University of Ballarat
Learn to succeed

COPYRIGHT NOTICE

UB ResearchOnline
http://researchonline.ballarat.edu.au

Detecting K-complexes for sleep stage identification using nonsmooth optimization

D. Moloney, N. Sukhorukova, P. Vamplew, J. Ugon,
G. Li, G. Beliakov, C. Philippe, H. Amiel, A. Ugon

Abstract

The process of sleep stage identification is a labour intensive task that involves the specialised interpretation of the polysomnographic signals captured from a patient's overnight sleep session. Automating this task has proven to be challenging for data mining algorithms because of noise, complexity and the extreme size of data. In this paper, we apply nonsmooth optimization to extract key features that lead to better accuracy. We develop a specific procedure for identifying K-complexes (special type of brain waves), which are crucial for distinguishing sleep stages. This procedure contains two steps. First, we extract "easily classified" K-complexes. Second, we apply nonsmooth optimization methods to extract features from the remaining data and refine the results from the first step. Numerical experiments show that this procedure is efficient for detecting K-complexes. It is also found that most of the classification methods perform significantly better on the extracted features.

Keywords: K-complexes, nonsmooth optimization, classification

1 Introduction

In this paper we focus on the detection of K-complexes, a special type of brain wave characterised by a sharp sudden increase in the wave amplitude [6]. Brain activity monitoring is part of PSG (polysomnography) a standard method used as a diagnostic tool in sleep medicine. Among the first to study the dynamics of the brain during sleep was M. Steriade [18], who remains an active researcher in the area [9]. K-complexes are defined by standardised scoring rules [16], based on the visual appearance of the signal. The manual scoring of K-complexes is very time consuming (due to the extreme size of data and noise) and subjective (due to variation of human perception). An accurate method for automatic detection of K-complexes would be very beneficial. In this paper we propose an new algorithm for detecting K-complexes. This procedure is based on an optimization model.

The automated detection of K-complexes is a challenging problem. A few algorithms have been proposed and reported in the literature. They are based on different techniques, including artificial neural network [3], continuous density hidden Markov model [11], wavelet transform [19] and matching pursuit approach [12]. However, medical doctors still report that the identification accuracy is not satisfactory [10, 15].

In this study, we approximate the brain signal by a wave with piecewise linear amplitude. This allows us to create an accurate model for the wave shapes (modelling patterns) and extract relevant characteristics of the waves (feature extraction). We minimise the deviation between the data and the modelling patterns. In this application the sum of absolute deviations is preferable to least square approximation since the data are very noisy [17]. However, this approach leads to the necessity of solving nonconvex and nonsmooth optimization problems. This is the main difference between our approach and the existing studies: the existing approaches are based on smooth functions, which are easier to work with (more approaches have been developed), but not so appropriate for K-complex detection (see section 4 for more information). After feature extraction, we apply classification algorithms over the obtained set of features, reducing the dimension of the corresponding classification problems and enhancing their classification accuracy.

The paper is organised as follows. We provide more information about the application of this work in section 2. In section 3 we provide the description of the data we use. In section 4 we provide a detailed description of the proposed procedure for K-complex detection. Section 5 contains the results of numerical experiments. Finally, in section 6 we provide the conclusions and future research directions.

2 Motivations

PSG generally consists of monitoring the patient's airflow (through both the nose and mouth), blood pressure, electrocardiographic activity, blood oxygen level, brain wave pattern, eye movement, and the movement of respiratory muscle and limbs. The PSG methods are used to help to diagnose and evaluate a number of sleep disorders. One such disorder is sleep apnoea disorder, when patients stop breathing during sleep. This causes so-called micro-arousals when the patient wakes up unconsciously for a brief period. These frequent awakenings are so short that the patient would not even remember them, however they completely destroy normal sleep patterns, leaving patients sleepy throughout the day. This increases chances of traffic and work accidents and may lead to other complications, including diabetes and cardiovascular problems.

PSG for sleep apnoea disorder diagnostics requires an overnight stay in a sleep clinic where the patient is monitored in a number of ways. The usual time taken is about 10 hours. Several electrodes are placed on the patient's body which include the chin, scalp, and the outer edge of eyelids. These electrodes must remain in place while sleeping because signals from electrodes are recorded while the patient is awake, with their eyes closed and during sleep. The PSG monitors many body functions including brain

activity (EEG), eye movements (EOG), muscle activity or skeletal muscle activation (EMG) and heart rhythm (ECG) during sleep. All together there are around 16-20 channels to monitor. The frequency of recording varies among channels, from 10 to 200 recordings per second. The EEG signal is recorded at a frequency of 100 readings per second. Therefore, the amount of data collected from one patient is very large and take a long time to process manually. Therefore an accurate automatic procedure for PSG analysis would be very beneficial.

The conventional approach for PSG analysis starts with sleep stage identification. There are five sleep states: awake, sleep stages one, two, three and REM (Rapid Eye Movement). Currently, doctors use a set of pre-defined rules to allocate each 30 second frame (epoch) to one of the five main sleep states. The application of these rules is performed either manually, by visual inspection of the PSG signal, or, more recently, by a software implementation of these rules on a computer. The allocation of each epoch to a sleep stage is performed using the set of standardized scoring rules, R&K (Rechtschaffen and Kales) rules [16]. One of the major shortcomings of these rules is the use of arbitrarily-defined thresholds to separate the sleep stages. This subjective assessment can lead to unreliable results and poor agreement between scorers. Another drawback arises from the fact that the rules were developed at a time when sleep staging was performed manually. This restricts the scoring to a small number of sleep stages and the use of fairly large epoch lengths, showing that scoring will be prohibitively time consuming. With all that said, R&K still remains a useful and popular clinical tool for some applications.

Another important study done in this area is "The AASM manual for the scoring of sleep and associated events" [10]. The goal of this study was to create a manual that reflected the current knowledge and would provide a more comprehensive standardised specification and scoring rules for characterising natural sleep as commonly performed in PSG. These rules and specification in the visual scoring of sleep retain much of the framework of R&K with some new definitions and rule modifications. These new rules provide a better method of analyzing data and are crafted as a platform to support the evolution of both manual (visual) and automatic (non-visual) methods for the future.

3 Data

One of the problems in the development of efficient methods for K-complex identification is data availability. There is no test dataset freely available for this type of research. This is partially because the scoring of K-complexes is normally not kept in PSG analysis records (it is only necessary to score the corresponding sleep stage). Therefore, it is either expensive (a qualified

doctor has to be involved to do extra time-consuming work) or inaccurate (a fellow researcher is performing the manual scoring after a short PSG analysis course) to create such kind of datasets. Another problem is that the same segment of data may be scored differently by two qualified scorers (see [10]).

In our experiments we use data scored at Tenon Hospital in Paris (France). On these data, it is more efficient to divide 30 second epochs into 3 parts (10 seconds each) and check the presence of K-complexes in each subepoch separately.

In this study, each observation is a 10 second segment of EEG recordings at the frequency of 100 recordings per second (1000 recordings which form 1000 features). As a training set we use a dataset with 28 non-K-complexes and 31 K-complexes (59 observations), as a test set we use a dataset with 38 non-K-complexes and 35 K-complexes (73 observations). Each signal consists of a sequence $(t_i, y_i), i = 1 \ldots 1000$, where $y_i, i = 1, \ldots, 1000$ are the EEG readings recorded at the moment t_i, and the $y_i, i = 1 \ldots 1000$ are the features of our observations.

In order to reduce the dimensionality of the data, we apply a special nonsmooth optimization based preprocessing. This preprocessing allows us to reduce the number of features from 1000 to 13 (see section 4 for details).

4 Approach

A K-complex is a brief negative high-voltage peak, followed by a slower positive one. An example of an idealised K-complex can be found in Wikipedia [22]. One of the main characteristics of K-complexes is an abrupt increase in amplitude. In real life PSG, the shape of K-complexes is not so clear. An example of a K-complex from our dataset is presented in Figure 1. The K-complex is located at the beginning of the data segment, its amplitude is considerably higher than it is in the rest of the segment.

4.1 Nonsmooth optimization and data extraction

The application of nonsmooth optimization to K-complex detection is based on minimising the deviation error (sum of absolute deviations) between the actual EEG curve and the modelled wave patterns. This approach is a proven method to extract wave characteristics so that we can obtain the targeted wave shape description. These characteristics can be used for the explicit description of wave shape patterns, so that we can obtain considerably lower dimensions for the corresponding classification problems.

The EEG curve is modelled as the sum of two sinusoidal curves. The amplitude of each curve is modelled as a piece-wise linear function (linear spline). This approach allows one to obtain more precise curve patterns than in the case of classical sine curves where the amplitude is a constant. Before we proceed, it is necessary to introduce the following definitions.
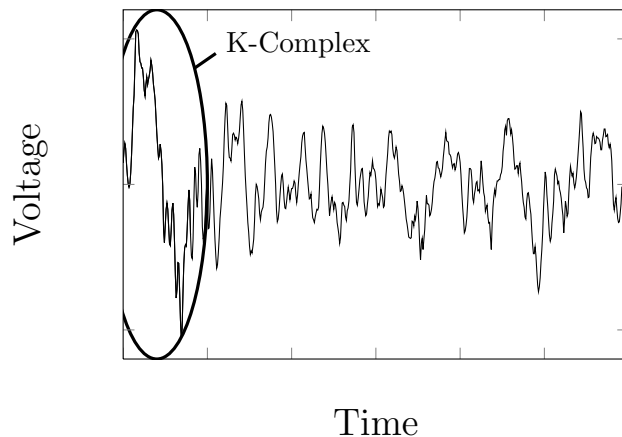
Figure 1: K-complex in practice

Polynomial splines are piece-wise polynomial functions [14]. In most applications polynomial splines are continuous. The points where the corresponding polynomials are joined together are called *spline knots*. Spline knots may be fixed or free. The highest degree of the corresponding polynomials is called the *spline degree*.

In this study we model the amplitude of the signal as a continuous spline function rather than a constant value. This approach is more flexible since it allows the amplitude to vary through the 10-second interval. Consider an example of polynomial spline construction (see [14]).

$$S_m(A, \theta, t) := a_0 + \sum_{j=1}^m a_{1j} t^j + \sum_{i=2}^n \sum_{j=1}^m a_{ij} ((t - \theta_{i-1})_+)^j, \qquad (4.1)$$

where m is the spline degree, $\theta = (\theta_1, \ldots, \theta_n)$ are the spline knots (in this paper we use either 1, 2 or 3 knots),

$$(\xi(x))_+ := \left\{ \begin{array}{ll} \xi(x), & \xi(x) > 0, \\ 0, & \xi(x) \leq 0, \end{array} \right.$$

$A = (a_0, a_{11}, \ldots, a_{nm}) \in \mathbb{R}^{mn+1}$ is called the vector of spline parameters. The knots may be fixed or free.

In order to use the polynomial splines for the amplitude one needs to know the spline parameters and knots. These can be obtained as a solution to the following optimization problem:

$$\min_{X, \omega, \tau} \sum_{i=1}^N |y_i - Amp(X, t_i) \sin(2\pi\omega t_i + \tau)|, \qquad (4.2)$$

where $y_i, i = 1, \ldots, N$ are the EEG readings recorded at the moment t_i, $Amp(X, t)$ is the amplitude function, modelled as a polynomial spline (see

5

the specifications below for the knots and spline parameters) ω is the frequency, τ is the curve shift which does not change the wave pattern, X is a vector which characterizes the amplitude function.

The optimization problem (4.2) is nonsmooth. Nonsmooth optimization problems are generally difficult and time consuming to solve. One of the ways to avoid nonsmoothness is to use the least squares method rather than the sum of the absolute deviations in (4.2). However, the least square method is not very robust when the corresponding data contain many outliers [17]. Since this is the case for EEG data, the least squares method may not be a suitable approach to extract key characteristics of brain waves and therefore nonsmooth optimization models are more suitable for this problem. In this study we propose a new nonsmooth optimization based approach (see subsection 4.1 for details) for detecting K-complexes.

For this paper we propose the following models for the amplitude function Amp.

- Linear spline with fixed knots, two intervals:

$$Amp = Amp_1(X,t) = a_0 + a_1t + a_2(t - \theta_1)_+.$$

  The knot θ_1 corresponds to the centre of a 10-second interval (5 seconds from the start of each subepoch), $X = (a_0, a_1, a_2)$.

  This model for the amplitude function is suitable for subepochs where the behaviour of the actual amplitude is changing approximately in the middle of the subepoch. For example, if it is growing in the first half of the subepoch and then stays relatively constant, one can expect $a_1 > 0$ and $a_2 = -a_1$. This model is also suitable for the subepochs where the behaviour of the actual amplitude is not changing. For example, if it is decreasing through the whole subepoch, one can expect $a_1 < 0$ and $a_2 = 0$. The dimension of the corresponding optimization problem is 5.

- Linear spline with free knots, two intervals:

$$Amp = Amp_2(X,t) = a_0 + a_1t + a_2(t - \theta_1)_+.$$

  Amp_2 is similar to Amp_1, but θ_1 is considered as an additional variable in the optimization problem (4.2) and therefore $X = (a_0, a_1, a_2, \theta_1)$. Amp_2 is more flexible than Amp_1, but the associated optimization problem is more complicated.

  This model is suitable for all the cases described above, but the change of the amplitude function may occur at any point inside subepochs, not necessarily in the middle. The dimension of the corresponding optimization problem is 6.

- Sharp peak amplitude:

$$Amp = Amp_3(X,t) = \max\{a_0, -a_1|t_i - a_2| + a_3\}, a_j \geq 0,$$

6

$$j = 0, \ldots, 3, \quad a_2 \leq 10.$$

This model is suitable for subepochs where the amplitude is relatively constant ($a_1 = 0$), or there is a sudden sharp increase of the amplitude inside the subepoch while on the rest of the epoch the amplitude stays the same. The corresponding $X = (a_0, a_1, a_2, a_3)$. The dimension of the corresponding optimization problem is 6.

This amplitude function is a special case of a linear spline with some restrictions on spline parameters and can be written in the form of (4.1).

In our experiments, we solve optimization problem (4.2) using the GANSO library [5, 20]. GANSO is a programming library which implements several methods of global, nonsmooth, nonconvex and nonlinear optimization. In practice, there is no optimization method which is able to find a global minimum to any function, in particular if this function is nonconvex and nonsmooth. In (4.2) the objective function is nonsmooth and nonconvex and therefore we can only claim that the obtained solutions are locally optimal (optimal in a certain neighbourhood). Therefore, in practice, a solution to a simpler problem may be more precise than a solution to a more complicated problem, which, in theory, describes the behaviour of the amplitude much better. This is especially critical when the dimension of the corresponding optimization problem is increasing.

Our numerical experiments with Amp_1, Amp_2 and Amp_3 show that Amp_1 is preferable. Amp_1 works faster than Amp_2 and the knot θ_1 for almost all the subepochs (for Amp_2) was placed at the beginning or the end of the epoch and therefore was not useful. Amp_3 failed to find any sudden increase in the amplitude. Therefore, in this paper we only present the results for Amp_1. For other datasets the preferences may be different.

Remark 1 *We also modelled the signal frequency as a polynomial function, since it decreases significantly on K-complexes (see Figure 1). However, this significantly increased the dimension of the corresponding optimization problem without any improvement.*

We model the waveform as a sum of two waves

$$W = W_1 + W_2 = Amp(X_1, t_i)\sin(2\pi\omega_1 t_i + \tau_1) + Amp(X_2, t_i)\sin(2\pi\omega_2 t_i + \tau_2). \tag{4.3}$$

Therefore, the corresponding optimization problem is

$$\min_{X, \omega, \tau} \sum_{i=1}^{N} |f(t_i) - Amp(X_1, t_i)\sin(2\pi\omega_1 t_i + \tau_1) - Amp(X_2, t_i)\sin(2\pi\omega_2 t_i + \tau_2)|. \tag{4.4}$$

The dimension of this problem is 10. In our experiments, we used the following two stage procedure to reduce the dimension.

In the first step, we minimise the error (exactly as it is in (4.2)):

$$\min_{X,\omega,\tau} \sum_{i=1}^{N} |y_i - Amp(X,t_i)\sin(2\pi\omega t_i + \tau)|. \tag{4.5}$$

Suppose that $(X_1^*, \omega_1^*, \tau_1^*)$ is the solution obtained at the first step (optimization problem (4.5)). Then the resulting wave is

$$W_1 = Amp(X_1^*, t_i)\sin(2\pi\omega_1^* t_i + \tau_1^*).$$

In the second step, we minimise the error over a new dataset, where the original data were substituted with the difference between the original data and the wave obtained before:

$$\min_{X,\omega,\tau} \sum_{i=1}^{N} |y_i - W_1 - Amp(X,t_i)\sin(2\pi\omega t_i + \tau)|. \tag{4.6}$$

Now suppose that (X_2, ω_2, τ_2) is the solution obtained at the second step (optimization problem (4.6)). Then the resulting wave is

$$W_2 = Amp(X_2, t_i)\sin(2\pi\omega_2 t_i + \tau_2).$$

Therefore, we are extracting two trends, and the sum $W = W_1 + W_2$ is the waveform we are looking for.

One can see that (4.6) is equivalent to the following problem:

$$\min_{X,\omega,\tau} \sum_{i=1}^{N} |y_i - Amp(X_1^*, t_i)\sin(2\pi\omega_1^* t_i + \tau_1^*) - Amp(X,t_i)\sin(2\pi\omega t_i + \tau)|. \tag{4.7}$$

Comparing (4.4) and (4.7), one can see that the problems are not equivalent. However, the two-step approach requires solving two smaller optimization problems (the corresponding dimension of each problem is 5) and therefore we use this approach in our numerical experiments.

The parameters of the extracted trend form the nonsmooth optimization output. In the case of Amp_1 the size of the output is 10 (twice the dimension of the corresponding optimization problem). Also, we take into account three more parameters which characterise the improvement of the objective function after nonsmooth optimization. Therefore, 1000 features of the original data have been replaced by 13 essential features.

In Figure 2 one can see an example of the approximation curve and the original PSG pattern (EEG). This is the same data segment as the one from Figure 1. One can see that the approximation curve's amplitude is considerably higher at the beginning of the segment (where the K-complex is located). Although the approximation does not follow precisely the original trend, it is close enough to detect the K-complex and therefore to produce the correct classification result.
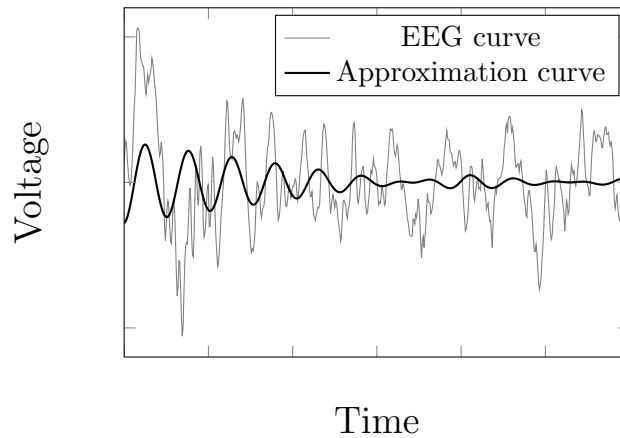
8

Time

Figure 2: Approximation example: the amplitude of the approximation curve is considerably higher at the beginning of the data segment (where the K-complex is located)

4.2 Weka and data classification methods

Weka is a collection of machine learning algorithms for data mining tasks (see [8]). Weka supports several standard data mining tasks, including data preprocessing, clustering, classification, regression, visualization and feature selection.

We use the following classification methods from Weka, applying them before and after data preprocessing through nonsmooth optimization:

- LibSVM - an integrated software for support vector machines (SVM) classification;

- Logistic - a generalised linear model used for binomial regression;

- MLP - a classifier that uses backpropagation to classify instances, also known as MultilayerPerceptron (a special type of neural network);

- RBF - a classifier that implements a normalised Gaussian radial basis function network, using the K-means clustering algorithm to provide the basis functions;

- SMO - a sequential minimal optimization algorithm for training a support vector classifier (a special case of LibSVM);

- Lazy IBK - a K-nearest neighbours classifier (uses normalised Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance);

- J48 and J48graft - classifiers based on C4.5 decision tree;

9

- LMT - a logistic model tree based approach, with logistic regression functions at the leaves;

- Random Tree - a classifier for constructing a tree that considers K random features at each node (no pruning).

- Random Forest - a classifier which consists of a collection of tree structured classifiers (see Random Tree for constructing trees);

All these methods were used with default parameters, except Lazy IBK, which was used with $K = 1, 5$ and 10. Weka is open source software, its web-site [21] provides all the necessary documentation. Therefore, we only provide a very short description of the classification methods used in this research.

5 Numerical experiments

First, we try classification methods from Weka without any application of nonsmooth optimization. The results are presented in Table 1.

From this table one can see two very important outcomes. First of all, RBF produces very good results. Second, MLP has not produces any result (software crash after running out of memory). This is most probably due to the large size of the corresponding data (1000 features).

The next step is to apply all the above classification methods after nonsmooth optimization based preprocessing. In this study we use optimization methods from the GANSO library [20]. The description to all these methods (ECAM, DFBM, DFBMECAM, DSO) can be found at the GANSO web-site [20]. Now we present a brief introduction to these optimization methods.

- Extended Cutting Angle Method (ECAM), [2, 4]. This method is based on the fact that under Lipschitz continuity (our objective functions satisfy this property), it is possible to estimate the smallest possible minimum of the objective function from its recorded values at various points. By using a large number of points, it is possible to approximate the original function closely enough by its underestimate, and then use the global minimum of the underestimate to approximate that of the original function.

  Lipschitz continuous functions are restricted in how fast they can change. Namely,
  $$|f(x) - f(y)| \leq Ld(x - y), \tag{5.1}$$
  where x and y are two points in the feasible region of f, L is the Lipschitz constant, d is the distance between x and y (for example, Euclidean).

- Derivative-Free Bundle Method (DFBM), [1]. An essential step of this method is to estimate the direction of descent using some information about the subdifferential, a generalisetion of the gradient to the case of nonsmooth functions. After obtaining a descent direction, the algorithm performs a line search along this direction. While the DFBM is a local method (it converges to a locally optimal solution, from any starting point), the fact that it uses an approximation to the subdifferential, allows it to converge to a sufficiently "deep" local minimum in multiextremal problems. This is an advantage of this method over other competing approaches that converge to the nearest local minimum.

- DFBMECAM. This is a combination of the DFBM with a version of ECAM, designed to improve line search used in DFBM, as well as to facilitate leaving shallow local minima.

- Dynamical Systems - Based Optimization (DSO), [13]. This method is based on the construction of a dynamical system using a number of values of the objective function, and associating certain "forces" with these data. The evolution of such a system yields a globalized descent trajectory, leading to a lower value of the objective function.

ECAM0.1 and ECAM0.001 are modifications of ECAM with the Lipschitz constant equal to 0.1 and 0.001, while ECAM itself takes the Lipschitz constant 1. ECAM modifications with a lower value for the Lipschitz constant work faster, but are not as precise as higher value modifications. The classification methods from Weka have been trained on the preprocessed training set and the classification accuracy is the test accuracy obtained on the preprocessed test set. Table 2 contains these results.

The classification of one subepoch by any method from Weka takes less than 30 seconds, while nonsmooth optimization preprocessing may take around 1 minute (DFBM, DFBMECAM, DSO) or even 10 minutes (ECAM, ECAM0.1, ECAM0.001).

From Table 2 one can see that even though the best results are not as good as the results obtained on the original dataset by RBF, they are still interesting, because

- none of the classification methods failed when performed on the preprocessed data;

- the accuracy of all the classification methods from Weka (except RBF and Random Forest) has been considerably improved after nonsmooth optimization based preprocessing.

Method	Test set accuracy
LibSVM	47.95%
Logistic	39.73%
MLP	OUT OF MEMORY
RBF	73.97%
SMO	43.84%
Lazy IB1	56.16%
Lazy IB5	49.32%
Lazy IB10	47.95%
J48	56.16%
J48graft	53.42%
LMT	45.21%
Random Forest	63.01%
Random Tree	53.42%

Table 1: Original datasets, 1000 features.

Method	ECAM0.001	ECAM0.1	ECAM	DFBM	DFBMECAM	DSO
LibSVM	52.54%	62.71%	59.32%	54.24%	54.24%	54.24%
Logistic	54.23%	54.23%	55.93%	59.32%	66.01%	55.93%
MLP	57.62%	45.76%	55.93%	59.32%	57.63%	61.02%
RBF	62.71%	44.02%	61.02%	61.02%	57.63%	55.93%
SMO	45.76%	50.85%	64.41%	62.71%	55.93%	47.46%
Lazy IB1	62.72%	50.85%	54.24%	40.68%	61.02%	64.41%
Lazy IB5	64.41%	62.72%	62.72%	50.85%	64.41%	62.72%
Lazy IB10	54.24%	54.24%	61.02%	59.32%	49.15%	59.32%
J48	49.15%	54.24%	50.85%	55.93%	55.93%	55.93%
J48graft	49.16%	54.24%	50.85%	55.93%	55.93%	55.93%
LMT	47.46%	55.93%	62.71%	61.02%	62.72%	45.77%
Rand. Forest	62.72%	55.93%	57.62%	55.93%	55.93%	55.93%
Rand. Tree	62.71%	54.24%	54.24%	44.07%	59.32%	66.10%

Table 2: Classification results after nonsmooth optimization preprocessing

The second observation is especially important, because if the parameters for RBF are not known (in our case the default parameters were suitable) the accuracy after preprocessing may be better than without preprocessing even for RBF. Also, in most cases, this observation is independent of the optimization method applied and therefore even a fast and not very precise method (DSO, DFBM, DFBMECAM) can be used in preprocessing. Overall, one can see that in general faster optimization methods (DSO, DFBM, DFBMECAM) work quite well and very often outperformed the slow and precise methods ECAM, ECAM0.1, ECAM0.001.

The best classification results have been obtained for the combination of DSO and Random Tree (66.1%) and the combination of DFBMECAM and Logistic (66.01%). Also, one can notice that some of the Weka methods perform better after certain optimization methods (Random Tree performs well after DSO or ECAM0.001), while others (Logistic, Lazy IB5, J48 and J48graft) are mostly independent of the chosen optimization methods.

Consider the results obtained by RBF without preprocessing (the best accuracy, Table 1) in detail. The test accuracy is 74%. 34 out of 38 non-K-complexes have been classified correctly, but only 20 out of 35 K-complexes have been classified correctly. The corresponding confusion matrix is

$$\left(\begin{array}{cc} 34 & 4 \\ 15 & 20 \end{array} \right).$$

In our confusion matrices element {11} corresponds to the number of correctly classified non-K-complexes, element {22} corresponds to the number of correctly classified K-complexes, element {12} corresponds to the number of false positives and element {21} corresponds to the number of false negatives.

Our ultimate goal is to create an automatic technique for fast and accurate detection of K-complexes. Therefore, such a high number of false negatives has to be improved since it is much better to highlight "suspicious" segments of data for the doctor to accept or reject rather than omit them completely from further consideration. We propose the following algorithm to decrease such a high rate of false negatives.

First, we apply RBF to the original dataset and remove the segments which have been classified as K-complexes. Then the rest of the dataset is to be reclassified after nonsmooth optimization preprocessing. The advantages of this procedure are that

- the first step of the procedure does not have too many false positives (4 out of 38);

	ECAM0.001	ECAM0.1	ECAM
LibSVM	$\begin{pmatrix} 3 & 31 \\ 1 & 14 \end{pmatrix}$	$\begin{pmatrix} 20 & 14 \\ 8 & 7 \end{pmatrix}$	$\begin{pmatrix} 12 & 22 \\ 4 & 11 \end{pmatrix}$
RBF	$\begin{pmatrix} 22 & 12 \\ 8 & 7 \end{pmatrix}$	$\begin{pmatrix} 23 & 11 \\ 13 & 2 \end{pmatrix}$	$\begin{pmatrix} 23 & 11 \\ 9 & 6 \end{pmatrix}$
Lazy IB1	$\begin{pmatrix} 19 & 15 \\ 9 & 6 \end{pmatrix}$	$\begin{pmatrix} 24 & 10 \\ 11 & 4 \end{pmatrix}$	$\begin{pmatrix} 18 & 16 \\ 10 & 5 \end{pmatrix}$
Lazy IB5	$\begin{pmatrix} 16 & 18 \\ 10 & 5 \end{pmatrix}$	$\begin{pmatrix} 26 & 8 \\ 12 & 3 \end{pmatrix}$	$\begin{pmatrix} 22 & 12 \\ 8 & 7 \end{pmatrix}$
Logistic	$\begin{pmatrix} 21 & 13 \\ 12 & 3 \end{pmatrix}$	$\begin{pmatrix} 27 & 7 \\ 11 & 4 \end{pmatrix}$	$\begin{pmatrix} 29 & 5 \\ 13 & 2 \end{pmatrix}$
MLP	$\begin{pmatrix} 18 & 16 \\ 7 & 8 \end{pmatrix}$	$\begin{pmatrix} 29 & 5 \\ 10 & 5 \end{pmatrix}$	$\begin{pmatrix} 26 & 8 \\ 13 & 2 \end{pmatrix}$
	DFBM	DSO	DFBMECAM
LibSVM	$\begin{pmatrix} 6 & 28 \\ 4 & 11 \end{pmatrix}$	$\begin{pmatrix} 33 & 1 \\ 15 & 0 \end{pmatrix}$	$\begin{pmatrix} 33 & 1 \\ 15 & 0 \end{pmatrix}$
RBF	$\begin{pmatrix} 20 & 14 \\ 8 & 7 \end{pmatrix}$	$\begin{pmatrix} 30 & 4 \\ 15 & 0 \end{pmatrix}$	$\begin{pmatrix} 23 & 11 \\ 13 & 2 \end{pmatrix}$
Lazy IB1	$\begin{pmatrix} 16 & 18 \\ 10 & 5 \end{pmatrix}$	$\begin{pmatrix} 23 & 11 \\ 11 & 4 \end{pmatrix}$	$\begin{pmatrix} 19 & 15 \\ 7 & 8 \end{pmatrix}$
Lazy IB5	$\begin{pmatrix} 12 & 22 \\ 4 & 11 \end{pmatrix}$	$\begin{pmatrix} 21 & 13 \\ 8 & 7 \end{pmatrix}$	$\begin{pmatrix} 18 & 16 \\ 9 & 6 \end{pmatrix}$
Logistic	$\begin{pmatrix} 10 & 24 \\ 6 & 9 \end{pmatrix}$	$\begin{pmatrix} 28 & 6 \\ 12 & 3 \end{pmatrix}$	$\begin{pmatrix} 21 & 13 \\ 12 & 3 \end{pmatrix}$
MLP	$\begin{pmatrix} 13 & 21 \\ 6 & 9 \end{pmatrix}$	$\begin{pmatrix} 28 & 6 \\ 13 & 2 \end{pmatrix}$	$\begin{pmatrix} 26 & 8 \\ 11 & 4 \end{pmatrix}$

Table 3: Reducing the number of false negatives: confusion matrices

- the second step, which is time consuming due to nonsmooth optimization, takes only part of the original dataset (49 out of 73).

Table 3 contains the results of the proposed procedure.

One can see that this procedure does not improve the classification accuracy, but reduces the number of false negatives (see ECAM0.1 combined with MLP). Therefore, even though it has been already observed that the faster group of nonsmooth optimization methods performs well, the highest accuracy has been achieved on ECAM0.1, a method from the slow group.

6   Main conclusions and further research

6.1   Conclusions

In this study we propose a new procedure to detect K-complexes, short lasting waves which serve as key points for detecting sleep stage two. This procedure is based on nonsmooth optimization and classification methods from Weka. A combination of RBF (to detect "easy classified" K-complexes) and ECAM0.1 with MLP (for the removal of extra false negatives) produced the best classification results.

The proposed combination works well on the available data, however, the lack of freely available test datasets prevents us from testing this procedure on larger sets of data and comparing with other researchers' approaches.

The proposed approach has two main advantages. Firstly, the proposed nonsmooth optimization based preprocessing allows one to reduce the size of the classification problem. Secondly, the accuracy of all the classification methods from Weka (except RBF and Random Forest) has been considerably improved after the preprocessing.

6.2   Further research directions

Although our nonsmooth optimization based preprocessing approach performs very well on our datasets, we identify some future research directions for further improvement of the procedure.

One of the main problems of the proposed algorithm is that the nonsmooth optimization part is time-consuming. This issue has to be addressed before implementing our approach in an automatic procedure for K-complexes detection, which can be used by medical doctors. One way to achieve this is to develop a specific optimization method for this particular problem. Such a specific method will work faster and more efficiently than general purpose optimization algorithms used in this study. Our future studies will address this important issue.

References

[1] A. Bagirov. A method for minimization of quasidifferentiable functions. Optimization Methods and Software, 17:31-60, 2002.

[2] A. Bagirov and A. Rubinov, Modified versions of the cutting angle method, In N. Hadjisavvas and P.M. Pardalos, editors, Convex analysis and global optimization, volume 54 of Nonconvex optimization and its applications, pages 245-268. Kluwer, Dordrecht, 2001.

[3] I. N. Bankman, V.G. Sigillito, R.A. Wise, and P. L. Smith, Feature based detection of the K-complex wave in the human electroencephalogram using neural networks. IEEE Transaction on Biomedical Engineering, vol. 39, no. 12, pp. 1305-1310, December, 1992.

[4] G. Beliakov, The cutting angle method - a tool for constrained global optimization, Optimization Methods and Software, 19:137-151, 2004.

[5] G. Beliakov and J. Ugon, Implementation of novel methods of global and non-smooth optimization: GANSO programming library, Optimization, 56 (2007), pp. 543-546.

[6] Paul R. Carney, Richard B. Berry, James D. Geyer./ editors, Clinical sleep disorders Philadelphia, Pa. ; London : Lippincott Williams & Wilkins, 2005.

[7] V. F. Demyanov and A. M. Rubinov (eds.), Quasidifferentiability and related topics, Kluwer Academic Publisher, Dordrecht, 2000, 400 p.

[8] E. Frank, I. Witten, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann, June 2005, Burlington, USA, 525 pages.

[9] P. Fuentealba, I. Timofeev, M. Bazhenov, T.J. Sejnowski, M. Steriade, Membrane bistability in thalamic reticular neurons during spindle oscillations, J. Neurophysiol. 2005 Jan; 93(1):294-304. Epub 2004 Aug 25

[10] C. Iber, S. Ancoli-Israel, A. Chesson and S. Quan. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Technology and Technical Specifications, Westchester: American Academy of Sleep Medicine, 2007.

[11] A. Kam, A. Cohen, A. B. Geva, and A. Tarasiuk, Detection of K-complexes in Sleep EEG Using CD-HMM, 26th Annual International Conference of the IEEE EMBS, 1-33-6, USA, September 1-5, 2004

[12] U. Malinowska, P. J. Durka, K. J. Blinowska, W. Szelenberger, A. Wakarow, Micro- and Macrostructure of sleep EEG, IEEE Eng. Medicine and Biology Magazine, 25: 26-31, July/August 2006.

[13] M.A. Mammadov and R. Orsi, H-infinity synthesis via a nonsmooth, nonconvex optimization approach, Pacific Journal of Optimization, 1:405-420, 2005.

[14] G. Nürnberger, Approximation by spline functions, Springer-Verlag, Berlin Heidelberg, 1989.

[15] Agarwal Rajeev, Jean Gotman, *Digital tools in polysomnography*, Journal of clinical neurophysiology, Vol. 19, Issue 2, pp. 136-143, 2002.

[16] A. Rechtschaffen and A. Kales, A Manual of Standardized Terminology, Techniques, and Scoring System for Sleep Stages of Human Subjects. Washington, DC: US Public Health Service, US Government Printing Office, 1968.

[17] A. Rubinov, N. Sukhorukova, J. Ugon The choice of a similarity measure with respect to its sensitivity to outliers Dynamic of Continuous, Discrete & Impulsive Systems, 17.5 (2010), pp. 709–721

[18] M. Steriade, P. Wyzinski, G. Oakson, Activities in synaptic pathways between the motor cortex and ventrolateral thalamus underlying EEG spindle waves. Int J Neurol, 1971;8(2):211-29.

[19] Z. Tang and N. Ishii, Detection of K-complex using a new method of recognising waveform based on discrete wavelet transform, IEICE transation on information systems, vol. 78, no.1, pp 77-85, 1995.

[20] GANSO library: http://www.ballarat.edu.au/ard/itms/CIAO/ganso/

[21] Weka web-site: http://www.cs.waikato.ac.nz/ml/weka/

[22] Wikipedia: http://en.wikipedia.org/wiki/Sleep_spindle