ANZIAM J. 52(2011), 319–332
doi:10.1017/S1446181112000016

DETECTING *K*-COMPLEXES FOR SLEEP STAGE IDENTIFICATION USING NONSMOOTH OPTIMIZATION

D. MOLONEY[1], N. SUKHORUKOVA ✉[1], P. VAMPLEW[1], J. UGON[1], G. LI[2], G. BELIAKOV[2], C. PHILIPPE[3], H. AMIEL[3] and A. UGON[3]

(Received 23 February, 2011; revised 10 October, 2011)

Abstract

The process of sleep stage identification is a labour-intensive task that involves the specialized 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, a special type of brain wave crucial for distinguishing sleep stages. The procedure contains two steps. We first extract "easily classified" *K*-complexes, and then 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 classification methods perform significantly better on the extracted features.

2010 *Mathematics subject classification*: 90C90.

Keywords and phrases: *K*-complexes, nonsmooth optimization, classification.

1. Introduction

This paper focuses on the detection of *K*-complexes, a special type of brain wave characterized by a sharp sudden increase in the wave amplitude [6]. Brain activity monitoring is part of polysomnography (PSG), a standard method used as a diagnostic tool in sleep medicine. Among the first to study the dynamics of the brain during sleep was Mircea Steriade (1924–2006) [17], who remained an active researcher in the area as late as 2005 [8].

[1]University of Ballarat, P. O. Box 663, Ballarat, Victoria 3353, Australia;
e-mail: d.moloney@ballarat.edu.au, n.sukhorukova@ballarat.edu.au, p.vamplew@ballarat.edu.au, j.ugon@ballarat.edu.au.
[2]Deakin University, 221 Burwood Highway, Burwood, Victoria 3125, Australia;
e-mail: gang.li@deakin.edu.au, gleb.beliakov@deakin.edu.au.
[3]Hôpital Tenon, 4 rue de la Chine, 75970 Paris Cedex 20, France; e-mail: carole.philippe@tnn.aphp.fr, amielhelene@gmail.com, adrien.ugon@tnn.aphp.fr.

K-complexes are defined by standardized scoring rules [15], based on the visual appearance of the signal. Manual scoring of *K*-complexes is time consuming, due to noise and the extreme size of data, and subjective, due to variations in human perception. An accurate method for automatic detection of *K*-complexes would therefore be very beneficial. However, automated detection of *K*-complexes is a challenging problem. Although various algorithms have been proposed, based on techniques including artificial neural networks [3], continuous density hidden Markov models [10], wavelet transforms [18] and a matching pursuit approach [11], medical doctors still report that identification accuracy is not satisfactory [9, 14].

In this paper we propose a new algorithm for detecting *K*-complexes. Our procedure is based on an optimization model. We approximate the brain signal by a wave with piecewise linear amplitude, which allows us to create an accurate model for the wave shapes (modelling patterns) and extract relevant characteristics (feature extraction). We minimize the deviation between the data and the modelling patterns. In this application the sum of absolute deviations is preferable to least squares approximation since the data are very noisy [16]. However, this approach necessitates 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 but not as appropriate for *K*-complex detection (see Section 4 for details). 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 organized as follows. More information about the application of this work is given in Section 2. A description of the data we use is given in Section 3. Our proposed procedure for *K*-complex detection is described in detail in Section 4. Results of numerical experiments are reported in Section 5, and conclusions and future research directions are given in Section 6.

2. Motivation

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. PSG methods are used to help diagnose and evaluate a number of sleep disorders. One such disorder is sleep apnoea disorder, in which the patient stops breathing during sleep. This causes so-called micro-arousals where the patient wakes up unconsciously for a brief period. These frequent awakenings are so short that the patient does not even remember them, but they completely destroy normal sleep patterns, leaving the patient sleepy throughout the day. This increases the 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, including

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

on the chin, scalp, and outer edges of the eyelids. The electrodes must remain in place while the patient is sleeping, because signals 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). In total 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 takes a long time to process manually. Thus 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 and three, and rapid eye movement (REM). Currently doctors use a set of predefined rules to allocate each 30-second frame, or epoch, to one of the five sleep states. The application of these rules is done either manually, by visual inspection of the PSG signal, or, more recently, by a software implementation. The allocation of each epoch to a sleep stage is performed using a set of standardized scoring rules, the Rechtschaffen and Kales (R&K) rules [15]. 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, meaning that scoring is prohibitively time consuming. Nevertheless, R&K remains a useful and popular clinical tool for some applications.

Another important study conducted in this area was the *AASM Manual for the Scoring of Sleep and Associated Events* [9]. The goal of this study was to create a manual that reflected the current knowledge and would provide a more comprehensive standardized specification and set of scoring rules for characterizing natural sleep as commonly performed in PSG. This set of rules and specification for the visual scoring of sleep retain much of the framework of R&K, with some new definitions and rule modifications. The new rules provide a better method of analysing data and are crafted as a platform to support the evolution of both manual (visual) and automatic (nonvisual) methods for the future.

3. Data

One of the problems faced in the development of efficient methods for K-complex identification is data availability. There is no freely available test data set for this type of research. This is partly 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 performs the manual scoring after a short PSG analysis course) to create such data sets. Another problem is that the same segment of data might be scored differently by two qualified scorers [9].

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

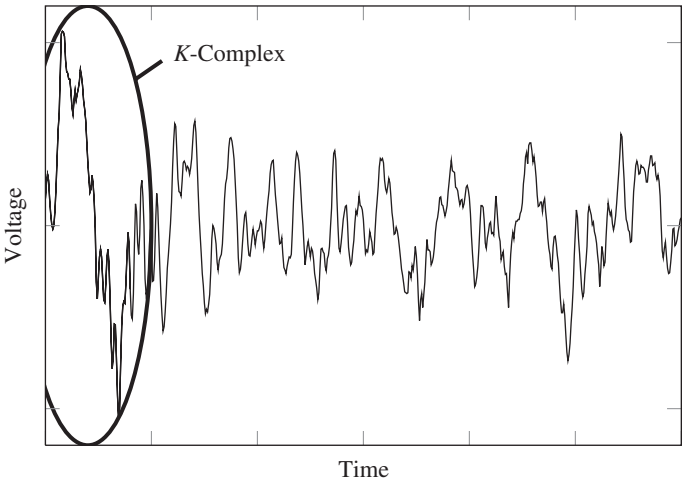

FIGURE 1. A K-complex in practice.

In our experiments we use data scored at Tenon Hospital in Paris, France. On these data, it is more efficient to divide the 30-second epochs into three parts (10-seconds each) and check for the presence of K-complexes in each subepoch separately. Each observation is a 10-second segment of EEG recordings at a frequency of 100 recordings per second (1000 recordings which form 1000 features). As a training set we use a data set with 28 non-K-complexes and 31 K-complexes (59 observations), and as a test set we use a data set 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 the y_i are the EEG readings recorded at time t_i and are the features of our observations. In order to reduce the dimensionality of the data, we apply a special nonsmooth optimization based preprocessing which allows us to reduce the number of features from 1000 to 13. The details are given in the next section.

4. Approach

A K-complex is a brief negative high-voltage peak, followed by a slower positive one. An example of an idealized K-complex can be found in Wikipedia [21]. 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 data set is presented in Figure 1. The K-complex is located at the beginning of the data segment; its amplitude is considerably larger than 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 minimizing the deviation error (sum of absolute deviations) between the actual EEG curve and the modelled wave patterns. This is a proven approach to extracting wave characteristics in order to

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

obtain the targeted wave shape description. These characteristics can be used for the explicit description of wave shape patterns, allowing us to obtain considerably lower dimensionality for the corresponding classification problems. The EEG curve is modelled as the sum of two sinusoidal curves, and the amplitude of each curve is modelled as a piecewise linear function (linear spline). This allows us to obtain more precise curve patterns than in the case of classical sine curves, where the amplitude is constant. Before we proceed, it is necessary to introduce the following definitions.

Polynomial splines are piecewise polynomial functions [13]. In most applications polynomial splines are continuous. The points where the corresponding polynomials are joined together are called *spline knots*, and these may be either 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 [13]:

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

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),

$$(\xi(x))_+ := \begin{cases} \xi(x) & \text{if } \xi(x) > 0 \\ 0 & \text{if } \xi(x) \le 0, \end{cases}$$

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

In order to use a polynomial spline for the amplitude, we need to know the spline parameters and knots. These can be obtained as a solution to the optimization problem

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

where the y_i, $i = 1, \ldots, N$, are the EEG readings recorded at time t_i, $\mathrm{Amp}(X, t)$ is the amplitude function, modelled as a polynomial spline (see 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, and X is a vector which characterizes the amplitude function.

Nonsmooth optimization problems like (4.2) are generally difficult and time-consuming to solve. One way to avoid the nonsmoothness is to use the least squares method rather than the sum of the absolute deviations used in (4.2). However, the least squares method is not very robust when the corresponding data contain many outliers [16]. 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 so nonsmooth optimization models are more suitable for this problem.

We propose the following models for the amplitude function Amp in (4.2).

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

- Linear spline with fixed knots, two intervals:

$$\text{Amp} = \text{Amp}_1(X, t) = a_0 + a_1 t + a_2(t - \theta_1)_+.$$

The knot θ_1 corresponds to the centre of a 10-second interval (5 seconds from the start of each subepoch), and $X = (a_0, a_1, a_2)$. This model for the amplitude function is suitable for subepochs where the behaviour of the actual amplitude changes 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 $a_2 = -a_1$. The model is also suitable for subepochs where the behaviour of the actual amplitude is not changing; for example, if it is decreasing throughout 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:

$$\text{Amp} = \text{Amp}_2(X, t) = a_0 + a_1 t + a_2(t - \theta_1)_+.$$

This model is similar to Amp_1, but θ_1 is considered as an additional variable in the optimization problem (4.2) and thus $X = (a_0, a_1, a_2, \theta_1)$. It is more flexible than Amp_1, but the associated optimization problem is more complicated. The model Amp_2 is suitable for all of the cases described above, but the change in the amplitude function may occur at any point inside a subepoch, not necessarily in the middle. The dimension of the corresponding optimization problem is 6.

- Sharp peak amplitude:

$$\text{Amp} = \text{Amp}_3(X, t) = \max\{a_0, -a_1|t_i - a_2| + a_3\}, \quad a_0, a_1, a_2, a_3 \geq 0, \quad a_2 \leq 10.$$

Here $X = (a_0, a_1, a_2, a_3)$. This model is suitable for subepochs where the amplitude is relatively constant ($a_1 = 0$), or there is a sudden sharp increase in the amplitude inside the subepoch while on the rest of the subepoch the amplitude stays the same. The dimension of the corresponding optimization problem is 6. This amplitude function is a special case of a linear spline with some restrictions on the spline parameters, and can be written in the form (4.1).

In our experiments, we solve the optimization problem (4.2) using the GANSO programming library [5, 19], 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 given function, particularly if the function is nonconvex and nonsmooth. The objective function in (4.2) is nonconvex and nonsmooth and therefore we can only claim that the obtained solutions are locally optimal (optimal in a certain neighbourhood). Thus, 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 large.

Our numerical experiments with Amp_1, Amp_2 and Amp_3 showed that Amp_1 is preferable. Amp_1 is faster than Amp_2, and for Amp_2 the knot θ_1 was for almost all

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

subepochs placed at the beginning or the end of the epoch and was therefore 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 data sets the preference 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:

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

Therefore, the corresponding optimization problem is

$$\min_{X,\omega,\tau} \sum_{i=1}^{N} |f(t_i) - \mathrm{Amp}(X_1, t_i)\sin(2\pi\omega_1 t_i + \tau_1) - \mathrm{Amp}(X_2, t_i)\sin(2\pi\omega_2 t_i + \tau_2)|. \quad (4.3)$$

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 minimize the error exactly as in the optimization problem (4.2). If $(X_1^*, \omega_1^*, \tau_1^*)$ is the solution obtained at the first step then the resulting wave is

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

In the second step we minimize the error over a new data set, where the original data are replaced by the difference between the original data and the wave obtained from the first step:

$$\min_{X,\omega,\tau} \sum_{i=1}^{N} |y_i - W_1 - \mathrm{Amp}(X, t_i)\sin(2\pi\omega t_i + \tau)|. \quad (4.4)$$

If (X_2, ω_2, τ_2) is the solution obtained at the second step (optimization problem (4.4)) then the resulting wave is

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

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

We see that (4.4) is equivalent to the problem

$$\min_{X,\omega,\tau} \sum_{i=1}^{N} |y_i - \mathrm{Amp}(X_1^*, t_i)\sin(2\pi\omega_1^* t_i + \tau_1^*) - \mathrm{Amp}(X, t_i)\sin(2\pi\omega t_i + \tau)|. \quad (4.5)$$

Comparing (4.3) and (4.5), we see that these problems are not equivalent. However, the two-step approach requires solving two smaller optimization problems (the corresponding dimension of each problem is 5) and we therefore use this approach in our numerical experiments.

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

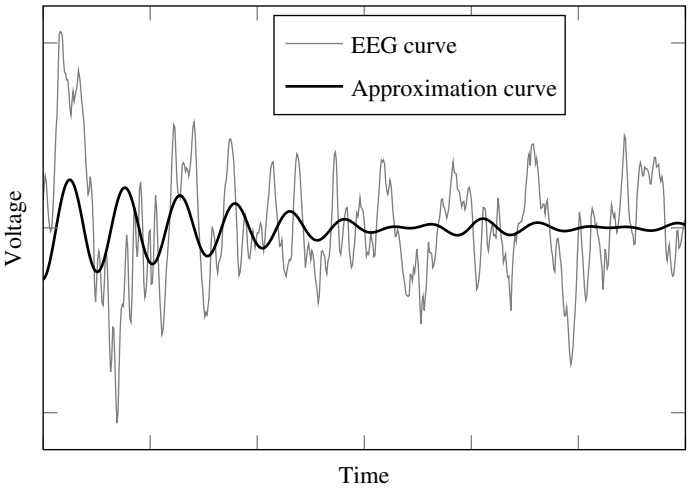

FIGURE 2. Approximation example. The amplitude of the approximation curve is considerably larger at the beginning of the data segment (where the K-complex is located).

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 characterize the improvement of the objective function after nonsmooth optimization. Therefore, 1000 features of the original data are replaced by 13 essential features.

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

4.2. Weka and data classification methods Weka is a collection of machine learning algorithms for data mining tasks [7]. It 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 – integrated software for support vector machine (SVM) classification;
- Logistic – a generalized linear model used for binomial regression;
- MLP – a classifier that uses back-propagation to classify instances, also known as the multilayer perceptron (a special type of neural network);
- RBF – a classifier that implements a normalized 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);

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

Table 1. Test set accuracy (%) for original data sets, 1000 features.

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

- Lazy IBK – a K-nearest-neighbours classifier (uses normalized 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 a C4.5 decision tree;
- 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 of these methods were used with default parameters, except Lazy IBK, which was used with $K = 1, 5, 10$. Weka is open source software; its website [20] provides all of 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 applying nonsmooth optimization. The results are presented in Table 1, from which we see two important outcomes. First, RBF produces very good results. Second, MLP produced no result (software crash after running out of memory), probably due to the large size of the corresponding data (1000 features).

The next step is to apply all of the above classification methods after nonsmooth optimization based preprocessing. We use optimization methods from the GANSO library [19]. Descriptions of the methods mentioned below (ECAM, DFBM, DFBMECAM, DSO) can be found at the GANSO library website [19]. Here we present a brief introduction to them.

- Extended cutting angle method (ECAM) [2, 4]. This method is based on the fact that under Lipschitz continuity, which our objective functions satisfy, one is able 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

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

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

$$|f(x) - f(y)| \leq Ld(x - y),$$

where x and y are two points in the feasible region of f, L is the Lipschitz constant, and d is the (Euclidean, for example) distance between x and y.

- 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 generalization 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 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) [12]. 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.

We also mention ECAM0.1 and ECAM0.001, which are modifications of ECAM with the Lipschitz constant taken as 0.1 and 0.001, with ECAM itself taking the Lipschitz constant as 1. ECAM modifications with a smaller Lipschitz constant are faster but less precise than 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. These results are given in Table 2. 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 data set using RBF, they are still interesting because:

- none of the classification methods failed when performed on the preprocessed data set;
- the accuracy of all of the classification methods from Weka (except RBF and Random Forest) has been considerably improved after nonsmooth optimization based 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

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

Table 2. Classification results (%) after nonsmooth optimization preprocessing.

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
Random Tree	62.71	54.24	54.24	44.07	59.32	66.10
Random Forest	62.72	55.93	57.62	55.93	55.93	55.93

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 outperform slow and precise methods (ECAM, ECAM0.1, ECAM0.001). The best classification results were obtained using the combination of DSO and Random Tree (66.10%), and the combination of DFBMECAM and Logistic (66.01%). Also, 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 method.

Consider the results obtained by RBF without preprocessing (the best accuracy in 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

$$\begin{pmatrix} 34 & 4 \\ 15 & 20 \end{pmatrix}.$$

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

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

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

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

First, we apply RBF to the original data set and remove the segments which have been classified as K-complexes. Then the rest of the data set 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);
- the second step, which is time consuming due to nonsmooth optimization, involves only part of the original data set (49 out of 73).

The results are shown in Table 3. One can see that this procedure does not improve the classification accuracy, but does reduce 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 and ECAM0.1 with MLP produced the best classification results. The proposed combination works well on the available data, but the lack of freely available test data sets prevents us from testing our 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 of the classification methods from Weka (except RBF and Random Forest) has been considerably improved after the preprocessing.

https://doi.org/10.1017/S1446181112000016 Published online by Cambridge University Press

6.2. Further research directions Although our nonsmooth optimization based preprocessing approach performs very well on our data sets, 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 component is time consuming. This issue has to be addressed before implementing our approach in an automatic procedure for K-complex detection which can be used by medical doctors. One way to achieve this is to develop a specific optimization method for this particular problem, which would work faster and more efficiently than the 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", *Optim. Methods Soft.* **17** (2002) 31–60; doi:10.1080/10556780290027837.

[2] A. Bagirov and A. Rubinov, "Modified versions of the cutting angle method", in: *Convex analysis and global optimization*, Volume 54 of *Nonconvex Optimization and its Applications* (eds N. Hadjisavvas and P. M. Pardalos), (Kluwer, Dordrecht, 2001) 245–268.

[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 Trans. Biomed. Eng.* **39** (1992) 1305–1310; doi:10.1109/10.184707.

[4] G. Beliakov, "Cutting angle method – a tool for constrained global optimization", *Optim. Methods Soft.* **19** (2004) 137–151; doi:10.1080/10556780410001647177.

[5] G. Beliakov and J. Ugon, "Implementation of novel methods of global and nonsmooth optimization: GANSO programming library", *Optimization* **56** (2007) 543–546; doi:10.1080/02331930701617429.

[6] P. R. Carney, R. B. Berry and J. D. Geyer (eds), *Clinical sleep disorders* (Lippincott Williams & Wilkins, Philadelphia, 2005).

[7] E. Frank and I. Witten, *Data mining: practical machine learning tools and techniques*, 2nd edn (Morgan Kaufmann, Burlington, MA, 2005).

[8] P. Fuentealba, I. Timofeev, M. Bazhenov, T. J. Sejnowski and M. Steriade, "Membrane bistability in thalamic reticular neurons during spindle oscillations", *J. Neurophysiology* **93** (2005) 294–304; doi:10.1152/jn.00552.2004.

[9] 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* (American Academy of Sleep Medicine, Westchester, IL, 2007).

[10] A. Kam, A. Cohen, A. B. Geva and A. Tarasiuk, "Detection of K-complexes in Sleep EEG Using CD-HMM", *26th Annual Int. Conf. of the IEEE EMBS* **2** (2004) 33–36; doi:10.1109/IEMBS.2004.1403083.

[11] U. Malinowska, P. J. Durka, K. J. Blinowska, W. Szelenberger and A. Wakarow, "Micro- and Macrostructure of sleep EEG", *IEEE Eng. Med. Biol. Mag.* **25** (2006) 26–31; doi:10.1109/MEMB.2006.1657784.

[12] M. A. Mammadov and R. Orsi, "H_∞ synthesis via a nonsmooth, nonconvex optimization approach", *Pac. J. Optim.* **1** (2005) 405–420.

[13] G. Nürnberger, *Approximation by spline functions* (Springer, Berlin, 1989).

[14] R. A. Rajeev and J. Gotman, "Digital tools in polysomnography", *J. Clin. Neurophysiol.* **19** (2002) 136–143.

[15] A. Rechtschaffen and A. Kales (eds), *A manual of standardized terminology, techniques, and scoring system for sleep stages of human subjects* (US Government Printing Office, Washington, DC, 1968).

[16] A. Rubinov, N. Sukhorukova and J. Ugon, "The choice of a similarity measure with respect to its sensitivity to outliers", *Dyn. Contin. Discrete Impuls. Syst. B* **17** (2010) 709–721.
[17] M. Steriade, P. Wyzinski and G. Oakson, "Activities in synaptic pathways between the motor cortex and ventrolateral thalamus underlying EEG spindle waves", *Int. J. Neurol.* **8** (1971) 211–229.
[18] Z. Tang and N. Ishii, "Detection of the *K*-complex using a new method of recognizing waveform based on the discrete wavelet transform", *IEICE Trans. Information Systems* **78** (1995) 77–85.
[19] GANSO, http://guerin.ballarat.edu.au/ard/itms/CIAO/ganso/.
[20] Weka, http://www.cs.waikato.ac.nz/ml/weka/.
[21] Wikipedia, "Sleep spindle", http://en.wikipedia.org/wiki/Sleep_spindle.