Multifractal Feature Vectors for Brain-Computer Interfaces

Nicolas Brodu

***Abstract*—This article introduces a new feature vector extraction for EEG signals using multifractal analysis. The validity of the approach is asserted on real data sets from the BCI competitions II and III. The feature extraction can be performed in real time with low-cost discrete wavelet transforms. Classification results obtained with the new feature vectors are close to the state of art techniques, while using a different information. Combining the new multifractal feature vector with existing ones may result in better performances, up to 5% in the present case. This work thus offers an alternative to the usual feature-extraction techniques, and opens new possibilities in the field of Brain-Computer interfaces.**

I. Introduction

ONE problem today in the field of brain-computer interfaces (BCI) is the task of extracting relevant information from a noisy signal. Indeed, EEG data reflects the global behavior of the brain, condensed over specialized zones covered by only a few electrodes. Noise is thus of two types: the interferences between the brain activity we wish to identify and the other body activities (like muscles), and the noise inherent to the data capture process using electrodes.

The idea of the present work is to explore the use of a data analysis technique which is robust to noise, and which precisely works best when the task is to characterize different types of noise. In fact multifractal analysis, which is the chosen technique, has been expressly designed so as to capture the different kinds of irregularity present in a signal into a condensed form, a "spectrum" of irregularity strengths. This is the information used by the present study: the first part of this paper presents how to estimate this spectrum from the data, what we can do with it, and how it is relevant to the BCI research domain.

A main strength of multifractal analysis compared to the usual feature extraction techniques is thus to make use of a different information in the signal. We might therefore expect that in this new feature space some classifiers work better, while others now fail. The second part of this article consists of an application of eleven different classifiers on the new feature space in order to identify how it might be best exploited.

Previous and related work in this domain include [1] and [2] but these were using the fractal dimension information, either on the series (time domain) or on a continuous wavelet transform (frequency domain). The present work considers the complete multifractal spectrum, not just the dimension information, together with an incremental discrete wavelet transform computational method that is suited to the real-time requirements of BCI.

The next section introduces what is multifractal analysis and what information it actually captures from the signal. Section III then details two different techniques for estimating a multifractal spectrum from the data. The outputs from both these techniques are then used in Section IV as inputs to classifiers on a reference BCI task. Section V discusses these results and Section VI extends the analysis on how to best combine the multifractal feature vector with exiting features. Section VII concludes on this work.

Manuscript received March 15, 2008.

Nicolas Brodu is with the INRIA Institute of Rennes, France; e-mail: nicolas.brodu@free.fr

II. Multifractal Analysis

Multifractal analysis consists in evaluating a repartition of the different irregularities present in a signal. It has been used in a variety of domains, including: medical applications [3], finance [4], environmental research [5], image and signal processing [6], and physics [7], to name a few. In order to show the reader what might be expected from a multifractal analysis Fig.1 cites results from Ivanov et al. [3].

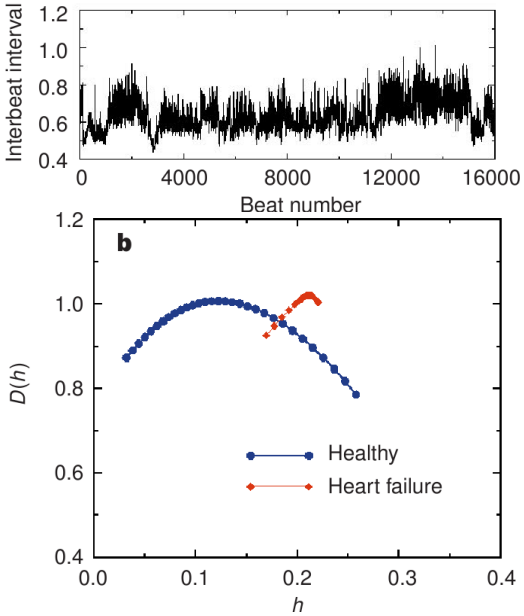

Fig. 1 (cited from [3]). Multifractal analysis is a relevant feature extraction analysis that might identify different dynamical regimes buried in noise. Here is an heart beat series from a healthy subject, plotted together with two multifractal spectra corresponding to two different system states.

So, how can we compute a spectrum curve similar to what is seen on Fig. 1? This is the task of what is called a multifractal formalism, of which several varieties exist.

Mathematically let us decompose the signal into a polynomial part p (Taylor expansion) and a residual r around a point t_0. Let us then write $x(t) = p(t\text{-}t_0) + r(t\text{-}t_0)$. The order of the polynomial part is already a strength information that might be used to characterize the regularity of the series x at the point t_0: the order is *0* for a discontinuous function, *1* for a continuous but not differentiable, etc, and possibly ∞ if the series is completely regular. When this is not the case, we might actually do better than just using the polynomial order information. Let us generalize and assume that there exists $\exists$ c a constant such that $|x(t)\text{-}x(t_0)| \leq c|t\text{-}t_0|^h$. The largest h (if it exists) for which this polynomial and residual exponential

fitting might be done is called the local Hölder exponent of x at t_0. It is then guaranteed that x is differentiable up to degree *floor(h)*, and h then better characterizes the strength of the irregularity: It gives how fast the residual diverges, a continuous measure of strength instead of the discrete polynomial degree information. All that now remains is to quantify how "frequent" are the different h across the series. More precisely, the multifractal spectrum associates to every h the Hausdorff dimension *D(h)* for the set of points having the local Hölder exponent h.

This definition might be mathematically elegant, but it is unfortunately quite cumbersome to implement directly in practice. So, how is the *h* / *D(h)* plot computed on Fig. 1? In that particular cases Ivanov *et al.* used the method described in [7] which relies on a continuous wavelet transform. This is a well-established method, but unfortunately the computational time necessary to carry it on prohibits a real-time use, a requirement for BCI.

Alternative and more efficient estimation techniques have been devised. The next section explains two such techniques that are able to handle the real-time requirement. The first technique relies on the scaling behaviors of the higher moments of *x(t)*, as explained by Mandelbrot *et al.* in [4]. With a simple Legendre transform, a concave approximation of *D(h)* might be recovered [7]. The second technique does not try estimate *D(h)* but instead directly computes the distribution of the various *h* values. The hypothesis is that what is relevant for classification is actually the relative frequency of the various *h* values, and whether this frequency information is present in *D(h)* or in a probability distribution does not matter. This second method relies on [8], but also extends it by additionally introducing kernel density estimation techniques so as to gain more precision compared to the histogram characterization of [8].

As is presented in the next section both techniques have their own advantages and drawbacks. Since they lead to different feature vectors they are also suited to different classification techniques, as explained in Section IV.

III. Obtaining Multifractal Feature Vectors

A. Common part: Discrete Wavelet Transform

Both techniques presented hereafter rely on a discrete wavelet transform (DWT) of the signal. The algorithm that was presented in [9] is used in order to update the DWT incrementally as new data becomes available. This allows the real-time use of the DWT with a constant-time update at each new sample from the electrodes, instead of an O(N) transform (where N is the number of samples used in the DWT) for the non-incremental version of the algorithm.

However the second part of [9] relied on a multifractal estimation technique based on [10], which has subsequently been proven not precise enough by [11] in certain cases (though the cases used in [9] are not affected). This paper thus considers two other techniques that might be plugged in using the incremental framework of [9]. These techniques are presented in Subsections *B* and *C*. A short summary of the incremental algorithm for updating a DWT is now presented, for the needs of this document.

Figure 2 explains the problem inherent to taking into account a new data value in an existing DWT. As we see, all current data is already paired for the computation of the next level of decomposition, and the new data value cannot be taken into account if only one frame is considered. However using both alternative frames allows the pairing immediately, at the cost of now duplicating the computations.

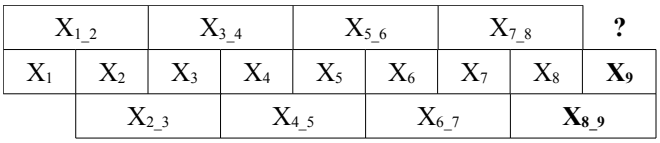

Fig. 2. The data is written in the middle row, the current frame of decomposition on top. A new data value X_9 becomes available. It cannot be paired immediately in the current frame. However, if both alternative pairings are maintained, X_9 can be taken into account immediately.

The trick explained in [9] consists in noticing that some computations might be shared between the different frames of decomposition of the data. While there are 2^L possible framings, with L the number of levels of decomposition, the elements of each level λ also correspond to 2^λ data (see Fig. 3). Fortunately, computations at each level λ might also be shared between $2^{L-\lambda}$ frames of decomposition, hence recovering a linear memory requirement.

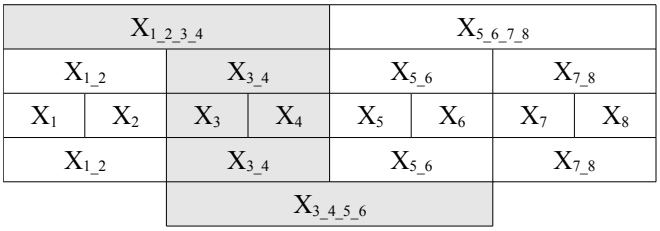

Fig. 3. Sharing computations between different frames of decomposition.

While there are exponentially many possible alignments as the number of levels increase (the alternative pairing of Fig. 2 is not represented on the Fig. 3 diagram), the window size over the data also exponentially increases. Moreover the blocks at one level, like X_{34} on Fig. 3, can be shared amongst all above levels ($X_{1\_2\_3\_4}$ and $X_{3\_4\_5\_6}$). The net result is the recovery of a linear memory requirement, not an exponential one. The interested reader might consult [9] for more information. A link to a reference source code implementing this technique is given in Appendix.

When a new data sample becomes available, all there is to do is to switch from the current to the alternative framing recursively at each level by successively applying the wavelet filter L times. Sharing computations between the different framings is implicitly performed. Hence the promised constant-time and incremental update of the DWT spectrum.

A final refinement, not present in [9], consists in using the wavelet leaders instead of the wavelet coefficients for all further multifractal estimation computations. The interest of wavelet leaders is explained in [12], together with mathematical proofs regarding the expected gain in precision compared to using the wavelet coefficients.

Concretely, this means for the present algorithm that maximal absolute values of the coefficients "below" the level λ update must be maintained during the recursive step. This is explained in Fig. 4.

$X_{1\_2\_3\_4}$				$X_{5\_6\_7\_8}$				$X_{9\_10\_11\_12}$			
X_{1_2}		X_{3_4}		X_{5_6}		X_{7_8}		X_{9_10}		X_{11_12}	
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}

Fig. 4. Wavelet leaders construction (See [12] for more details).

Start from X_7. The wavelet coefficient X_{7_8}, above X_7, is replaced by the coefficient with maximal absolute value in the marked bold-border zone at that level and below. Similarly for $X_{5\_6\_7\_8}$, which takes the maximal absolute value coefficient in the corresponding zone. Wavelet leaders can be computed as presented in Fig. 4, on 3 dyadic intervals around the considered point at each level, or they might also be considered only on the one interval (light gray cells) which results in more variance but less bias[1]. The local Hölder exponent estimation method presented in the main text uses the values in the dark gray cells, for each data point, after the wavelet leaders were computed on only one dyadic interval (though the reference code also supports computing on the 3 intervals).

Now that we have the wavelet leaders it is possible to exploit them so as to estimate a multifractal spectrum of the original series.

B. Method 1: Legendre transform of the moments

The goal of this method is to build statistics in order to relate the series *x(t)* with its representation at a different scale *x(st)/s*. This can be conveniently transposed to relations between the wavelet transform coefficients at different scales, or even better the wavelet leaders [12].

For a range of exponents q, let us compute $f(\lambda,q)$ such that $f(\lambda,q)=\left(\frac{1}{N(\lambda)}\sum_{k=1}^{N(\lambda)}\left|F(\lambda)_k\right|^q\right)^{\frac{1}{q}}$, with $F(\lambda)$ the wavelet leaders at level λ and $N(\lambda)$ the number of wavelet leaders. The experiments in the next part of this article use 64 q values, from -4 to 4, 0 excluded.

The exponential relation between the $f(\lambda,q)$ across the different levels λ gives a value $g(q)$: $f(\lambda,q) \approx s^{g(q)}$ with $s=2^{\lambda}$ the scale. A Legendre transform (see Fig. 5) is finally performed to get the multifractal spectrum [7]: $D(h) = \min_q (qh - g(q) + 1)$.

As we see here $q=2$ also extracts the power in each wavelet frequency band, a commonly used feature vector in the BCI domain. The multifractal spectrum is a complementary information: it does not use the power directly, but rather the relations between the power in different bands, and for various exponents instead of just the squared coefficients. Therefore the multifractal feature vector captures a different information than the usual power-in-the-band feature vector.

This article only explains briefly what computing the multifractal spectrum entails, so the reader knows what operations are necessary. For more information please refer to the aforementioned references.

C. Method 2: Density of the local Hölder exponents

This second method relies on the fact that local Hölder exponents h might also be directly estimated at each sample

[1] Thanks to Pierrick Legrand for pointing this out.

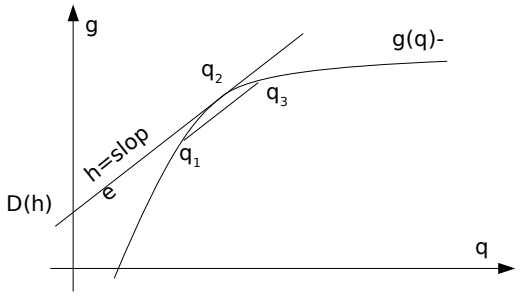

Fig. 5. Legendre transform of g(q) to get the multifractal spectrum D(h).

of the original signal, as presented in [8]. Once this is done, all there is to do is to collect statistics as to how frequent each h appears. [8] uses histograms for that, but this method suffers from a sensitivity to the arbitrary bin size and bin origins. Using a kernel density estimation technique removes the bins altogether, eliminating these problems (the counterpart is the introduction of a kernel size parameter, see below). Figure 6 shows the kernel density estimation of h points distributed according to the theoretical black curve. The resulting probability distribution use a Gaussian kernel.

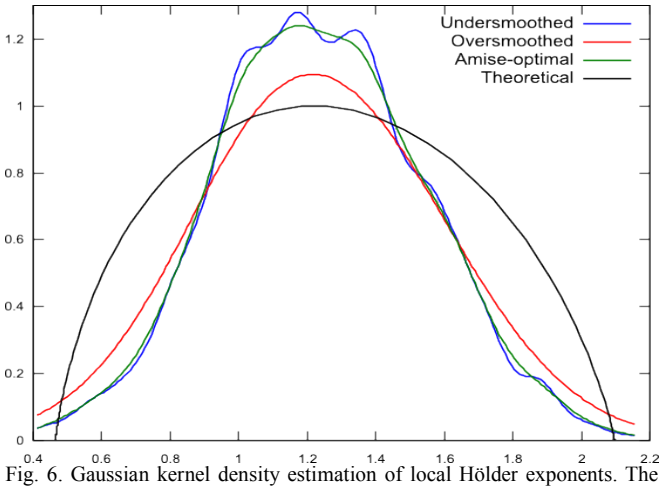

Fig. 6. Gaussian kernel density estimation of local Hölder exponents. The black curve is the theoretical spectrum of a multinomial cascade, the other curves are kernel density estimations using various kernel width.

An immediately apparent limitation of using a Gaussian kernel is the infinite kernel support, whereas theoretically the Hölder exponents may take values only in a finite interval. Future work might consider a finite-support kernel, for example the parabolic-shaped Epanechnikov one.

However for the current problem the shape of the curve is not really important:

- The precision obtained for the spectrum maxima is experimentally found to be better with the kernel density estimation technique than with the Legendre transform technique.
- The feature vector obtained from the spectrum is either a set of points around this maximum (so as to capture the curvature) or a sampling of the curve. In either case what counts for the classifiers is the difference between the two classes, not the difference between the curve obtained and the theoretical spectrum.

As aforementioned there is now the problem of the sensitivity to the kernel width, as represented on Fig. 6. Optimal width estimators exist [13], and they might be

computed from the data. Unfortunately, these estimators are only optimal in the AMISE [13] sense, and not optimal with respect to finding the maxima of $D(h)$. Moreover computing the AMISE-optimal kernel width might be quite costly depending on the required precision. In our case, a better solution is to consider:

- That the optimal kernel width need not be computed at each sample update, the same way an histogram bin size is usually kept fixed. Provided enough computational resources an adaptive kernel width is of course possible, though probably not the best choice for using these resources (see the next point).
- Over-smoothing is actually beneficial in our case, in the sense that under-smoothing might introduce spurious maxima in the spectrum and we do not want this risk (see Fig. 6.). Thus using a fast kernel width selection method that is known for over-smoothing is not a problem: we reduce the risk of spurious Hölder exponent h with maximal $D(h)$.

Combining these two points, a fast kernel width selection method that is called only once in a while, makes the kernel density estimation technique competitive for the real-time requirements of BCI.

IV. Classifing EEG Signals

A. Description of the classification task

The problem considered in this section in the subject of the BCI competition II (2003), data set III [14]. Section VI considers the BCI competition III, data set IIIb. Both share a similar experimental framework. A subject is equipped with electrodes for capturing the cerebral motor activity EEG output. This individual is asked to imagine moving her left or right hand according to an arrow that is displayed on the screen (at random). The classification task is, given the signals captured from the C3, C4, and Cz electrodes during the experiment, to find back which was the direction of the arrow, which hand was active for the given signal.

The data set III from the BCI competition II contains recordings for 3 electrodes, distributed as 140 training and 140 testing instances. The time series is given for a duration of 9 seconds, with the first 2 seconds measuring the base activity, the next second the activity after preparing the graphical display and emitting an audio signal, and in the last 6 seconds the directional arrow is presented to the subject together with a visual feedback corresponding to a simple classifier (see [14] for more details).

We can thus represent the classification problem as a learning task between N time series, with N the number of electrodes, and a class C associated to that time series (left or right). This raises two questions:

- How to extract relevant features from the time series? Indeed, processing directly the series in a classifier is impractical. Moreover the significant information present in the series is lost in noise. Therefore, a good feature extraction technique is often the critical point in a BCI system
- How to classify the obtained feature vectors? This might be solved by any machine learning technique suited to classification problems.

Given a new feature extraction technique, the multifractal one in the present case, we might expect that some classifiers will work better and other will fail. In order to estimate the qualities of the new feature space, it is thus necessary to apply as many classifiers as possible, or at least as many classifiers implying different assumptions on the data as possible. Some such techniques are presented in the next section.

B. Presentation of the different classifiers

1) Linear min sum of squared error

The target classes are written as +1 or -1 entries in a column vector of size the number of training instances. The training feature vectors are represented as the rows of a matrix A, with an additional column in A containing the constant value 1.

The least square solution to the equation $Ax=B$ is computed. The additional column of A recenters the feature vectors so the problem is now symmetrical with respect to the two classes +1 or -1. For a new feature vector a the predicted class is simply deduced from the sign of $(a,1)\cdot x$.

2) Linear Fisher Discriminant with Gaussian intersection threshold

The within-class "scatter matrix" S is computed as the weighted sum of the class covariance matrices: $S=w_0C_0+w_1C_1$ with w_i the proportion of instances belonging to class i, and C_i the covariance matrix of the instances in class i. If μ_i is the mean feature vector of class i, then we can compute the projection vector $w = (\mu_0-\mu_1)^T S^{-1}$.

For a new instance feature vector a, $w\cdot a$ is a scalar that can be thresholded to get a class prediction: a is predicted to class -1 if $w\cdot a > t$, to class 1 otherwise. The threshold t is here chosen as the intersection point of the one-dimensional Gaussians defined by the projections of each class mean and variance. The Gaussians intersection located between each class projected mean m_i is chosen if it exists, otherwise the point at $(m_0+m_1)/2$.

3) Multi-layer Perceptron with one hidden layer

A multi-layer perceptron is created. The input layer is fed with a rescaled version of the feature vectors, such that values are now in the range $[-1...1]$ for each feature.

A hidden layer of sigmoidal nodes is considered, using the transfer function $f(x) = x / (1+abs(x))$ due to its reduced computational costs compared to the more usual tanh (see [15]). The number of hidden nodes is chosen using a 10-fold cross-validation on the training set.

The output layer consists of two nodes, realizing a categorical mapping on the classes. These nodes take the +1 or -1 values if the instance belong to the class for that node. Unlike the linear case above, the categorical mapping is necessary here because the value of the positive output node for one class generally does not correspond to the value of the negative output node for the other class, so combining both produces better results than just using one single node.

Learning is performed by batch back-propagation and simple gradient descent with a learning rate of 0.07 and 500 learning steps.

4) Maximum Likelihood Gaussian mixture

A multivariate Gaussian distribution is fit to each class feature vectors, using the mean μ_i and covariance matrix C_i for that class i. The mixture of Gaussians is performed with w_i the proportion of instances belonging to each class i.

Prediction is simply the class i with maximum likelihood for the instance a to predict: $i = max_j\ p(j|a)$.

5) Variational Bayesian Gaussian mixture

The problem with the maximum likelihood technique is that it is computed without taking into account generalization and may severely overfit. The Variational Bayesian method aims at integrating over all possible prior μ_i, C_i and w_i in order to provide the best generalization capabilities, by assuming these priors are distributed identically in the training and in the test set.

This technique is quite complicated and detailing it is out of the scope of this article. The interested reader is invited to consult [16] for more information, together with the reference source code for this article (see appendix). Compared to [16] and the original [17], the only difference here is that we explicitly know the class mappings, so we initialize the hyperparameters expectation-maximization algorithm from the maximum-likelihood solution.

6) K-Nearest neighbors

Assuming a distance function $d(f_1,f_2)$ between feature vectors f_1 and f_2, it is possible for a new instance feature vector a to compute the distance $d(a,f_x)$ to all known (training) instances x.

Let us then select the K nearest neighbors $x_1 ... x_K$. K might be arbitrarily chosen, but in the next section K is set using a 10-fold cross-validation. In order to give more influence to closest match, distances are normalized between 0 and 1 using the farthest found of the K neighbors, and a kernel is applied so as to reverse the 0/1 relation. Each of the K nearest instance is weighted this way, and a vote occurs for the class decision.

Two distance functions are considered:

- The squared Euclidian distance, after rescaling the features in the *[-1..1]* interval. This is abbreviated nne in the next section.
- The distance function is the Kolmogorov-Smirnov statistic between the two samples. This method is abbreviated nnks in the next section.

7) Support Vector Machines (SVM)

A complete presentation of support vector machines is out of scope of this article. The interested reader might for example consult the seminal article [18], a recent text book [19], or the article [20] accompanying the libsvm library, together with the references therein. The libsvm library is the engine that won two IJCNN challenges (in 2001 and 2002), and the reference source code for the current study uses the updated November 2007 version of libsvm.

Four kernels are supported, applied on input data rescaled in the [-1..1] interval.

- Linear: This is simply margin maximization in the feature space, with a user-set parameter for the cost of making a mistake.
- Gaussian: A Gaussian kernel is applied. The user might specify the kernel width in addition to the cost factor.
- Polynomial: A polynomial of desired degree is used as the kernel. In addition to previous user parameters, the polynomial degree and constant may be specified.
- Sigmoid: A sigmoid is chosen. The sigmoid bias is another free parameter to set.

The results presented in the next section use 10-fold cross-validation in order to set all aforementioned model parameters.

V. Classification Results In The New Feature Space

For each of the recordings of the BCI competition II data set III, each classifier presented in the previous section (noted by their abbreviations in this section) was applied on the two multifractal spectrum estimation methods presented in Section III on a sliding time window of 3 seconds: Legendre Transform (L) and local Hölder (H) exponents density (with a kernel size 0.3). For each of these two approaches the two following extraction techniques were employed: 5 points h in increasing order for which $D(h)$ is at (0.92, 0.96, 1, 0.96, 0.92) of its maximum value (M), and 20 density values for points regularly sampled (S) in range [-0.1..0.9]. The combinations give 4 feature vectors: LM, HM, LS, HS using the above abbreviations. These 4 extraction combinations are applied for each of the 3 electrodes so as to give respectively 15 and 60 dimensional feature vectors. Classifiers that support cross-validation are trained with 10-fold cross-validation.

For each of the feature vector / classifier combination, Table 1 reports the training and testing errors on the BCI II problem III data set, counted at their best point in time so as to be comparable with the results given for the competition.

Classifier		msq	lda	mlp	ml	vb	nne	nnks	svml	svmg	svmp	svms
Training	HS	95.7	95.7	85.7	100	40.8	100	100	82.1	97.9	100	90
	LS	93.6	94.3	85.7	100	50.7	100	100	87.1	74.3	100	91.4
	HM	88.6	88.6	85.7	92.9	81.4	100	100	85.0	100	85.7	85.7
	LM	84.3	85.0	78.6	90.7	73.6	100	100	82.1	100	87.1	82.1
Testing	HS	67.9	67.9	70	62.1	62.9	68.6	70	**73.6**	67.1	**73.6**	**72.9**
	LS	71.4	72.1	**77.1**	65.7	60.7	70	69.3	72.1	71.4	**72.9**	**74.3**
	HM	77.1	77.9	78.6	76.4	**79.3**	**81.4**	**79.3**	**79.3**	**80.0**	**79.3**	**79.3**
	LM	**80.7**	**80.7**	77.9	71.4	76.4	71.4	72.1	**79.3**	74.3	76.4	76.4

Table 1. Comparative study of the classifier / feature vector performances (using 10-fold cross-validation). The classifiers are written in the same order as they were presented in the previous section, and only their abbreviation is used here due to space constraints. Performances are given as the percentage of classification success. Values in **bold** indicate the 3 best results across classifiers for the same feature, values with a gray background indicate the best results across features for each classifier, both including ex-æquo.

As we see in Table 1. The classifiers using a larger sampling of the full spectrum (LS, HS) tend to overfit, excepted the Variational Bayesian approach which precisely was designed not to overfit (though unfortunately it also does not generalize well in this (S) case). When using the (M) feature vectors, only 5 points in the spectrum are used and this tend to prevent some of the observed overfitting (the training classification results are lower and the testing higher). In none of the classifiers considered here does the sampling of the full spectrum (S) give better result than the

reduced sampling one (M). One would thus be tempted to conclude that using *h* only the points around the maximum *D(h)* is better than using the full spectrum. However there are probably better ways than reducing the number of points in the spectrum sampling for handling the overfitting issue.

The results with a gray background show that using the local Hölder exponents method generally produces better results *both* on the testing *and* on the training set, at least for the (M) feature. So the local Hölder exponents method is probably better than the Legendre Spectrum estimation one. Globally in this experiment the (HM) technique works best.

Concerning the classifiers, for the configuration with less overfitting (LM) the linear classifiers perform best. The results are more difficult to interpret for the other configurations (HM, LS, HS) as there is an interference between the generalization capabilities of the non-linear classifier with the overfitting phenomena.

VI. Discussion And Feature Combination

The feature vectors based on multifractal analysis do certainly succeed in extracting some significant information from the data, since all the classifiers considered here perform better than random. The results presented in the previous section are perhaps not as good as the best ones from BCI Competition II (the five top results were between 82.9% and 89.3%), but not as bad as the worst there too (the remaining results were between 50.7% and 76.4%). There are several possible ways of improving the situation, all leading to further research:

1. The discrete wavelet transform considers here a large frequency band. Narrowing or giving more weight to the physiologically relevant frequencies could be a way to improve the new feature vectors. Moreover, if the signal-to-noise ratio increases, this might also help prevent overfitting as the classifiers would not train as much on biologically irrelevant details.

2. No accumulation through time was performed on the classifiers, whereas in BCI competition II the best methods aggregate the results and gather confidence over the whole time series.

More generally, the multifractal spectrum provides some information that is not present in the power-in-a-frequency-band based feature vectors. Combining both approaches might lead to improved results, due to the inclusion of more information. There are many ways for doing so: Combining classifiers might be done through time (confidence accumulation as in BCI competition II) or across features (using both multifractal and another method) or across electrodes (using a priori spatial filtering or a posteriori by combining predictions), and more (like adaboost self-combination approaches).

In order to validate the approach, the following experiment considers :

- The application of the new feature vector to different subjects.
- The combination of the new feature vector with a widely used band-filtering technique. The results demonstrate that multifractal analysis really extracts a different information from the signal, and that combining them results in better performances.

Subjects S4 and X11 from the BCI Competition III, data set IIIb[2] are used here because they follow the same experimental protocol as used in the previous section (motor imagery on EEG). Handling missing label and spatial filtering are not necessary unlike some other data sets from the BCI Competition III, so we are really measuring the effectiveness of the multifractal extraction technique without interference from these other considerations. 540 instances are available for each individual, distributed as 270 instances per class.

This new experiment goal is to analyze if the multifractal feature can be combined with other techniques. A classical band-filtering extraction is performed here: the signal components between 8-12 Hz and between 16-24 Hz are extracted, then squared, averaged over the feedback period, and log-transformed[3].

The previous experiment has dealt with the comparison of multifractal extraction techniques. So, only the local Hölder exponents, spectrum maxima estimation (HM) method is used here as it performed globally better than the others in the previous experiment with a similar setup. A kernel width of 0.7 was chosen here after some preliminary tests, and only the spectrum maximum is retained (so the nnks classifier is not available here). Additionally the first wavelet decomposition level was excluded from the multifractal estimations as it covers frequencies that were filtered out from the signals by the competition organizers. The exact parameters are 5 levels of decomposition (the first one being ignored), 9 coefficients at the highest level, and a moving average of 25 data points. This setup covers exactly the 375 samples corresponding to the feedback period for each individual.

Different techniques were considered in order to combine the predictions issued with both feature vectors[4]:

- Sum: Arithmetic weighted average of each instance class decision by the classifier global confidence in the result.
- Product: Geometric weighted average of each instance class decision by the classifier global confidence in the result. The product rule is sensitive to the quality of the probability estimates for each instance and may result in irrelevant predictions, despite its attracting theoretical properties [19].
- Median: For each instance, choose the class with maximal median value of the confidence × decision levels. This combination rule is supposed to be less sensitive to large outliers than the above average rules.
- Majority: A vote occurs for each class. The class with maximum counts is returned. In case of a tie, the class with maximum overall confidence × decision level is returned.

[2] Individual O3VR is not included here because of space limitation since it corresponds to a different setup, and since it brings in less instances due to a mistake in the competition data sets.

[3] Thanks to Fabien Lotte for providing these band-filter feature vectors.

[4] It would be mathematically incorrect to simply normalize and mix both features in a single vector. Results obtained this way are also worse in the present case than those given in the main text by combining predictions.

– Weighted Majority: A vote occurs for each class, weighted by each classifier global confidence. The difference with Sum is that instance counts are used, not instance decision levels.
– Max decision: For each instance, consider only the class with max decision absolute value across all classifiers (ignores the classifiers confidence except to break ties).
– Max confidence: Consider only the classifier with maximal global confidence (ignores each specific instance decision level except to break ties).

Terminology: An instance class decision is the result of an individual classifier for that instance, a probability that this decision is correct. A classifier confidence in the result is the global classifier confidence that it works well generally (ex: estimated generalization accuracy, etc).

The results of this experiment are provided in tables 2-7.

S4 trn	msq	lda	mlp	ml	vb	nne	svml	svmg	svmp	svms
mfa	*71.3*	***72.2***	*71.1*	*71.3*	*70.2*	***79.4***	***71.9***	*71.7*	*71.7*	*71.7*
band	*70.2*	*70.6*	*71.0*	*72.6*	*70.4*	***83.0***	*70.7*	***74.0***	*72.5*	***73.1***
sum	**74.4**	**74.4**	74.2	70.6	70.4	**88.0**	74.0	**75.5**	73.3	74.2
prod	**74.4**	**74.4**	74.3	59.1	54.4	**88.0**	73.9	**75.5**	73.3	74.2
med	**74.4**	**74.4**	74.2	70.6	70.4	**88.0**	74.0	**75.5**	73.3	74.2
maj	**74.4**	**74.4**	74.2	70.6	70.4	**88.0**	74.0	**75.5**	73.3	74.2
wmaj	71.3	72.2	71.2	**72.6**	70.4	**79.4**	71.8	**73.4**	71.8	72.3
maxd	74.4	73.9	74.2	70.7	70.4	**88.0**	73.8	**76.1**	74.4	**74.8**
maxc	70.2	70.6	71.0	72.6	70.4	**83.0**	70.7	**74.0**	72.5	**73.1**

Table 2. Training classification accuracy for the S4 individual, BCI Competition III, data set IIIb, averaged over 50 runs with distinct random seeds. The first two rows correspond to performances obtained using each *feature vector alone*. The next rows correspond to the combination rules given in the main text. As for table 1 values in **bold** indicate the 3 best results across classifiers for the same feature or combination rule, values with a gray background indicate the best result for each classifier, both including ex-æquo.

In this experiment the training results for the combined methods are most of the time better than these obtained using each single feature vector alone. Yet there is generally no additional overfitting: The better classification performance is also present for the test set as is shown by Table 3.

S4 tst	msq	lda	mlp	ml	vb	nne	svml	svmg	svmp	svms
mfa	***69.8***	***70.2***	***69.8***	***70.2***	*68.7*	*63.1*	*69.5*	*68.8*	*68.7*	*68.5*
band	***69.1***	***69.4***	*68.2*	***70.0***	*67.2*	*64.1*	*68.7*	*68.3*	*68.4*	*68.3*
sum	**74.1**	**73.3**	**74.0**	66.9	67.2	69.1	73.1	70.6	70.4	71.1
prod	**74.1**	73.1	**74.0**	56.1	49.4	68.5	**73.2**	70.6	70.4	71.1
med	**74.1**	**73.3**	**74.0**	66.9	67.2	69.1	73.1	70.6	70.4	71.1
maj	**74.1**	**73.3**	**74.0**	66.9	67.2	69.1	73.1	70.6	70.4	71.1
wmaj	**69.8**	**70.2**	69.7	**70.0**	67.2	63.1	69.4	68.3	68.0	68.1
maxd	**73.7**	73.1	**74.0**	67.0	67.2	69.1	**73.2**	71.4	71.9	72.1
maxc	**69.1**	**69.4**	68.2	**70.0**	67.2	64.1	68.7	68.3	68.4	68.3

Table 3. Testing classification accuracy for the S4 individual, BCI Competition III, data set IIIb. The same conventions as in Table 2 are used for highlighting the best results.

Even the Nearest Neighbors benefited from the combination, despite the fact it overfits. Only the Gaussian Mixture models do not seem to support being combined in this experiment. For the individual S4 it is interesting to note that results from using the multifractal feature alone are similar to these from using the band-power feature alone, even slightly better on average. Their combination generally leads to improved results, up to +5% classification improvement, and +5% is also the best case with the linear methods, across different combination rules, at 74.1%. That best score is also more than could be achieved using either of the multifractal or the power-band feature alone.

The variances for the methods that make use of a random initialization are given in table 4.

S4 std.	train					test				
dev.	mlp	svml	svmg	svmp	svms	mlp	svml	svmg	svmp	svms
mfa	*0.3*	*0.5*	*0.5*	*0.5*	*0.5*	*0.2*	*1.1*	*1.4*	*1.4*	*1.4*
band	*0.9*	*0.4*	*1.5*	*0.9*	*1.4*	*0.6*	*0.5*	*0.9*	*1.1*	*0.8*
sum	0.4	0.9	2.9	3.0	2.7	0.5	1.1	2.9	3.3	3.3
prod	0.4	0.9	2.9	3.0	2.7	0.5	1.1	2.9	3.3	3.3
med	0.4	0.9	2.9	3.0	2.7	0.5	1.1	2.9	3.3	3.3
maj	0.4	0.9	2.9	3.0	2.7	0.5	1.1	2.9	3.3	3.3
wmaj	0.6	0.6	1.9	2.2	2.0	0.9	1.1	1.8	2.2	2.0
maxd	0.4	0.9	2.1	1.7	1.7	0.6	1.2	2.1	2.3	2.3
maxc	0.9	0.4	1.5	0.9	1.4	0.6	0.5	0.9	1.1	0.8

Table 4. Standard deviations for the results in table 2 and 3, for the classifiers that make use of a random initialization.

While the multi-layer perceptron is relatively stable, the SVM models exhibit a large variability. The linear methods (msq, lda) and the Gaussian Mixture models (ml, vb) do not make use of a random initialization and are therefore the most stable. Based on table 3 and 4, it seems that the best compromise for a reliable and good score would be either the linear or the multi-perceptron classifier. The results for the X11 individual are presented in tables 5 and 6 and present the same variability, though these results are very different for the multifractal feature.

X11 trn	msq	lda	mlp	ml	vb	nne	svml	svmg	svmp	svms
mfa	*52.8*	*53.1*	*53.3*	*53.5*	*53.0*	***100.0***	***53.4***	***53.7***	*51.2*	*51.8*
band	*73.0*	*73.1*	*73.4*	***74.1***	*73.5*	***81.1***	*73.4*	***73.9***	*73.5*	*73.1*
sum	73.0	73.1	73.3	**74.3**	**73.5**	**100.0**	56.8	56.3	54.0	55.2
prod	73.0	**73.1**	**73.3**	54.3	38.7	**100.0**	56.8	56.3	54.0	55.2
med	73.0	73.1	73.3	**74.3**	**73.5**	**100.0**	56.8	56.3	54.0	55.2
maj	73.0	73.1	73.3	**74.3**	**73.5**	**100.0**	56.8	56.3	54.0	55.2
wmaj	73.0	73.1	73.4	**74.1**	73.5	**81.1**	73.4	**73.9**	73.5	73.1
maxd	73.1	**73.9**	73.4	71.3	**73.5**	**100.0**	56.8	56.3	54.0	55.2
maxc	73.0	73.1	73.4	**74.1**	73.5	**81.1**	73.4	**73.9**	73.5	73.1

Table 5. Training classification accuracy for the X11 individual, BCI Competition III, data set IIIb, averaged over 50 runs. The same conventions as in Table 2 are used for highlighting the best results.

There does not seem to be any useful information in the multifractal feature vector for the X11 individual. The test results in Table 6 confirm this point.

X11 tst	msq	lda	mlp	ml	vb	nne	svml	svmg	svmp	svms
mfa	***51.1***	*50.6*	***51.0***	*50.2*	*50.6*	*49.3*	*50.9*	*50.8*	*50.5*	***51.2***
band	*72.6*	***73.5***	*72.5*	***73.3***	*68.7*	*69.3*	***72.8***	*72.6*	*72.1*	*71.8*
sum	**72.8**	**73.7**	**72.4**	**72.4**	68.7	56.9	55.1	55.0	52.8	53.5
prod	**72.8**	**73.7**	**72.4**	50.2	33.5	46.7	55.1	55.0	52.8	53.5
med	**72.8**	**73.7**	**72.4**	**72.4**	68.7	56.9	55.1	55.0	52.8	53.5
maj	**72.8**	**73.7**	**72.4**	**72.4**	68.7	56.9	55.1	55.0	52.8	53.5
wmaj	72.6	**73.5**	72.5	**73.3**	68.7	69.3	**72.8**	72.6	72.1	71.8
maxd	**72.4**	**73.9**	**72.5**	66.7	68.7	50.0	55.1	55.0	52.8	53.5
maxc	72.6	**73.5**	72.5	**73.3**	68.7	69.3	**72.8**	72.6	72.1	71.8

Table 6. Testing classification accuracy for the X11 individual, BCI Competition III, data set IIIb, averaged over 50 runs. The same conventions as in Table 2 are used for highlighting the best results.

Table 7 confirms the relative stability of the multi-layer perceptron compared to the SVM.

X11	train					test				
std. dev.	mlp	svml	svmg	svmp	svms	mlp	svml	svmg	svmp	svms
mfa	*0.8*	*0.4*	*1.8*	*1.8*	*2.1*	*0.9*	*0.4*	*0.6*	*1.7*	*1.4*
band	*0.4*	*0.4*	*1.3*	*0.9*	*1.0*	*0.4*	*0.4*	*0.6*	*1.2*	*1.3*
sum	0.5	3.3	2.8	2.8	1.9	0.4	3.8	2.7	1.5	1.8
prod	0.5	3.3	2.8	2.8	1.9	0.4	3.8	2.7	1.5	1.8
med	0.5	3.3	2.8	2.8	1.9	0.4	3.8	2.7	1.5	1.8
maj	0.5	3.3	2.8	2.8	1.9	0.4	3.8	2.7	1.5	1.8
wmaj	0.4	0.4	1.3	0.9	1.0	0.4	0.4	0.6	1.2	1.3
maxd	0.5	3.3	2.8	2.9	1.9	0.4	3.8	2.7	1.5	1.8
maxc	0.4	0.4	1.3	0.9	1.0	0.4	0.4	0.6	1.2	1.3

Table 7. Standard deviations for the results in table 5 and 6, for the classifiers that make use of a random initialization.

For the X11 individual, the multifractal feature vector does not seem to contain any useful information. However the multifractal worked (slightly) better than the power-band feature for the S4 individual. Is multifractal subject to such a great variability from individual to individual? A possible answer is shown in Fig. 7, where the average spectrograms are plotted for the S4 and X11 individuals.

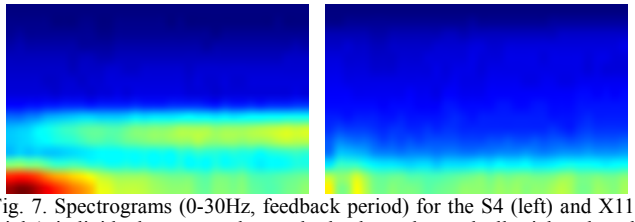

Fig. 7. Spectrograms (0-30Hz, feedback period) for the S4 (left) and X11 (right) individuals, averaged over both electrodes and all trials, plotted using the same energy scale.

The multifractal technique relies on the relations between the different frequency bands in the signal. In the S4 case there is a clear component at two main frequencies, while in the X11 case there is only really one usable band and the multifractal feature extraction thus fails. Moreover, the signal preprocessing performed by competition organizers (filtering between 0.5 and 30Hz) might also interfere with the multifractal extraction process: it would be interesting to perform the same experiment on the unfiltered data, if these data were available.

The problem for further generalization is to be able to select a good combination rule and classifier from only the cross-validated training results. Fortunately we can clearly see if the multifractal technique works for a given individual / EEG signal on the training set, see tables 1, 2, and 5.

Analyzing the highlighted maxima in tables 3 and 6 gives a best overall score and robustness to either of the Sum, Median, or Majority combination rules, together with the linear methods. The multi-layer perceptron is not far behind. The support vector machines are quite sensitive to the combination rule used, and suffer from a large variance in the results from one run to another, though they performed best for the first experiment (Table1). The Gaussian Mixture methods were never the best in any of these experiments.

Conclusion

The multifractal approach to EEG signal classification represents an alternative or a complement to current feature vectors. It presents original properties that are not available in competing techniques. It makes use of the relations between power-related quantities at different frequencies, rather than using the power information itself. Multifractal analysis thus deserves more attention and consideration in standard BCI tools, which would promote more research on its alternative properties.

Appendix: Source Code

Free-libre software implementing multifractal feature vector extraction, and independently implementing all the classifiers mentioned in this article (i.e. applicable to other feature vectors as well), is available at:

http://gforge.inria.fr/projects/mesincom/

revision 92 or more recent, and on the author web site.

Aknowledgments

The author thanks Fabien Lotte, Anatole Lécuyer, and the OpenVibe ANR project team for helpful comments and reviews. As required by the BCI Competitions, a reference [21] is given to one of the challenge organizers publications.

References

[1] R. Boostani and M.H. Moradi, “A new approach in the BCI research based on fractal dimension as feature and Adaboost as classifier”. Journal of Neural Engineering 1(4):212-7, Dec. 2004

[2] Wei-Yen Hsu, Chou-Ching Lin, Ming-Shaung Ju and Yung-Nien Sun, “Wavelet-based fractal features with active segment selection: Application to single-trial EEG data”. Journal of Neuroscience Methods 163(1):145-160, Jun. 2007.

[3] P. C. Ivanov, L. A. Nunes Amaral, A. L. Goldberger, S. Havlin, M. G. Rosenblum, Z. Struzik, H. E. Stanley, “Multifractality in healthy heartbeat dynamics”. Nature 399:461-465, 1999.

[4] B. Mandelbrot, A. Fisher, L. Calvet, “A multifractal model of asset returns”. Cowles Fundation discussion paper number 1164, 1997.

[5] I. Tchiguirinskaia, S. Lu, F. J. Molz, T. M. Williams, D. Lavallée, “Multifractal versus monofractal analysis of wetland topography”. Stochastic Environmental Research and Risk Assessment 14:8-32, 2000.

[6] E. Lutton, J-L. Véhel, “Evolutionary multifractal signal / image denoising”. Evolutionary Computer vision, EURASIP Book Series, 2007.

[7] J. F. Muzy, E. Bacry, A. Arneodo, “Multifractal formalism for fractal signals: The structure-function approach versus the wavelet-transform modulus-maxima method”. Physical Review E 47(2):875-884, 1993.

[8] P. Legrand, “Classification par modélisation de distributions de paramètres ponctuels ou locaux”, unpublished technical report, Apr. 2006.

[9] N. Brodu, “Real-time update of multi-fractal analysis on dynamic time series using incremental discrete wavelet transforms”. Unpublished preprint, http://arxiv.org/abs/nlin/0511041, 2005.

[10] P. Manimaran, P.K. Panigrahi, J.C. Parikh, “Wavelet analysis and scaling properties of time series”. Physical Review E 72:046120, 2005.

[11] R.B. Govindan, “Comment on 'Wavelet Analysis and scaling properties of time series'”. Unpublished preprint, http://arxiv.org/e-print/physics/0701009, Jan. 2007.

[12] P. Abry, S. Jaffard, B. Lashermes, “Revisiting Scaling, Multifractal, and Multiplicative Cascades with the Wavelet Leader Lens”, Proceedings of the Wavelet Applications in Industrial Processing II SPIE conference, 5607:103-117, 2004.

[13] V. C. Raykar, R. Duraiswami. “Fast optimal bandwidth selection for kernel density estimation”. Proceedings of the sixth SIAM International Conference on Data Mining 524-528, Apr. 2006.

[14] A. Schlögl, “Outcome of the BCI-competition 2003 on the Graz data set”. Technical report, available online at the competition web site http://ida.first.fraunhofer.de/projects/bci/competition_ii/results/TR_BCI2003_III.pdf, May 2003.

[15] D.L. Elliott. “A Better Activation Function for Artificial Neural Networks”. Technical Report 93-8, Institute for Systems Research, University of Maryland. 1993.

[16] N. Nasios, A.G. Bors, “Variational Learning for Gaussian Mixture Models”, IEEE transactions on systems, man and cybernetics – part B: Cybernetics 36(4). Aug. 2006.

[17] H. Attias, “Advances in Neural Information Processing Systems”, MIT Press, Cambridge, MA 2000.

[18] B. Boser, I. Guyon, V. Vapnik, “A training algorithm for optimal margin classifiers”. Fifth Annual Workshop on Computational Learning Theory, 144-152, 1992 .

[19] S. Theodoridis and K. Koutroubas, “Pattern Recognition, Third Edition”. Academic Press, 2006.

[20] C-C. Chang, C-J. Lin, “LIBSVM: A library for Support Vector Machines”, 2001. Free-libre software, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[21] A. Schlögl, C. Neuper, G. Pfurtscheller, “Estimating the mutual information of an EEG-based Brain-Computer-Interface”, Biomedizinische Technik 47(1-2): 3-8, 2002.