$ class and has a compact support. $(\alpha \approx 0.1)$. For analysis purposes we took the Daubechies' wavelet of 5th order i.e. c_n are non-zero only for $n = 0, .., 9$ and φ is of C^1 class. Figure 1 shows the example of scaling function φ for this case. For details of the construction see ([6] or [27]).

Let us define a function

$$\psi\left(x\right) = \sum_{n \in Z} \left(-1\right)^n c_{-n+1} \varphi\left(2x - n\right). \tag{9}$$

The graph of function ψ for φ given above is presented in figure 2.

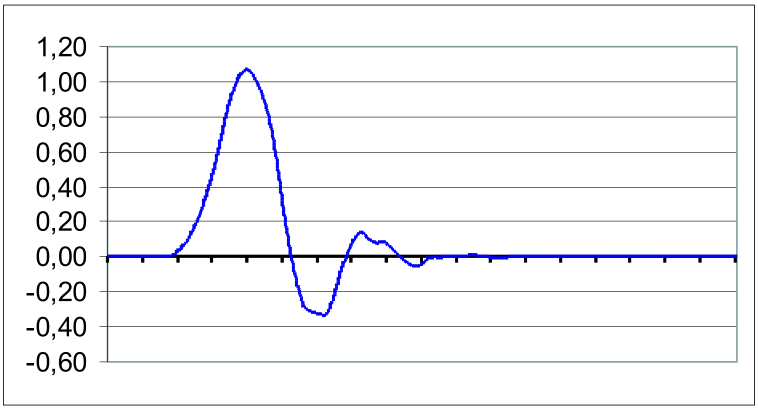

Fig. 1. Daubechies' scaling function of 5th order.

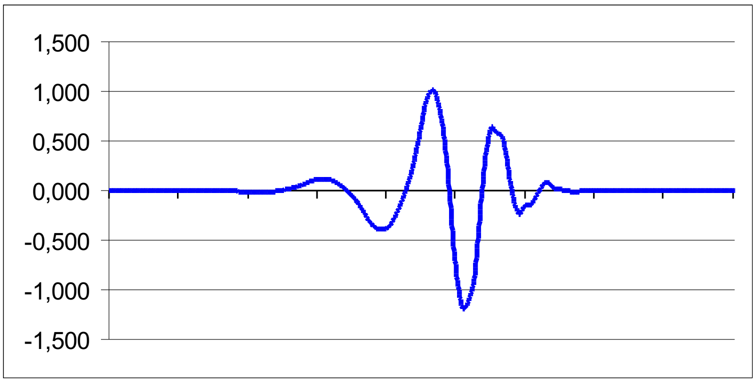

Fig. 2. Daubechies' wavelet of 5th order.

For the simple argument, that dyadic dilations and integer translations of this function $\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k)$ with $j, k \in Z$ are an orthonormal basis for $L^2(R)$, see [26]. It is equivalent to the following formula

$$f = \sum_{j,k \in Z} \langle f, \psi_{jk} \rangle \psi_{jk}$$

The main tool in processing of wavelet coefficients is hard thresholding procedure investigated thoroughly by D. Donoho, I. Johnstone et al. [10], [11], [12]. Let us consider the set

$$A_\vartheta = \left\{ (j,k) \in Z^2 : |\langle f, \psi_{jk} \rangle| \geq \vartheta \right\}$$

for given and non-negative ϑ. Then we sum up all coefficients with indices in A_ϑ:

$$\widetilde{f} = \sum_{(j,k) \in A_\vartheta} \langle f, \psi_{jk} \rangle \psi_{jk}.$$

The effectiveness of hard thresholding is in close relation with unconditionality of wavelet bases in many function spaces: Lebesgue spaces $L^p(R)$, Sobolev spaces and Besov-Triebel-Lizorkin spaces. The unconditionality is shown in

([18], [6], [27]). It was also proved that among all orthonormal bases unconditional bases are optimal for hard thresholding procedure [9].

We apply the following procedure of wavelet analysis of EEG data. We find the form of a wavelet ψ related to the sequence (c_n) by formulas (7-9). Then we compute the scalar products $\langle f, \psi_{jk} \rangle$. Simpson quadrature is irrelevant because of unconditionality, so we can use a standard integration procedure.

Wavelet analysis of the EEG signals resulted in decrease of the number of wavelet coefficients above the threshold $\vartheta = 1.0$

$$\frac{\# A_\vartheta}{\# A_0} \approx 0.05$$

in comparison to all coefficients about 20 times. The main point is that because of unconditionality

$$\widetilde{f} = \sum_{(j,k) \in A_\vartheta} \langle f, \psi_{jk} \rangle \, \psi_{jk}.$$

is very close to original f keeping much of visual information contained in EEG. In figures 3 and 4 we present the original signal and its wavelet coefficients after thresholding.

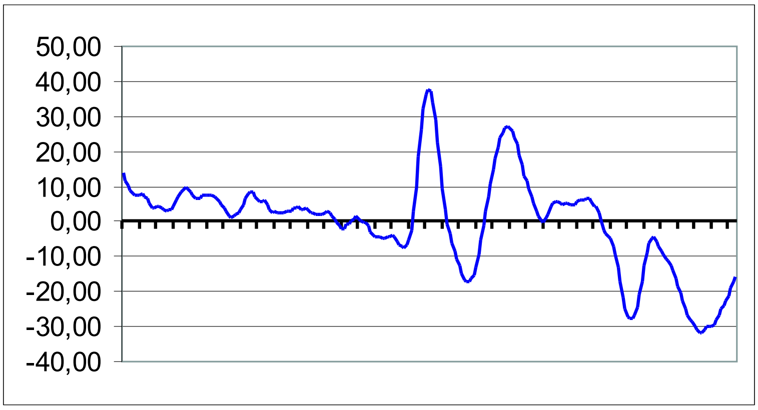

Fig. 3. Example of EEG score.

For application of discretisation procedures and other data mining techniques we need to choose as an attribute

$$(e, j, k) \longmapsto \langle f_e, \psi_{jk} \rangle$$

where f_e is the EEG signal from e-th electrode.

The straightforward procedure of taking all coefficients with absolute value bigger than $\vartheta = 1.0$ would result in 86×21 = 1806 attributes for every patient, which is too big number. Therefore, we have decided to apply *frequential analysis*.

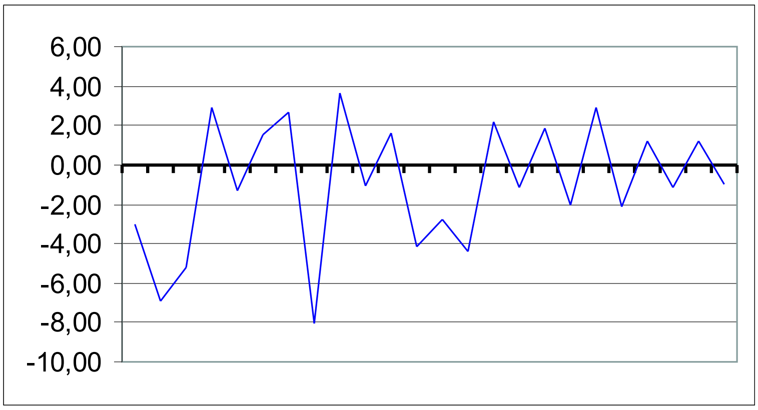

Fig. 4. Wavelet coefficients of EEG signal

For the frequential analysis we choose wavelet coefficients $\langle f, \psi_{jk} \rangle$ related to pairs (j,k) such that the number of their representatives greater than ϑ in all electrodes and for all patients is greater than or equal to the certain threshold M. In other words

$$F_M = \{(j,k) : \#\{(e,p) : |\langle f_{e,p}, \psi_{jk} \rangle| \geq \theta\} \geq M\}$$

where $f_{e,p}$ is the EEG signal from e-th electrode and for p-th patient.

By the appropriate choice of the threshold M we can consider wider or narrower group of attributes according to the needs of the actual approach. We scale the attributes from F_M using the best cuts approach on the level of 100 best cuts using Johnson's heuristics. For details of the method we defer the interested reader to [14], [13] ,[25]. This procedure of data preparation called by us 'WaRS method' was applied successfully in [25] and allowed in combination of rough sets (RSES library) and ANN to obtain a good classification of EEG signal.

4 Modified classifier

Although valid in some cases the straightforward constructed classifier (WaRS) has some features that, in our opinion, may show to be inconvenient or troublesome. In particular we have to be aware of the facts that:

(1) The amount of data (number of cases) as compared to the size of data (number of measurements) is relatively small and therefore some features of even a single object may significantly influence the overall performance.
(2) Presence of significant amount of noise in EEG data makes rules obtained by initial algorithm doubtful. According to the experiments their do not show necessary resistance to noised data and the preprocessing stage is

not designed in the way that eliminates every possible kind of distortion.

(3) The classifier uses rules and weights for them that are statically set during the process of construction. There is not known, universal method for making modifications to those weights without complete reconstruction of the classifier (i.e. rule and weight calculation).

(4) Weights that are given by stability coefficients are usually spanned over the wide range. In some cases the value of stability coefficient for a rule may be as small as $const \cdot 10^{-16}$ or as big as $const \cdot 10^{7}$. This forces us to use high numerical precision in our algorithms and makes those algorithms more complicated and costly.

(5) The process of decision making is not very straightforward. It comprises of checking an object against the rules and then establishing and comparing summarised weights. It requires three basic steps and in order to do so, we have to store the set of rules and weights.

Wanting to resolve in widest possible extent the issues mentioned above we decided to modify our initial approach.

To receive more significant and better justified by data classifier we, at the first step, decided to extend our set of data. Instead of the original EEG score we have used its noised copies. We called this part of process *noisification.*

Let us consider an EEG signal for p-th patient measured at e-th electrode $f_{e,p}(t)$. We add to it white noise being

$$f^{i}_{e,p}(t) = f_{e,p}(t) + \alpha\varepsilon_t$$

where ε_t is a random variable with a normal distribution $N(0, 1)$ independent for different t, while α describes the level of noise added. In the next stage we process these signal by WaRS method. The replacement of original signals with their 'noisified' copies gives the advantage that the classifier based on them does not rely on peculiarities of EEG signal that might have (and often have) a casual character. For the group A we used 11 copies of each EEG score and for B - 25. The additional point of this procedure is that the number of 'noisified' copies from each group is equal. We applied to main levels of noise: $\alpha = 0.1$ and $\alpha = 0.2$. Their influence can be seen mostly on high frequency wavelet coefficients (see figures 5 and 6 in comparison with figure 4).

The procedure specified above gives us larger set of objects and therefore, the regularities in data we search and exploit for classification purposes, may be better funded and verified.

To achieve higher level of confidence in the classifier validity and flexibility we decided to employ methods coming from the area of adaptive learning. Approach that seemed most promising was that of artificial neural networks

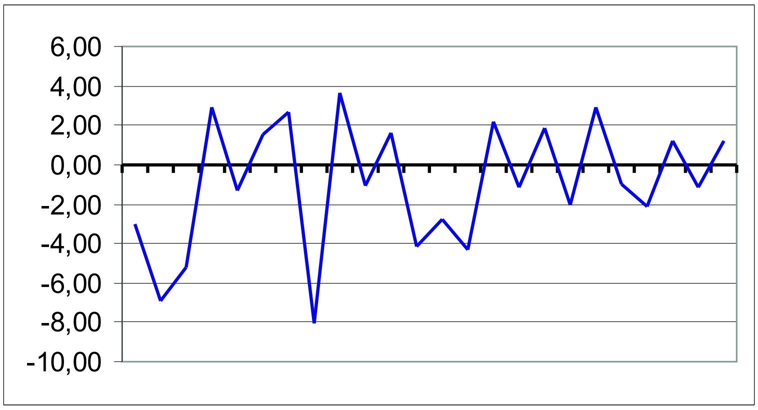

Fig. 5. Wavelet coefficients of noised EEG signal from figure 3 after thresholding. Noise level $\alpha = 0.1$

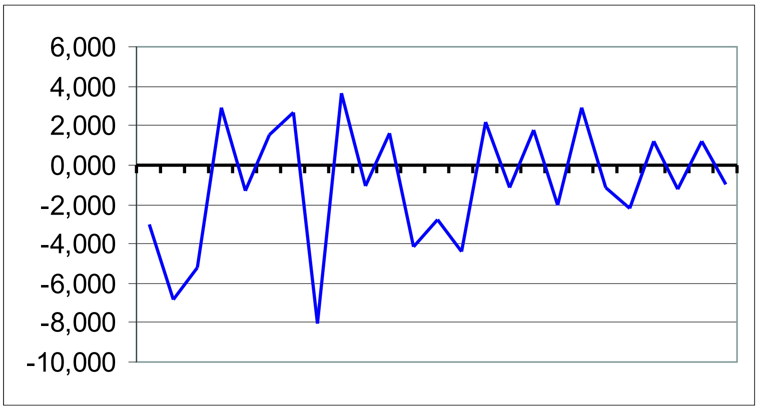

Fig. 6. Wavelet coefficients of noised EEG signal from figure 3 after thresholding. Noise level $\alpha = 0.2$

(ANN). Neural network based systems usually show good behaviour with regard to the noise and possibility of adding new data.

So, how do we want to include neural networks in our classification system?

In the first step we want to replace this part of initial classifier which is connected with weights. Instead of examining the output of all rules and summarising them using weights, we treat those outputs as an input vector $\vec{x}$ to a simple neural network.

Below we explain point by point how the modified classifier is being constructed (see figure 7):

(1) From original data (EEG) we extract features that are used for classification. To do that we use wavelet and frequential analysis coupled with scaling techniques. For classification we use a set of binary attributes (around 100).

(2) With the set of binary attributes we perform dynamic rule calculation. Derived rules are then significantly shortened. As an effect we receive small number (usually below 20) of rules having only one conditional term in the premise.

(3) With every object we connect a vector of values obtained by applying all the rules to that object. More precisely, for a given object o_i we introduce vector $\overrightarrow{ro(i)} = (ro(i)_1, ..., ro(i)_k)$, where k is the number of rules we have, according to the formula

$$ro(i)_j = \begin{cases} -1 \text{ if } \ r_j \text{ applies to } o_i \text{ and decision is } 0 \\ 0 \text{ if } r_j \text{ do not apply to } o_i \\ 1 \text{ if } \ r_j \text{ applies to } o_i \text{ and decision is } 1 \end{cases} \tag{10}$$

, where $r_1, ..., r_k$ are the rules.

(4) By applying the rules to all available objects we get a new training set for use with neural network. The network itself is very simple and contain only one neuron equipped with $k + 1$ inputs corresponding to rule output vector $\overrightarrow{ro(i)}$ and bias. Our neuron uses either sigmoid or hyperbolic tangent as its activation function. This network is further trained to recognise objects from set of examples (constructed in point 3). Gradient based methods like gradient descent with regularization and momentum or Lavenberg-Marquardt are used to perform neural network learning.

The layout
7

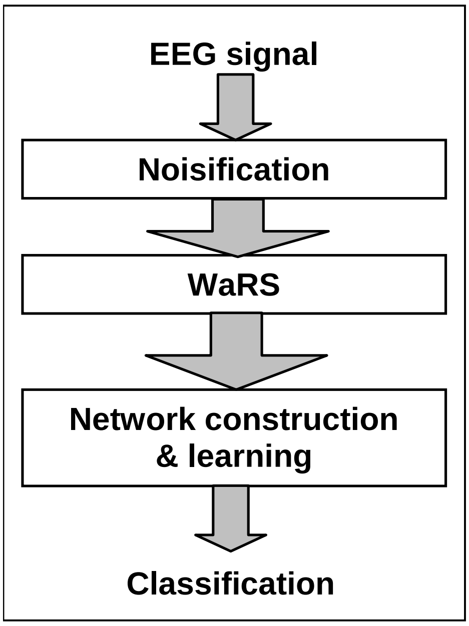

Fig. 7. The layout of proposed classification system.

The second box (entitled WaRS) from scheme presented in figure 7 corresponds to the application of methods from the domain of wavelet and frequential analysis coupled with rough set rule calculation as described in previous sections. General setup of this part is similar to the initial version of WaRS classifier but works with extended ('noisified') set of data. The only possible difference is that stability coefficients, although calculated, may be omitted in the further steps of construction.

Let us make the remark that with set of rules and single neuron we can always achieve at least the level of accuracy that is given by rule-based system using weights based on stability coefficients. We can just simulate the process of rule weighting. If we take a single, non-biased neuron that has threshold activation function φ such that:

$$\varphi(I) = \begin{cases} 0 \text{ iff } I \leq 0 \\ 1 \text{ otherwise} \end{cases} \tag{11}$$

and we connect its inputs with the attributes produced in point 3 of the procedure specified above then, by setting every $j-$th $(j = 1, ..., k)$ weight to be equal (according to (5) and previously introduced notation)

$$w_j = \frac{Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P(A)}}(r)}{\sum\limits_{r \in B_i} Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P(A)}}(r)} \tag{12}$$

where $i \in \{1, 2\}$, we got required effect. Thanks to the way we introduced the attributes $ro(.)_1, ..., ro(.)_k$ in (10) the actual output of the neuron can be represented as

$$y = \varphi\left(\frac{\sum\limits_{r \in B_2} Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P(A)}}(r)}{\sum\limits_{r \in B_2} Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P(A)}}(r)} - \frac{\sum\limits_{r \in B_1} Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P(A)}}(r)}{\sum\limits_{r \in B_1} Supp_{\mathbf{A}}(r) \cdot SC_{\mathbf{A}}^{\mathbf{P(A)}}(r)}\right) \tag{13}$$

which is exactly the formula for decision making in rough set rule-based system, just expressed in slightly different terms. In that way behaviour of rule and stability coefficient based classifier may be preserved.

It is necessary to mention some fundamental advantages of putting neural network into our classification scheme.

(1) Such a solution is more data driven and better reflects information which is in the data. It gives potential advantage over methodologies that rely on preset, statical rules of precedence not always adequate in the actual situation.
(2) There is a possibility of classifier modification without necessity of total reconstruction. New object, in case it was misclassified, is added as new learning example. As long as this new example is not completely different

from the previous ones and do not contradict all the knowledge we have learned so far, it is possible to add the knowledge it contains to our system by simply adjusting (learning) weights in neural network. Only in case, when new object is obviously contradictory to our prior findings we perform total reconstruction of classifier.

(3) Proposed solution allows more convenient and versatile representation of our classification system. It also opens the way for performing modifications to our system that go deeper and touch not only neuron learning but underlying rules and cuts too.

(4) Relatively small size of our classifiers' top layer allows to interpret them in the way that have real meaning. We may speak of influence that particular features have on our final decision, basing on the neuron weights. Since there is no more than teen of them it is quite transparent how they behave. Moreover, the features that we look at may be tracked back through the steps of frequential and wavelet analysis, so they can be connected with elements of actual EEG signal.

The modified version of classifier undergone several test to examine its accuracy and flexibility. Additional tests with noisified data were also performed in order to have a comparison with other methods as well as to find out which step in our modified procedure is most important. We wanted to check whether scaling, dynamic scaling, rule selection and rule weights may or may not be omitted in classification. We were also interested in verifying how manipulating some coefficients may improve or spoil clarity of our classifier. To do so, we constructed classifiers for not scaled data and without the use of rules. Below we discuss one by one the results on consecutive datasets.

The results we present undergone thorough verification. They were averaged over several repetitions of cross-validation test. Typically we used 5 fold cross-validation technique, so we had 80% of examples for learning and 20% for testing. In case of classifiers based on neural network we allowed output of our classifier to differ from the exact value. If the difference between classifier output and required value of decision was less than some preset value we regarded it as a correct one. This tolerance margin was decided during experimental evaluation and finally we applied the value 0.15. So, the outputs greater than 0.85 are treated as 1 and those less than 0.15, are considered being equal to 0.

First experiment was performed with the use of data received from wavelet and frequential analysis, so only part of WaRS method had been utilised. The data was not scaled. For this experiment we used 289 real-valued attributes that were chosen from 1806 wavelet coefficients at the stage of frequential analysis. To get the idea, how complicated is this data to learn, we made an attempt to construct a neural network that classifies it well. It turned out that simple, one neuron classifier is able to achieve only limited accuracy. The neuron uses biased, sigmoidal activation and is trained using gradient descent methods with adaptive learning rate and momentum. Since this result was quite accurate we checked it against generality. Unfortunately, the one-neuron solution showed major overfitting. It was basically impossible to add even a few new cases to trained network without loss of overall quality. An attempt to learn five new examples using previously trained network resulted in drop of accuracy by 25% on class A and 17% on class B. Therefore, we have also constructed more complicated network for the task. The smallest network that achieved good result had one hidden layer containing 55 biased, sigmoidal neurons and one neuron of the same kind in output layer. Network was fully connected, biased and trained with the Lavenberg-Marquardt (LM) version of backpropagation algorithm. We also made and experiment using the gradient descent learning method (GD) for comparison. In both cases the average accuracy was almost the same but GD method was slightly faster and less memory-consuming. Unfortunately in case of both simple and more complicated neural network we got no clue about the main mechanisms that drive the decision making. The total number of weights and their rather insignificant distribution does not allow identification of most important features in our data in the efficient way. Table 1 below shows accuracy achieved by these networks.

Table 1
Results of experiment with not scaled data.

Method	Training (A)	Training (B)	Testing (A)	Testing (B)
Single neuron	89%	90,2%	87,7%	90,1%
55-1 Network (LM)	93,1%	90,7%	92,8%	94%
55-1 Network (GD)	92,2%	91,4%	92,9%	92,4%

Next two experiments involved data that was scaled using 100 additional, redundant cuts. The data used for learning contained 105 binary attributes. The difference between those two runs was the amount of noise used in the process of data pre-processing with wavelets. Rough set rules and weights were computed. Application of rule shortening with factor between 0.3 and 0.6 gave us a set of rules with single conditional term. There were between 7 and 16 rules in consecutive runs of cross-validation test.

In the first of the experiments accuracy on training set for rough set (RSES) method (rules with weights) was 100%, but for testing not so. For the rules computed at this stage we constructed neuron that learned weights for those rules. The biased, sigmoidal neuron was learned using gradient descent methods with addition of momentum and adaptive learning rate selection. For comparison we also performed learning for a single neuron having architecture as above on the table before rule application (105 binary attributes). On the training set accuracy was 100%. On the testing set average accuracy was also very good. Unfortunately, the process of learning in later case was signifi-

Table 2
Results of experiment with scaled data and noise level $\alpha = 0.1$.

Method	Training (A)	Training (B)	Testing (A)	Testing (B)
RSES	100%	100%	57%	91,5%
Rules+neuron	92%	84%	91,4%	83%
Neuron (105 attr.)	100%	100%	99,8%	99,9%

Table 3
Results of experiment with scaled data and noise level $\alpha = 0.2$.

Method	Training (A)	Training (B)	Testing (A)	Testing (B)
RSES	100%	100%	71,4%	79,5%
Rules+neuron	90%	91%	89,8%	90,7%
Neuron (105 attr.)	100%	100%	99,5%	99,5%

cantly slower and the network obtained that way had no intuitive explanation. Table 2 summarises results of this experiment.

Next experiment was basically the same as previous one. The difference was the data that has been prepared with more noise added than in the previous case. The noisification coefficient was equal to $\alpha = 0.2$ ($\alpha = 0.1$ in the previous experiment). Distortion in data affected the whole process of classifier establishment. Although the number of binary attributes after scaling was the same (105), their setting was significantly different. This fact effected in extraction of a larger number of shortened, rough set rules. While in previous experiment (less noisified data) this number was between 7 and 13 (9 on the average), now this limit shifted up to 9-16 with average around 11. The results from rough set classifier and a single neuron trained on rule-based attributes are presented in the table 3 below. As in previous case neural network on data before rule application was tested for comparison purposes.

Last experiment followed the pattern of previous ones. The data preparation was performed with the setting $\alpha = 0.1$. The important difference was the way in which training and testing sets were constructed. Let us remind, that every object in the data table has, due to noisification procedure, several slightly modified versions. In this experiment once the object is randomly selected for testing set, all its modified (noisified) versions are automatically added to the same sample. Therefore no information relative to this object is present in the training set. As expected the results were significantly below the previous.

We gladly notice that experimental results have positively verified some of our expectations. One conclusion that comes from the results presented above is that we can produce quite simple classifier for basically complicated task. The classifier construction is a trade-off between complexity and accuracy. If we compare the accomplishments of a single neuron with and without augmenting the rules into the classifier, it is visible that in this later case it behaves better with regard to classification accuracy (difference up to 10%). This results comes mostly from higher flexibility and expressive power connected with larger number of inputs. The correlations that are present in the data (105 attributes) also contribute to this result. On the other hand, this more accurate classification system has rather limited meaning to us. The principles of classification are formulated using the language composed of 105 attribute-weight pairs not to mention bias. This is basically not what we looking for while constructing our classifier. There are no weights in the trained neuron that are prevailing or negligeable, so we get no clear suggestion about the importance of particular features.

The comparison with rough set based classifier using weights given by stability coefficients show also some interesting properties of our proposed system. In all of the experiments the rough set classifier (built over RSES library) achieved better (or equal) quality on training set. But the other approaches performed relatively better on testing data. There were much less cases of major disproportion between accuracy on class A and B. This is very valuable. Generally, proposed methods showed greater stability of results which suggest that they are able to describe more general and relevant properties that exist inside our information. It can be seen that adaptive selection of weights for rules result in better flexibility and generality of derived solution.

5 Conclusion

Ideas presented in the paper have been experimentally verified. From their application to areal problem of EEG analysis we got the following conclusions:

- Proposed methods can be efficiently applied to the classification of complicated EEG measurements. Results of experiments show, that those methods are capable to identify and manage important components of information that exist in our data. It is promising signal while considering application of so constructed system to other kinds of data.
- By application of neural networks connected with already investigated 'smart' techniques (like WaRS) one may improve flexibility and adaptiveness of classification procedure. That allows higher confidence in solution proposed.
- Application of wavelets, frequential analysis, scaling techniques and rough set methods effected in significant reduction in size of data that has to be considered while establishing decision. In this way, the whole process of decision support becomes less time-consuming and more transparent.

- In connection with previous point, let us notice that the features produced by the methods proposed are meaningful. Every particular attribute used in classification may be back-tracked to the wavelet coefficient it originated in. The values of those coefficients represent some characteristics appearing in the signal. Therefore, combinations of our binary attributes correspond to existence of particular shapes in the original EEG.
- Both, reduction in size and straight connection between signal and final decision making open the way for further investigation. Since only a fraction of information present in the EEG is necessary to achieve decent classification, it would be interesting to confront those findings with existing medical knowledge.

The possible extensions of this work are, as usual, numerous. As already mentioned, it would be most interesting to confront our approach with knowledge of medical experts. It would be also nice to have larger and more diversified data sets in order to find out how general the proposed methods really are. With feedback from medical community we may also consider some modifications and additions to the approach presented going in direction of their requirements. Those modifications may touch the core method as well as the way of result verification.

Acknowledgment

Authors want to thank the editors of this volume for invitation to publish this paper. Many thanks to our colleagues: Jan Bazan, Nguyen Sinh Hoa, Nguyen Hung Son and others for stimulating discussions and tools they allowed us to use.

Our work was supported by research programme of European Union ESPRIT project No. 20288 - CRIT 2. M. Szczuka received also support from Polish State Committee for Scientific Research grant 8T11C02412 and 8T11C02519.

References

[1] Agrawal R., Manilla H., Srikant R., Toivonen H., Verkamo I., Fast Discovery of Association Rules, In: Proceedings of the Advances in Knowledge Discovery and Data Mining, AAAI-Press/MIT Press, 1996, pp. 307-328

[2] Arbib M. A.(ed.), The Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge MA, 1995

[3] Bazan J., A Comparison of Dynamic and non-Dynamic Rough Set Methods for Extracting Laws from Decision Tables, In: Skowron A., Polkowski L.(ed.), Rough Sets in Knowledge Discovery 1, Physica Verlag, Heidelberg, 1998, pp. 321-365

[4] Bazan J., Approximate reasoning methods for synthesis of decision algorithms (in Polish), Ph. D. Thesis, Department of Math., Comp. Sci. and Mechanics, Warsaw University, Warsaw 1998

[5] Daubechies I.,Orthonormal bases of compactly supported wavelets, Comm. Pure Appl. Math., 41 (1988) p. 909-996

[6] Daubechies I., Ten lectures on wavelets, SIAM, Philadelphia 1992

[7] Hagan M. T., Demuth H. B., Beale M., Neural Network Design, PWS Publishing Company, Boston, 1996.

[8] Demuth H. B., Beale M., Neural Network Toolbox, The MathWorks Inc. Natick MA, 1997

[9] Donoho D. L., Unconditional bases are optimal bases for data compression and for statistical estimation, Applied and Computational Harmonic Analysis 1, 1993, pp.100-115

[10] Donoho D. L., Johnstone I., Neo-classical minimax problems, thresholding and adaptive function estimation, Bernoulli 2, 1996, no. 1, pp. 39–62.

[11] Donoho D. L., Johnstone I. M., Kerkyacharian G., Picard D., Density estimation by wavelet thresholding, Ann. Statist. 24, 1996, no. 2, pp. 508–539.

[12] Donoho, D. L. De-noising by soft-thresholding, IEEE Trans. Inform. Theory 41, no. 3, 1995, pp. 613–627

[13] Garey M., Johnson D., Computers and Intarctability: A Guide to the Theory of NP-completness, W.H. Freeman&Co., San Francisco, 1998, (twentieth print)

[14] Nguyen Sinh Hoa, Nguyen Hung Son, Discretization Methods in Data Mining, In: Skowron A., Polkowski L.(ed.), Rough Sets in Knowledge Discovery 1, Physica Verlag, Heidelberg, 1998, pp. 451-482

[15] Karayannis N. B., Venetsanopoulos A. N., Artificial Neural Networks: Learning algorithms, Performance Evaluation and Applications, Kluwer, Dordrecht, 1993

[16] Komorowski J., Pawlak Z., Polkowski L. and Skowron A., Rough sets: A tutorial. In: S.K. Pal and A. Skowron (eds.), Rough fuzzy hybridization: A new trend in decision–making, Springer-Verlag, Singapore 1999, pp. 3–98.

[17] Lawton W., Tight frames of compactly supported wavelets, J.Math.Phys. 31, 1990, pp. 1898-1901

[18] Meyer Y., Wavelets and Operators , Cambridge University Press, Cambridge, 1992

[19] Pawlak Z., Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer, Dordrecht, 1991

[20] Petrosian A., Prokhorov D., Homan R., Dasheiff R., Wunsch D., Recurrent Neural Network based Prediction of Epileptic Seizures in Intra- and Extracranial EEG, to appear in Neurocomputing, 1999

[21] Rauszer C., Skowron A., The Discernibility Matrices and Functions in Information Systems, In: Słowiński R. (ed.), Intelligent Decision Support, Kluwer, Dordrecht 1992.

[22] Rodin E., Litzinger M., Thompson J., Complexity of focal spikes suggests relative epileptogenicity, Epilepsia. 36(11), Nov.1995, pp.1078-83

[23] Senhadji L. , Dillenseger J. L., Wendling F., Rocha C., Kinie A., Wavelet analysis of EEG for three-dimensional mapping of epileptic events Annals of Biomedical Engineering. 23(5), Sep-Oct. 1995, pp.543-52

[24] Świniarski R.W.,Rough Sets and Principal Component Analysis and Their Applications in Data Model Building and Classification In: S.K. Pal and A. Skowron (eds.), Rough fuzzy hybridization: A new trend in decision–making, Springer-Verlag, Singapore 1999, pp. 275-300.

[25] Wojdyłło P., Wavelets, Rough Sets and Artificial Neural Networks in EEG Analysis, Proceeding of RSCTC'98, Lecture Notes in Artificial Intelligence 1424, Springer Verlag, Berlin, 1998, pp. 444-449

[26] Wojdyłło P., Wavelets and Mallat's Multiresolution Analysis, Fundamenta Informaticae 34, 1998, pp. 469-474

[27] Wojtaszczyk P., A Mathematical Introduction into Wavelets, London Society Students Mathematical Books 37, Cambridge University Press, Cambridge, 1997

[28] Wróblewski J., Covering with Reducts - A Fast Algorithm for Rule Generation, Proceeding of RSCTC'98, Lecture Notes in Artificial Intelligence 1424, Springer Verlag, Berlin, 1998, pp. 402-407

[29] Ziarko W., Variable Precision Rough Set Model, Journal of Computer and System Sciences, 40 (1993), pp. 39-59