Neural Comput & Applic (2016) 27:791–804
DOI 10.1007/s00521-015-1953-5

ORIGINAL ARTICLE

Classification of electromyography signals using relevance vector machines and fractal dimension

Clodoaldo A. M. Lima[1] · André L. V. Coelho[2] · Renata C. B. Madeo[1] · Sarajane M. Peres[1]

Received: 4 November 2014 / Accepted: 5 June 2015 / Published online: 24 June 2015

Abstract Surface electromyography (EMG) signals have been studied extensively in the last years aiming at the automatic classification of hand gestures and movements as well as the early identification of latent neuromuscular disorders. In this paper, we investigate the potentials of the conjoint use of relevance vector machines (RVM) and fractal dimension (FD) for automatically identifying EMG signals related to different classes of limb motion. The adoption of FD as the mechanism for feature extraction is justified by the fact that EMG signals usually show traces of self-similarity. In particular, four well-known FD estimation methods, namely box-counting, Higuchi's, Katz's and Sevcik's methods, have been considered in this study. With respect to RVM, besides the standard formulation for binary classification, we also investigate the performance of two recently proposed variants, namely constructive mRVM and top-down mRVM, that deal specifically with multiclass problems. These classifiers operate solely over the features extracted by the FD estimation methods, and since the number of such features is relatively small, the efficiency of the classifier induction process is ensured. Results of experiments conducted on a publicly available dataset involving seven distinct types of limb motions are reported whereby we assess the performance of different configurations of the proposed RVM+FD approach. Overall, the results evidence that kernel machines equipped with the FD feature values can be useful for achieving good levels of classification performance. In particular, we have empirically observed that the features extracted by the Katz's method is of better quality than the features generated by other methods.

Keywords EMG signal classification · Relevance vector machines · Fractal dimension · Feature extraction

1 Introduction

In the last decades, the surface electromyography (EMG) signal has been widely investigated for the purpose of neuromuscular disorder diagnosis, rehabilitation and control of prosthetic devices as well as man–machine interface, targeting individuals with amputations or congenitally deficient limbs [2, 8, 17, 24, 26, 37, 41]. This is because the EMG signal provides a highly useful characterization of the neuromuscular system, also allowing that many pathological processes—whether arising in the nervous system or in the muscles—manifest themselves by alterations in the signal properties.

In order to accomplish the analysis and processing of EMG signals, mainly aiming at performing pattern classification, different approaches have been proposed in the literature, most of which are composed of two interdependent modules [15, 26]: (1) *feature extraction* and (2) *classification*. Feature extraction is especially helpful if the

✉ Clodoaldo A. M. Lima
c.lima@usp.br

André L. V. Coelho
acoelho@unifor.br

Renata C. B. Madeo
renata.si@usp.br

Sarajane M. Peres
sarajane@usp.br

[1] Information Systems Program, School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, Brazil

[2] Graduate Program in Applied Informatics, Center of Technological Sciences, University of Fortaleza, Fortaleza, Brazil

pattern to be represented is a sequence of values taken as a function of time, say $x(t)$, such as the EMG signal. In general, there are four classes of feature extraction approaches to representing 1D signals, namely those based on time, frequency, time–frequency, and nonlinear dynamics.

It has been shown that biomedical signals, such as the EMG, are inherently nonlinear in nature, exhibiting well-defined properties, such as scale invariance, scaling range, power law scaling, and self-similarity [14, 38]. The phenomenon of self-similarity, in particular, whereby a small scale structure can resemble the large-scale structure of an object, has been investigated for the purpose of characterizing different biomedical signals as well as for identifying different patterns available in these signals [25, 32]. In fact, EMG signals usually show noticeable traces of self-similarity that could be captured by fractal dimension (FD) measures [22], representing a way to extract discriminative features directly from these signals [13]. Grossly speaking, FD amounts to a non-integer or fractional dimension of a geometric object [4, 44].

In [33], among different nonlinear methods investigated for representing EMG signals, fractal dimension was found to be especially interesting for its sensitiveness to the magnitude and rate of the generated muscle force. On the other hand, in the work of Hu et al. [22], FD was calculated from filtered surface EMG signals in order to discriminate between forearm supination (FS) and forearm pronation (FP) movements. The results reported by the authors showed that the values of fractal dimension of filtered FS surface EMG signals and those of filtered FP surface EMG signals distribute in two different regions, demonstrating the usefulness of FD in capturing different motion patterns of surface EMG signals. More recently, Phinyomark et al. [34] have investigated the specific case of low-level EMG signal classification through a single-channel system, which comes to be a difficult pattern classification task. The authors concluded that detrended fluctuation analysis (DFA), which is an advanced fractal analysis method suited for the identification of low-level muscle activations, performs better than other conventional features in the classification of EMG signals from bifunctional movements, such as flexion–extension. By other means, Ancillao et al. [3] have conducted an experimental study investigating the correlation between the fractal dimension of the surface EMG signal recorded over the main erector muscle of the human leg, viz. the rectus femoris muscle, during a vertical jump and the height reached in that jump. The authors concluded that FD is able to properly characterize the EMG signal, and a linear regression analysis showed a very high correlation coefficient between the fractal dimension and the height of the jump achieved by all the 20 healthy subjects recruited.

Regarding the classification stage, this can be briefly defined as the process of assigning one out of C discrete labels (classes) for a given input vector $\boldsymbol{x}$ [5]. The classification of EMG signals, in particular, appears to be a hard pattern recognition task to pursue since there are usually lots of interferences and fluctuations happening in the EMG signal [21]. Numerous empirical studies have been conducted investigating the use of different types of classifiers operating on different types of features extracted from the EMG signal. These classifiers include artificial neural networks (ANN) [9], linear and quadratic discriminant analysis [6, 35], Bayesian classifiers [16], fuzzy classifiers [7], and also support vector machines (SVM) [12, 28, 31, 45]. In a recent work [46], Yousefi and Hamilton-Wright conducted a critical review of some of the classification methodologies used in EMG characterization and also present the state-of-the-art accomplishments in this field, emphasizing neuromuscular pathology.

Most of the aforementioned classifiers are based on the idea of solely minimizing the training error, which is usually called empirical risk. However, the combination of limited amounts of training data and the quest for high classification accuracy over these data often leads to overfitting problems [5]. In addition, the levels of accuracy exhibited by these classifiers are usually much sensitive to the feature dimension of the given pattern set. Since they are not plagued by these deficiencies, SVM appear as the method of choice in coping with highly complex classification problems, such as those involving biomedical signals.

The relevance vector machines (RVM) were introduced by Tipping [42] as a Bayesian variant of SVM, which means that they also do not suffer from the aforementioned drawbacks. The RVM yield a probabilistic sparse model identical in functional form to the SVM, representing a new approach to pattern classification that has recently attracted a great deal of interest. In many problems, RVM classifiers have produced competitive results to other kernel-based classifiers, being recently thoroughly investigated in the context of electroencephalogram (EEG) signal classification for epilepsy diagnosis [29, 30].

In order to deal directly with multiclass classification problems, the RVM formulation has been recently adapted [36]. A straightforward multiclass adaptation of RVM is problematic due to the bad scaling of the maximization of the marginal likelihood procedure with respect to the number of classes [10] and dimensionality of the Hessian required for the Laplace approximation [5]. In [36], Psorakis et al. conceived an approach to circumvent these difficulties, bringing about two multiclass multikernel RVM methods (hereafter referred to as mRVM) that are able to address multikernel learning while producing both sample-wise and kernel-wise sparse solutions.

In this paper, we investigate the conjoint use of RVM and FD for tackling the task of EMG signal classification. For this purpose, besides the standard RVM formulation, two types of mRVM, namely constructive mRVM and top-down mRVM, as well as different methods for calculating the FD of an EMG signal, were considered. As far as the authors are aware of, this is the first work providing a thorough assessment of the potentials of combining RVM and FD into a single EMG signal classification framework. Several experiments have been conducted on a dataset involving seven distinct types of limb motions, and the performance of distinct configurations of the RVM+FD approach is reported.

The rest of the paper is organized as follows. In Sects. 2 and 3, we present four methods for estimating the FD from a 1D signal and the mathematical formulations behind RVM and mRVM models, respectively. In Sect. 4, we characterize the EMG dataset used in the experiments and outline some procedures adopted for data preprocessing. We then present and discuss the results achieved by different configurations of the RVM+FD approach, taking as reference the performance delivered by SVM models. Finally, Sect. 5 concludes the paper and brings remarks on future work.

2 Fractal dimension

In a nutshell, fractal dimension alludes to a statistical index of complexity, indicating how the details in a given physical pattern (or object) change with the scale at which they are measured [1, 4]. The value of this index is usually a non-integer, fractional number, hence the designation of a fractal dimension. There are many notions of FD, and various algorithms have been proposed to compute them [44]. None of these methods, however, should be considered as universal, which justifies an empirical comparison of their abilities as feature extractors from EMG signals. In the following subsections, we outline the four methods adopted in our experiments.

2.1 Box-counting method

The idea behind the box-counting (BC) method is to apply successive hypercube grid coverings over a curve (e.g., an 1D signal), yielding as a result a value which is usually very similar to that produced by the Hausdorff Dimension, which is another standard method for calculating the FD [4]. Since in each iteration of the BC method, a finer covering is applied, the method is said to perform a finer and finer analysis on the fractal. Usually, when this method is used, the final FD measure is named as box-counting dimension.

For the calculus of the BC dimension, the successive coverings generated by the method are reflected on a log–log curve (a.k.a. BC curve), which is composed of points that represent the relation between the shrinking of the hypercubes and their occupation rates. The straight line that best approaches the BC curve represents the behavior of the observations from the signal under analysis. The power law of this curve (i.e., the slope of the straight line that best fits it) represents the BC of the fractal.

Formally speaking, the calculation of the BC dimension (D) is given by [4]:

$$D = \lim_{n \to \infty} \frac{\log(Nn(\Lambda))}{\log(2^n)},$$

where $\Lambda \in \mathfrak{H}(\Re^m)$ is an attractor in the Euclidean metric space whose points are compact subsets of $\Re^m$; $Nn(\Lambda)$ is the number of boxes intersecting the attractor; and n denotes the nth iteration of the process. Simply put, the BC method covers $\Re^m$ with a grid of boxes with lateral size equal to $1/2^n$.

2.2 Higuchi's method

As the former, the Higuchi's method [19, 44] is iterative in nature. However, it is especially indicated to handle waveforms as objects. Consider $s = \{s(1), s(2), \ldots, s(N)\}$ as an epoch of the time series to be analyzed. Then, construct k new time series (aka sub-epochs) s_m^k, each of which being defined as [44]

$$s_m^k = \left\{ s(m), s(m+k), s(m+2k), \ldots, s\left(m + \left\lfloor \frac{(N-m)}{k} \right\rfloor k \right) \right\},$$

where N is the total length of the data sequence s; $m = 1, 2, 3, \ldots, k$ indicates the initial time value; k indicates the discrete time interval between points (delay); and $\lfloor \cdot \rfloor$ means the floor operator.

For each of the sub-epochs s_m^k, the average length $L_m(k)$ is computed as

$$L_m(k) = \frac{1}{k} \left\{ \frac{(N-1)}{\left\lfloor \frac{(N-m)}{k} \right\rfloor k} \sum_{i=1}^{\left\lfloor \frac{(N-m)}{k} \right\rfloor} |s(m+ik) - s(m+(i-1)k)| \right\},$$

where $(N-1)/\lfloor (N-m)/k \rfloor k$ is a normalization factor.

Then, the length of the epoch $L(k)$ for the time interval k is computed as the mean of the k values, for $m = 1, 2, \ldots, k$, as given in Eq. (1). This procedure is repeated for each k, ranging from 1 to $k_{\max}$ ($k_{\max} = 5$ in our experiments).

$$L(k) = \sum_{m=1}^{k} L_m(k). \tag{1}$$

The total average length $L(k)$, for scale k, is proportional to k^D, where D is the FD of the curve describing the shape of the epoch as calculated by Higuchi's method. Otherwise, if $L(k)$ is plotted against k on a double-logarithmic scale, then the coefficient of the linear regression of this plot can be taken as an estimate of the FD of the epoch [19].

2.3 Katz's method

Consider $s(i) = (x_i, y_i)$, $i = 1, 2, \ldots, N$, where x_i are values of the abscissa and y_i are values of the ordinate. If the points $s(i)$ and $s(j)$ are represented as (x_i, y_i) and (x_j, y_j), respectively, the Euclidean distance between the points is computed as:

$$\mathrm{dist}(s(i), s(j)) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

According to the Katz's method, the FD of the curve representing a time series can be defined as [27]:

$$D = \frac{\log(L)}{\log(d)}, \tag{2}$$

where L is the total length of the curve or the sum of the Euclidean distances between successive points in the same curve, and d is the diameter estimated as

$$d = \max(\mathrm{dist}(s(i), s(j))), \quad i, j = 1, \ldots, N.$$

If there are no intersections of the curve, i can be set equal to 1 and d can be estimated as the maximum distance between the first sample and the farthest of all subsequent samples in $s(i), i = 2, \ldots, N$.

Obviously, d and L should be dimensionless number to calculate the logarithms in Eq. (2). However, this is not always true. Katz [27] proposed to normalize d and L by the length of the average step, defined as L/N_l. In this way, Eq. (2) becomes

$$D = \frac{\log(N_l)}{\log(N_l) + \log(d/L)}, \tag{3}$$

where $N_l = N - 1$.

2.4 Sevcik's method

Let $y_i, i = 1, \ldots, N$ be a set of values sampled from a signal between time zero and $t_{\max}$ with sampling period δ. Suppose also that the waveform is submitted to a double-linear transformation that maps it into a unit square. Then, the normalized abscissa x_i^* and the normalized ordinate y_i^* of the square can be defined, respectively, as [40]

$$x_i^* = \frac{x_i}{x_{\max}},$$
$$y_i^* = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}},$$

where $x_{\max}$ ($y_{\max}$) denotes the maximum value of x_i (y_i), and $y_{\min}$ is the minimum value of y_i. Thus, the FD of the waveform can be approximated by [40]

$$D = 1 + \frac{\ln(L)}{\ln(2N_l)}$$

where ln is the natural logarithm, L is the length of the curve in the unit square and $N_l = N - 1$.

3 Relevance vector machines and their multiclass versions

As mentioned before, RVM can be regarded as a Bayesian variant of SVM, aimed at overcoming some of the SVM limitations [5, 30, 42]. In this section, we present the basic formulation underlying standard RVM classifiers and also the recently proposed multiclass versions [18, 36].

3.1 Relevance vector machines

The standard formulation of the RVM assumes, for a given input $\boldsymbol{x}_n$, that the error between the classifier output, given by $f(\boldsymbol{x}_n; \boldsymbol{w})$, and the desired output t_n, where $t_n \in \{0, 1\}$, has a normal distribution with zero mean and variance σ^2. It also assumes that the samples $\{\boldsymbol{x}_i, t_i\}_{i=1}^N$ are independently generated, so that the likelihood of the observed dataset can be written as [42]:

$$p(\boldsymbol{t}|\boldsymbol{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}||\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w}||^2\right\},$$

where $\boldsymbol{t} = [t_1, \ldots, t_N]^T$, $\boldsymbol{w} = [w_0, \ldots, w_N]^T$, and $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_N)]^T$, with $\boldsymbol{\phi}(\boldsymbol{x}_i) = [1, K(\boldsymbol{x}_i, \boldsymbol{x}_1), \ldots, K(\boldsymbol{x}_i, \boldsymbol{x}_N)]^T$. The function $K(\cdot, \cdot)$ denotes a kernel function defined on a (high-dimensional) dot product space [39], whereas the final decision function is given by $f(\boldsymbol{x}_n; \boldsymbol{w}) = \sum_{i=0}^N w_i K(\boldsymbol{x}_i, \boldsymbol{x}_n)$.

RVM uses an *a priori* probability over the model parameters (weights) controlled by a set of hyper-parameters. Each weight becomes associated with a hyper-parameter, and most likely values for the weights are estimated iteratively from the training data [42]. In a Bayesian perspective, the model parameters $\boldsymbol{w}$ and σ^2 can be estimated initially from an *a priori* distribution and then reestimated by calculating a posterior distribution using the observed data likelihood. Tipping [42] proposed the following *a priori* distribution for each model parameter:

$$p(w_j|\alpha_j, \sigma^2) = \sqrt{\frac{\alpha_j}{2\pi}} \exp\left\{-\frac{\alpha_j w_j^2}{2}\right\} = \mathcal{N}(0, \alpha_j^{-2}),$$

where $j = 0, \ldots, N$ and $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_N]^T$ is the hyper-parameter vector, which is estimated iteratively from the training data.

Given an *a priori* distribution, the Bayes rule can be used to determine the posterior distribution of the model parameters through $p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{t}) = p(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{t})$.

Moreover, for a new sample $\boldsymbol{x}_n$, the prediction of the corresponding label t_n can be provided by

$$p(t_n|\boldsymbol{t}) = \int p(t_n|\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{t})d\boldsymbol{w}d\boldsymbol{\alpha}d\sigma^2.$$

However, an analytical expression for the posterior distribution of the model parameters is still not available. In order to solve this problem, it is necessary to adopt an effective approximation. The posterior distribution of the parameters can be decomposed into two components according to

$$p(\boldsymbol{w}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{t}) = p(\boldsymbol{w}|\boldsymbol{t}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{t}). \tag{4}$$

The first term of the right-hand side of Eq. (4) is the posterior probability of the weights $\boldsymbol{w}$ given σ^2 and $\boldsymbol{\alpha}$. The computation of these probabilities is well detailed in [42].

Once the weights were obtained, the hyper-parameters α_i are updated according to $\alpha_i = \frac{\lambda_i}{w_i^2}$, where w_i^2 is the square of the ith average weight, λ_i is defined as $\lambda_i = 1 - \sum_{ii}$, and $\sum_{ii}$ is the ith element of the main diagonal of the covariance matrix, which may be interpreted as a measure of how well each parameter w_i is estimated. The optimization of the hyper-parameters continues until a pre-defined threshold is achieved or until certain number of iterations is performed.

Sparsity emerges when most of the α_i go to infinity, thus effectively removing the corresponding basis functions; the remaining basis functions are called the *relevance vectors* (RV) [42]. For large-scale problems, this number can be high and testing complexity might become prohibitive, namely $O(N_{\text{ts}}N_{\text{RV}})$, where N_{ts} is the number of samples in the test set and N_{RV} is the number of relevance vectors.

Standard RVM models can be used to handle classification problems with multiple classes by decomposing the problem into several binary classification tasks, each solved efficiently by an RVM model. The simplest approach, known as the *one-versus-one* approach, is to decompose the problem with C classes into $\frac{C(C-1)}{2}$ binary problems. A binary classifier is built to discriminate between each pair of classes, while discarding the rest of the classes. When testing a new sample, a voting is performed among the classifiers and the class which received the most votes is deemed to be the outcome.

3.2 Multiclass relevance vector machines

Two different types of mRVM were proposed in [18, 36], namely the constructive type (referred to as mRVM1) and the top-down type (mRVM2). The idea of both is not to train multiple RVM classifiers but to train only a single model that could deal directly with multiclass problems. While mRVM1 achieves sparsity by starting with an empty model and adding samples from the training set based on their contribution to the model, the strategy underlying mRVM2 is to follow a top-down strategy by loading the whole training kernel into memory and iteratively removing non-relevant samples.

The training phase of mRVM2 is similar to that of mRVM1, being both based on the expectation maximization (EM) algorithm. The main difference in that is the mRVM2 does not adopt the marginal likelihood maximization as mRVM1 does [see Eq. (5)] but rather employs an extra E-step for the updates of the hyper-parameters [18]. Moreover, mRVM2 is relatively more expensive than mRVM1 because each sample i has different scales α_{ic} across classes. However, if mRVM2 prunes a sample, such sample cannot be reintroduced into the model. In what follows, we present the main equations underlying the formulation of the mRVM1. The reader is referred to [36] for more detailed explanations.

Consider a training set $\{\boldsymbol{x}_n, t_n\}_{n=1}^N$, where $\boldsymbol{x}_n \in \Re^m$ and $t_n \in \{1, \ldots, C\}$. Let $\boldsymbol{k}_n$ be the nth row of the kernel matrix $\boldsymbol{K}$ ($\boldsymbol{K} \in \Re^{N\times N}$), expressing how the nth sample correlates with the others from the training set. The learning process involves the inference of the model parameters $\boldsymbol{W} \in \Re^{N\times C}$ in such a way that the quantity $\boldsymbol{W}^T\boldsymbol{K}$ acts as a sort of voting system expressing which data relationships are important to capture for increasing the model's discriminative properties.

Moreover, let $\boldsymbol{Y} = \{y_{11}, \ldots, y_{1N}; \ldots; y_{c1}, \ldots, y_{cN}; \ldots; y_{C1}, \ldots, y_{CN}\} \in \Re^{C\times N}$ denote a matrix of auxiliary variables introduced for the purpose of multiple class discrimination, acting as targets for $\boldsymbol{W}^T\boldsymbol{K}$. The variables y_{cn} are assumed to obey a standardized noise model, i.e., $y_{cn}|\boldsymbol{w}_c, \boldsymbol{k}_n \sim \mathcal{N}_{y_{cn}}(\boldsymbol{w}_c^T\boldsymbol{k}_n, 1)$, whereas the model parameters w_{nc} follow a standard zero-mean Gaussian distribution, namely $w_{nc} \sim \mathcal{N}(0, 1/\alpha_{nc})$, where α_{nc} belongs to the scaling matrix $\boldsymbol{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N)^T \in \Re^{N\times C}$.

The formulation of mRVM1 adopts as objective the maximization of the marginal likelihood $p(\boldsymbol{Y}|\boldsymbol{K}, \boldsymbol{A}) = \int p(\boldsymbol{Y}|\boldsymbol{K}, \boldsymbol{W})p(\boldsymbol{W}|\boldsymbol{A})d\boldsymbol{W}$. In order to differentiate this likelihood, Psorakis et al. [36] followed the assumption that each sample n has a common scale α_n shared across all classes. So, for mRVM1, the vector of hyper-parameters $\boldsymbol{\alpha}_n$ associated with a sample turns out to be a simple scalar α_n. The maximization of the marginal likelihood results in a criterion to either add a sample n, delete it, or update its associated α_n. So, the model can start with a single sample and then proceed in a constructive manner.

In order to achieve this goal, the log of the marginal likelihood is decomposed into contributing terms based on each sample, that is,

$$\begin{aligned}\mathfrak{L}(\boldsymbol{A}) &= \log p(\boldsymbol{Y}|\boldsymbol{K},\boldsymbol{A}) \\ &= \sum_{c=1}^{C} -\frac{1}{2}\left[N\log 2\pi + \log|\boldsymbol{\mathcal{C}}| + \boldsymbol{y}_c^T\boldsymbol{\mathcal{C}}^{-1}\boldsymbol{y}_c\right],\end{aligned} \tag{5}$$

where $\boldsymbol{\mathcal{C}} = \boldsymbol{I} + \boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^T$, whose determinant and inverse were derived by Tipping and Faul [43] as a function of $\boldsymbol{\mathcal{C}}_{-i}$, that is, the value of $\boldsymbol{\mathcal{C}}$ with the ith sample removed. The determinant of $\boldsymbol{\mathcal{C}}$ is given by

$$|\boldsymbol{\mathcal{C}}| = |\boldsymbol{\mathcal{C}}_{-i}||1 + \alpha_i^{-1}\boldsymbol{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}\boldsymbol{k}_i|,$$

whereas the inverse of $\boldsymbol{\mathcal{C}}$ is given by

$$\boldsymbol{\mathcal{C}}^{-1} = \boldsymbol{\mathcal{C}}_{-i}^{-1} - \frac{\boldsymbol{\mathcal{C}}_{-i}^{-1}\boldsymbol{k}_i\boldsymbol{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}}{\alpha_i + \boldsymbol{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}\boldsymbol{k}_i}. \tag{6}$$

Equipped with these results, Eq. (5) can be rewritten as:

$$\mathfrak{L}(\boldsymbol{A}) = \mathfrak{L}(\boldsymbol{A}_{-i}) + \sum_{c=1}^{C} -\frac{1}{2}\left[\log\alpha_{ic} - \log(\alpha_i + s_i) + \frac{q_{ci}^2}{\alpha_i + s_i}\right],$$

where s_i and q_{ci} are called sparsity factor and quality factor, respectively, and these are defined as $s_i = \boldsymbol{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}\boldsymbol{k}_i$ and $q_{ci} = \boldsymbol{k}_i^T\boldsymbol{\mathcal{C}}_{-i}^{-1}y_c$. The sparsity factor can be seen as a measure of how much the descriptive information of the ith sample is already captured from the existing samples. On the other hand, the quality factor measures how good the ith sample is in helping to describe a specific class [36].

By setting the derivative $\partial\mathfrak{L}(\boldsymbol{A})/\partial\alpha_i = 0$, one obtains

$$\alpha_i = \frac{Cs_i^2}{\sum_{c=1}^{C} q_{ci}^2 - Cs_i}, \quad \text{if } \sum_{c=1}^{C} q_{ci}^2 > Cs_i \tag{7a}$$

$$\alpha_i = \infty, \quad \text{if } \sum_{c=1}^{C} q_{ci}^2 \leq Cs_i. \tag{7b}$$

The quantity $\theta_i = \sum_{c=1}^{C} q_{ci} - Cs_i$ captures the contribution of the ith sample to the marginal likelihood in terms of how much additional descriptive information it provides to the model. By resorting to this quantity, it is possible to establish some rules for including or excluding a given sample, or updating its hyper-parameter [18]:

- IF $\theta_i > 0$ and $\alpha_i < \infty$ THEN set/update α_i with (7a);
- IF $\theta_i \leq 0$ and $\alpha_i < \infty$ THEN set α_i with (7b).

Then, the M-step and E-step of EM are used to estimate $\boldsymbol{W}$ and the posterior expectations of the auxiliary variables $\boldsymbol{Y}$, respectively. The weights are estimated as:

$$\hat{\boldsymbol{w}}_c = (\boldsymbol{K}\boldsymbol{K}^T + \boldsymbol{A}_c)^{-1}\boldsymbol{K}\tilde{\boldsymbol{y}}_c^T.$$

Assuming a given class i, the E-step calculates the expected value of y_{in} as

$$\tilde{y}_{in} = \hat{\boldsymbol{w}}_i^T\boldsymbol{k}_n - \left(\sum_{j\neq i}\tilde{y}_{jn} - \hat{\boldsymbol{w}}_j^T\boldsymbol{k}_n\right),$$

whereas $\forall c \neq i$, the E-step yields

$$\tilde{y}_{cn} \leftarrow \hat{\boldsymbol{w}}_c^T\boldsymbol{k}_n - \frac{\mathcal{E}_{p(u)}\left\{\mathcal{N}_u(\hat{\boldsymbol{w}}_c^T\boldsymbol{k}_n - \hat{\boldsymbol{w}}_i^T\boldsymbol{k}_n, 1)\boldsymbol{\Phi}_u^{n,i,c}\right\}}{\mathcal{E}_{p(u)}\left\{\boldsymbol{\Phi}_u(u + \hat{\boldsymbol{w}}_i^T\boldsymbol{k}_n - \hat{\boldsymbol{w}}_c^T\boldsymbol{k}_n)\boldsymbol{\Phi}_u^{n,i,c}\right\}},$$

where $u \sim \mathcal{N}(0,1)$ and $\boldsymbol{\Phi}$ denotes the Gaussian cumulative distribution function.

In the classification phase, the test sample $\boldsymbol{x}_n$ is labeled as of the class i whose auxiliary variable y_{in}, $1 \leq i \leq C$, is maximum, i.e., $t_n = \arg\max_i(y_{in})$.

4 Computational experiments

In what follows, we provide details about the dataset used in the experiments and how the experiments were set up. Then, we present the accuracy results revealed by the RVM and mRVM models, considering the different methods to calculate the fractal dimension. For each model, we also report the optimized kernel parameter value and the associated number of relevance vectors, so as to measure the complexity of the induced models. In this paper, the one-versus-one approach was adopted when using the standard RVM approach.

4.1 Description of the dataset

The EMG signal dataset used in our experiments was originally collected by Chan and collaborators [6, 17]. The authors used eight channels of surface EMG to collect signals from the right arm of 30 normally limbed subjects. Each subject underwent four sessions, with one to two days of separation between sessions. Each session consisted of six trials. EMG signals were collected from seven sites on the forearm and one site on the biceps. An electrode was placed on the wrist to provide a common ground reference.

Table 1 Class distribution for each trial

Trials	Classes							
	#1	#2	#3	#4	#5	#6	#7	Total
#1	105	100	100	99	99	96	105	704
#2	100	100	100	101	103	100	99	703
#3	100	98	100	105	98	100	102	703
#4	98	102	99	104	97	101	102	703
#5	95	103	96	102	99	104	103	702
#6	102	101	99	98	96	102	103	701

Table 2 Best results in terms of cross-validation error achieved for each feature type and kernel machine

Feature	Trial	SVM			RVM			mRVM1			mRVM2		
		σ	Error	#SV	σ	Error	#RV	σ	Error	#RV	σ	Error	#RV
Box Counting (BC)	#1	8.00	**35.80 ± 1.67**	349.20 ± 0.45	2.00	35.88 ± 2.51	302.60 ± 103.75	4.00	38.51 ± 5.26	145.60 ± 184.43	2.00	35.96 ± 2.57	14.80 ± 11.41
	#2	8.00	36.44 ± 1.95	349.20 ± 0.45	4.00	36.70 ± 1.99	298.20 ± 113.59	4.00	38.04 ± 5.92	145.00 ± 184.97	2.00	**35.76 ± 4.71**	14.80 ± 11.41
	#3	8.00	25.35 ± 2.11	349.40 ± 0.89	4.00	**25.23 ± 1.92**	296.00 ± 118.51	4.00	27.54 ± 3.06	147.20 ± 183.02	2.00	26.51 ± 2.94	14.40 ± 10.53
	#4	8.00	34.14 ± 1.66	349.20 ± 0.45	2.00	**34.05 ± 1.95**	303.60 ± 101.52	2.00	36.46 ± 6.52	146.80 ± 183.36	2.00	36.20 ± 6.36	14.40 ± 10.53
	#5	4.00	**37.03 ± 2.60**	349.00 ± 0.00	4.00	40.06 ± 2.06	297.40 ± 115.38	2.00	40.67 ± 6.77	145.00 ± 184.97	2.00	38.85 ± 6.47	14.80 ± 11.41
	#6	1.00	**35.77 ± 1.36**	349.00 ± 0.00	2.00	36.06 ± 3.38	302.60 ± 103.75	2.00	38.14 ± 5.48	145.80 ± 184.25	2.00	36.08 ± 5.59	16.00 ± 14.07
Higuchi (HG)	#1	4.00	**17.98 ± 1.73**	349.50 ± 0.71	4.00	19.83 ± 1.21	205.50 ± 202.94	8.00	20.34 ± 1.34	16.50 ± 3.54	2.00	20.31 ± 2.36	18.00 ± 11.31
	#2	1.00	**10.78 ± 1.85**	349.50 ± 0.71	2.00	12.91 ± 1.91	204.50 ± 204.35	2.00	13.74 ± 1.18	14.50 ± 6.36	1.00	12.88 ± 2.20	19.00 ± 12.73
	#3	4.00	**12.80 ± 1.83**	350.00 ± 1.41	2.00	14.25 ± 1.32	234.00 ± 162.63	2.00	16.59 ± 1.31	19.50 ± 0.71	2.00	14.54 ± 1.56	18.50 ± 12.02
	#4	8.00	**10.87 ± 1.05**	349.50 ± 0.71	2.00	11.41 ± 1.44	206.00 ± 202.23	4.00	12.23 ± 1.79	22.00 ± 4.24	2.00	12.66 ± 1.24	15.50 ± 7.78
	#5	8.00	**13.48 ± 1.29**	349.00 ± 0.00	2.00	14.70 ± 1.77	207.00 ± 200.82	4.00	16.32 ± 1.07	18.50 ± 0.71	2.00	16.07 ± 1.56	17.00 ± 9.90
	#6	4.00	**12.89 ± 1.73**	349.00 ± 0.00	2.00	14.35 ± 1.42	206.00 ± 202.23	4.00	14.81 ± 1.92	18.00 ± 1.41	2.00	15.61 ± 2.03	17.00 ± 9.90
Katz (KT)	#1	8.00	4.46 ± 0.61	349.33 ± 0.58	16.00	4.38 ± 1.12	348.33 ± 1.15	32.00	**3.72 ± 0.38**	239.33 ± 190.81	4.00	5.08 ± 1.06	14.00 ± 6.08
	#2	8.00	3.44 ± 0.49	349.33 ± 0.58	2.00	3.33 ± 1.33	269.33 ± 137.99	32.00	**3.13 ± 0.69**	239.33 ± 190.81	2.00	3.93 ± 0.60	14.67 ± 7.23
	#3	16.00	3.58 ± 1.02	349.67 ± 1.15	16	3.67 ± 0.95	349.67 ± 1.15	32.00	**3.13 ± 0.77**	239.00 ± 190.53	16.00	3.98 ± 0.71	14.33 ± 6.66
	#4	8.00	**3.15 ± 1.19**	349.33 ± 0.58	16.00	3.70 ± 0.75	349.33 ± 0.58	32.00	3.98 ± 0.81	239.00 ± 190.53	1.00	5.00 ± 0.97	15.33 ± 8.39
	#5	4.00	3.99 ± 0.98	349.00 ± 0.00	32.00	**3.68 ± 1.19**	349.00 ± 0.00	32.00	3.79 ± 0.48	239.00 ± 190.53	1.00	4.90 ± 0.98	16.00 ± 9.54
	#6	8.00	**3.48 ± 0.38**	349.00 ± 0.00	16.00	3.91 ± 1.04	349.00 ± 0.00	32.00	4.31 ± 1.5	238.00 ± 189.67	1.00	4.37 ± 0.94	16.00 ± 9.54
Sevcik (SV)	#1	2.00	33.97 ± 2.15	349.25 ± 0.50	2.00	**33.21 ± 1.68**	289.25 ± 119.50	2.00	36.24 ± 2.58	188.50 ± 183.80	2.00	35.80 ± 5.54	16.25 ± 11.18
	#2	8.00	**32.69 ± 2.28**	349.25 ± 0.50	4.00	34.19 ± 1.90	287.25 ± 123.50	4.00	37.10 ± 2.28	184.25 ± 188.52	2.00	37.53 ± 5.30	14.75 ± 8.18
	#3	4.00	28.42 ± 2.33	349.50 ± 1.00	4.00	**27.97 ± 0.80**	282.50 ± 133.00	2.00	33.12 ± 2.39	183.50 ± 189.38	1.00	31.47 ± 3.48	14.00 ± 6.68
	#4	8.00	**31.44 ± 2.01**	349.25 ± 0.50	2.00	31.61 ± 2.29	286.75 ± 124.50	2.00	38.06 ± 1.39	184.50 ± 188.23	2.00	35.08 ± 3.93	15.00 ± 8.68
	#5	4.00	**32.31 ± 2.50**	349.00 ± 0.00	2.00	33.08 ± 2.39	291.25 ± 115.50	2.00	38.75 ± 1.44	185.25 ± 187.38	1.00	37.79 ± 3.70	16.75 ± 12.18
	#6	1.00	32.53 ± 1.89	349.00 ± 0.00	2.00	**32.41 ± 2.28**	288.75 ± 120.50	2.00	38.83 ± 2.06	182.75 ± 192.00	2.00	35.92 ± 4.45	16.75 ± 12.18

Values in bold indicate, for each trial, the best average performance index achieved among the contestants

These signals were amplified with gain of 1000 and a bandwidth of 1 Hz to 1 KHz. Signals were sampled at 3 KHz using an analog-to-digital converter.

Seven distinct limb motions (classes) were performed: hand open, hand close, supination, pronation, wrist flexion, wrist extension, and rest. For each trial, the subject repeated each limb motion four times, holding each motion for 3 s, each time. The order of these limb motions was randomized. Chan and Green [6] only used the session four in their experiments. Data from the first two trials were used as training data, and data from the remaining four trials were used as testing data. In this paper, we also make use of data from session four, but the investigated models were assessed separately on each trial using 5 × 2 cross-validation.

4.2 Experimental setup

The main purpose of this paper is to empirically assess the performance of RVM models in the task of EMG signal classification. In the experiments, we have considered only the radial basis function kernel [39], which has an associated hyper-parameter to be calibrated beforehand, namely the radius σ. The value of this parameter was varied in our experiments. Although we know that there are several heuristics to select the values of hyper-parameters, we have opted to set the value of σ as one in the range $\{2^i, i = -3, -2, -1, 0, 1, 2, 3, 4, 5\}$. For each of the nine values in this range, a 5×2-fold cross-validation run per trial was performed in order to measure the average performance of the methods.

In what concerns data preprocessing, samples were extracted from the EMG signals using a sliding window of 256 ms in length, spaced 32 ms apart [15]. Then, the FD values, as calculated by the different methods described in Sect. 2, were used to build up the feature vectors. The dimension of each transformed sample (i.e., feature vector) was of eight features, since there were eight channels and one FD value was calculated for each channel. The class distribution for each trial is presented in Table 1.

4.3 Simulation results

In Table 2, we report the best accuracy results achieved by the different kernel machines, including SVM, considering the four types of FD features. The results are given in terms of average and standard deviation of the generalization error calculated over the 5×2-fold cross-validation process. In this table, we highlight the best calibrated kernel parameter value for each kernel machine and also present the number of relevance vectors or support vectors associated with each model. The accuracy results are complemented with those reported in Table 3, which relates to the application of the two-sided Wilcoxon rank sum test over the cross-validation errors [11]. The Wilcoxon rank sum test is a nonparametric statistical procedure that helps answering the following question: Do two independent samples, say **x** and **y**, represent two different populations? The null hypothesis is that data in **x** and **y** are samples from continuous distributions with equal medians. Assuming a 5 % significance level, having a p-value lower than 0.05 indicates that the Wilcoxon rank sum test rejects the null hypothesis, and so the difference in performance between the given kernel machines is statistically significant [20]. In our case, the test is applied per trial and one of the samples always relates to the kernel machine with the lowest average cross-validation error for the given trial.

On the other hand, Tables 4 and 5 show the specificity and sensitivity values delivered by the best calibrated kernel machines, as reported in Table 2, for each

Table 3 Results of the Wilcoxon rank sum test over the cross-validation errors

Feature	Trial	SVM	RVM	mRVM1	mRVM2
		p value	p value	p value	p value
Box counting (BC)	#1	–	0.820	0.006	0.384
	#2	0.622	0.791	0.449	–
	#3	0.791	–	0.059	0.226
	#4	0.970	–	0.008	0.021
	#5	–	0.021	0.005	0.045
	#6	–	0.910	0.003	0.104
Higuchi (HG)	#1	–	0.017	0.013	0.041
	#2	–	0.017	0.002	0.037
	#3	–	0.096	0.000	0.054
	#4	–	0.426	0.063	0.004
	#5	–	0.121	0.001	0.002
	#6	–	0.058	0.040	0.007
Katz (KT)	#1	0.014	0.058	–	0.002
	#2	0.254	0.910	–	0.021
	#3	0.363	0.254	–	0.031
	#4	–	0.272	0.089	0.003
	#5	0.405	–	0.623	0.048
	#6	–	0.149	0.159	0.016
Sevcik (SV)	#1	0.910	–	0.005	0.273
	#2	–	0.185	0.001	0.007
	#3	0.520	–	0.000	0.007
	#4	–	0.623	0.000	0.023
	#5	–	0.384	0.000	0.001
	#6	0.910	–	0.000	0.031

combination of FD method and trial. Each of the last 14 columns in these tables refers to either a specificity or a sensitivity result for a certain class. Sensitivity (also called the true positive rate) measures the proportion of actual positives of a class which are correctly identified as such, whereas specificity (aka the true negative rate) measures the proportion of negatives of a class which are correctly identified as such.

The features were normalized to have 0 as mean and 1 as standard deviation. Since the accuracy results produced by using the feature values extracted by the Katz's method were significantly better than those obtained by using the feature values extracted by the other FD methods, we decided to inspect in more detail the effect of the calibration of the kernel parameter value for the cases where the Katz's method was employed. Thus, Figs. 1, 2, 3 and 4 show the way the accuracy rate (i.e., 1—error rate) obtained by the different kernel machines has varied as a function of the kernel parameter value. Figures 5, 6, 7, and 8 do the same job but focus on the sensitivity. The choice of the trial #1 was arbitrary since the purpose here is only to contrast the profiles produced by the different machines.

Table 4 Best specificity (Spec) results achieved by models for each class and FD method using trial #1

Model	FD	σ	Spec 1	Spec 2	Spec 3	Spec 4	Spec 5	Spec 6	Spec 7
SVM	**BC**	1.00	0.66 ± 0.06	0.56 ± 0.06	0.54 ± 0.05	0.56 ± 0.04	0.74 ± 0.08	0.60 ± 0.05	0.82 ± 0.05
	HG	4.00	0.70 ± 0.08	**0.97 ± 0.03**	0.84 ± 0.06	0.73 ± 0.05	0.86 ± 0.05	0.95 ± 0.02	0.70 ± 0.05
	KT	8.00	**0.95 ± 0.03**	0.95 ± 0.02	**0.95 ± 0.02**	**0.98 ± 0.01**	**0.96 ± 0.02**	**0.96 ± 0.03**	**0.94 ± 0.03**
	SV	8.00	0.69 ± 0.05	0.62 ± 0.07	0.51 ± 0.05	0.65 ± 0.10	0.77 ± 0.06	0.71 ± 0.06	0.68 ± 0.04
RVM	**BC**	2.00	0.67 ± 0.08	0.66 ± 0.04	0.52 ± 0.06	0.60 ± 0.10	0.72 ± 0.04	0.60 ± 0.06	0.72 ± 0.04
	HG	4.00	0.64 ± 0.06	0.96 ± 0.04	0.84 ± 0.04	0.69 ± 0.06	0.83 ± 0.05	0.95 ± 0.03	0.73 ± 0.07
	KT	16.00	0.96 ± 0.03	0.94 ± 0.03	0.97 ± 0.03	0.98 ± 0.02	0.96 ± 0.02	0.95 ± 0.02	0.94 ± 0.02
	SV	2.00	0.66 ± 0.08	0.63 ± 0.07	0.55 ± 0.06	0.63 ± 0.07	0.76 ± 0.09	0.76 ± 0.06	0.71 ± 0.05
mRVM1	**BC**	2.00	0.67 ± 0.04	0.73 ± 0.09	0.42 ± 0.09	0.48 ± 0.08	0.73 ± 0.09	0.61 ± 0.10	0.65 ± 0.08
	HG	8.00	0.57 ± 0.06	0.94 ± 0.05	0.89 ± 0.04	0.69 ± 0.05	0.88 ± 0.06	0.96 ± 0.02	0.67 ± 0.08
	KT	32.00	0.95 ± 0.02	0.95 ± 0.03	0.98 ± 0.02	0.99 ± 0.02	0.97 ± 0.02	0.95 ± 0.03	0.95 ± 0.02
	SV	1.00	0.63 ± 0.05	0.59 ± 0.03	0.39 ± 0.12	0.62 ± 0.05	0.79 ± 0.07	0.72 ± 0.05	0.71 ± 0.07
mRVM2	**BC**	2.00	0.67 ± 0.06	0.70 ± 0.09	0.49 ± 0.11	0.59 ± 0.09	0.72 ± 0.05	0.61 ± 0.10	0.69 ± 0.08
	HG	2.00	0.62 ± 0.03	0.96 ± 0.03	0.85 ± 0.06	0.68 ± 0.06	0.84 ± 0.09	0.94 ± 0.02	0.70 ± 0.06
	KT	4.00	0.93 ± 0.05	0.94 ± 0.02	0.97 ± 0.02	0.98 ± 0.02	0.96 ± 0.02	0.91 ± 0.03	0.95 ± 0.03
	SV	0.50	0.64 ± 0.07	0.62 ± 0.07	0.45 ± 0.10	0.64 ± 0.05	0.76 ± 0.08	0.70 ± 0.14	0.69 ± 0.07

Table 5 Best sensitivity (Sens) results achieved by models for each class and FD method using trial #1

Model	FD	σ	Sens 1	Sens 2	Sens 3	Sens 4	Sens 5	Sens 6	Sens 7
SVM	**BC**	1.00	0.93 ± 0.02	0.92 ± 0.02	0.91 ± 0.02	0.92 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.86 ± 0.03
	HG	4.00	0.94 ± 0.02	0.98 ± 0.01	0.97 ± 0.01	0.95 ± 0.02	0.97 ± 0.01	0.99 ± 0.00	0.95 ± 0.01
	KT	8.00	0.99 ± 0.01	0.99 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.01
	SV	8.00	0.91 ± 0.02	0.91 ± 0.02	0.91 ± 0.02	0.91 ± 0.01	0.93 ± 0.02	0.97 ± 0.01	0.93 ± 0.02
RVM	**BC**	2.00	0.93 ± 0.03	0.90 ± 0.02	0.90 ± 0.02	0.91 ± 0.02	0.92 ± 0.01	0.93 ± 0.02	0.91 ± 0.03
	HG	4.00	0.94 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.94 ± 0.01	0.97 ± 0.01	0.99 ± 0.01	0.93 ± 0.02
	KT	16.00	1.00 ± 0.00	0.99 ± 0.01	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.01	0.98 ± 0.01
	SV	2.00	0.92 ± 0.02	0.92 ± 0.02	0.91 ± 0.02	0.91 ± 0.02	0.93 ± 0.02	0.96 ± 0.01	0.92 ± 0.03
mRVM1	**BC**	2.00	0.91 ± 0.02	0.89 ± 0.03	0.91 ± 0.03	0.91 ± 0.02	0.89 ± 0.02	0.90 ± 0.04	0.92 ± 0.02
	HG	8.00	0.94 ± 0.02	0.97 ± 0.01	0.96 ± 0.02	0.95 ± 0.01	0.97 ± 0.01	0.99 ± 0.01	0.94 ± 0.02
	KT	32.00	1.00 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.01	0.98 ± 0.00
	SV	1.00	0.91 ± 0.02	0.92 ± 0.02	0.92 ± 0.02	0.91 ± 0.02	0.89 ± 0.04	0.95 ± 0.02	0.89 ± 0.03
mRVM2	**BC**	2.00	0.93 ± 0.02	0.90 ± 0.04	0.91 ± 0.02	0.90 ± 0.03	0.92 ± 0.03	0.93 ± 0.03	0.90 ± 0.02
	HG	2.00	0.93 ± 0.02	0.97 ± 0.01	0.98 ± 0.01	0.95 ± 0.01	0.96 ± 0.01	0.99 ± 0.01	0.93 ± 0.02
	KT	4.00	0.99 ± 0.00	0.99 ± 0.01	0.99 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.01
	SV	0.50	0.91 ± 0.04	0.91 ± 0.05	0.92 ± 0.03	0.89 ± 0.04	0.91 ± 0.03	0.96 ± 0.01	0.90 ± 0.02

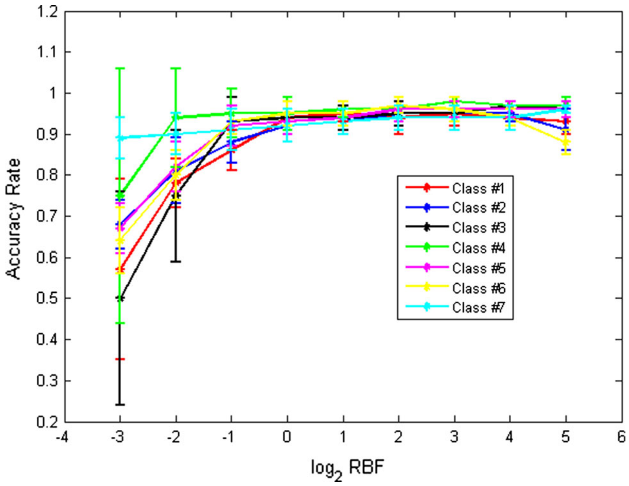

Fig. 1 Variation of accuracy rate per class as a function of the kernel parameter value for SVM using trial #1 and feature vector extracted through the Katz's method

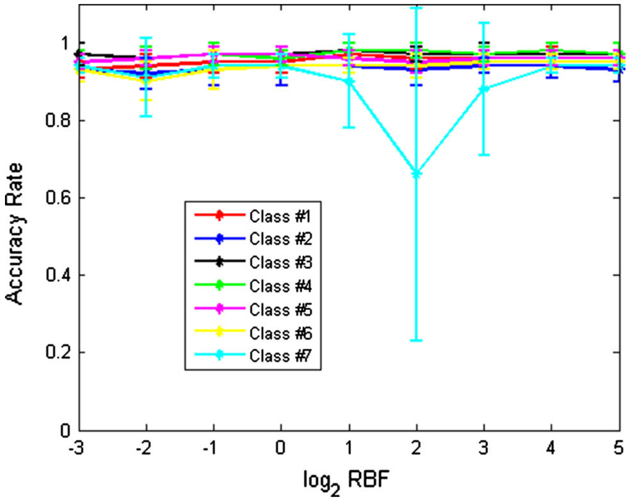

Fig. 2 Variation of accuracy rate per class as a function of the kernel parameter value for RVM using trial #1 and feature vector extracted through the Katz's method

The bars in Figs. 1, 2, 3, and 4 represent the variance in accuracy rate per class (one standard deviation from the mean) for each value of σ considered.

Finally, in Tables 6 and 7, we provide the average processing time elapsed during the training and testing phases for each combination of classifier model, fractal dimension estimation method, and experimental trial.

4.4 Discussion

From the results presented in Tables 2 and 3, it is possible to conclude that, in general, the accuracy rates displayed by SVM and RVM were rather similar to each other, prevailing in the majority of the cases over those produced by mRVM2. On the other hand, the performance of mRVM1 varies in accordance with the feature extractor adopted. Considering specifically the BC and Sevcik's methods, SVM and RVM usually outperformed the others, as testified by the low p-values associated with mRVM1 and mRVM2. For Higuchi's method, SVM performed consistently better than mRVM1 and mRVM2, but was comparable to RVM in most of the cases. On the other hand, irrespective of the type of kernel machine, the accuracy rates obtained with the aforementioned FD methods were significantly worse than those achieved with

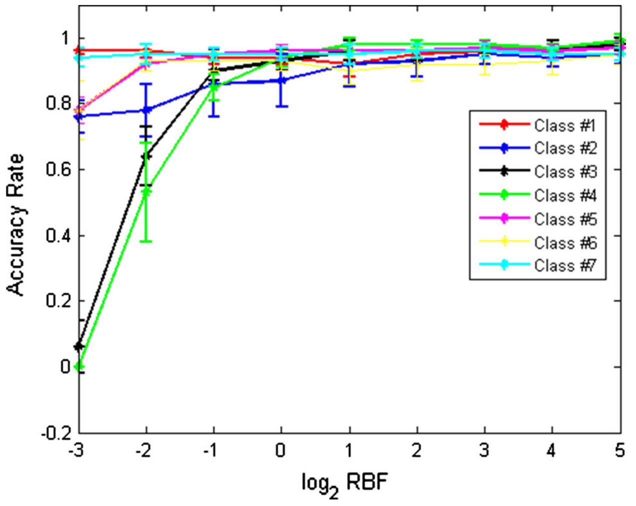

Fig. 3 Variation of accuracy rate per class as a function of the kernel parameter value for mRVM1 using trial #1 and feature vector extracted through the Katz's method

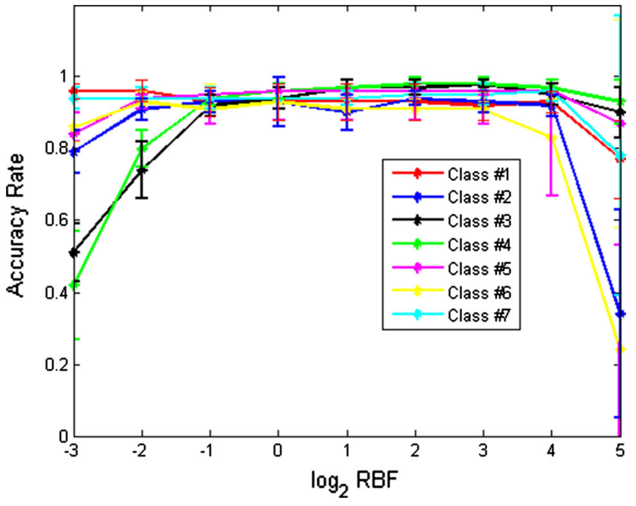

Fig. 4 Variation of accuracy rate per class as a function of the kernel parameter value for mRVM2 using trial #1 and feature vector extracted through the Katz's method

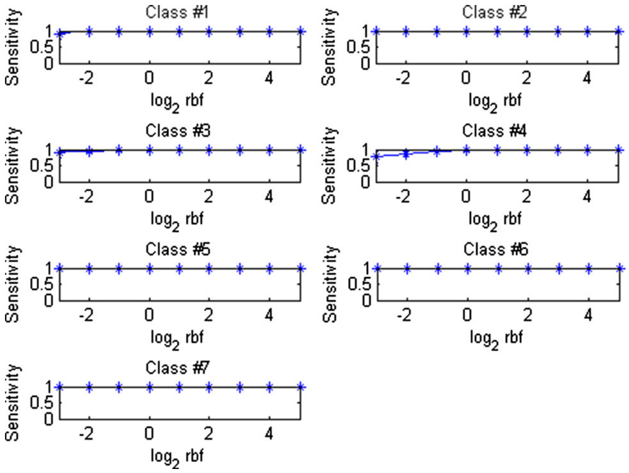

Fig. 5 Variation of sensitivity per class as a function of the kernel parameter for SVM using trial #1 and feature vector extracted through the Katz's method

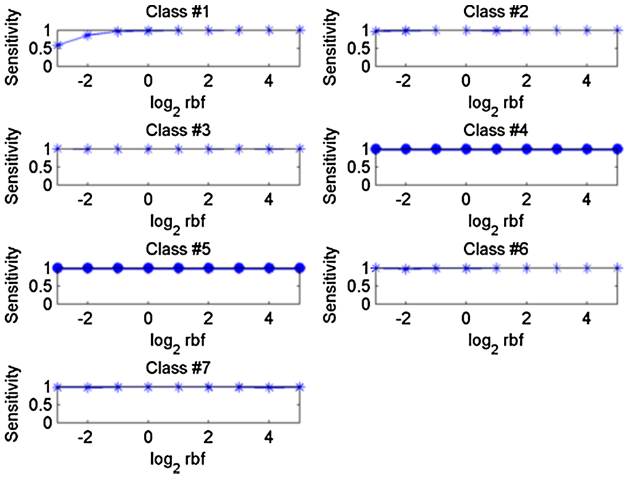

Fig. 7 Variation of sensitivity per class as a function of the kernel parameter for mRVM1 using trial #1 and feature vector extracted through the Katz's method

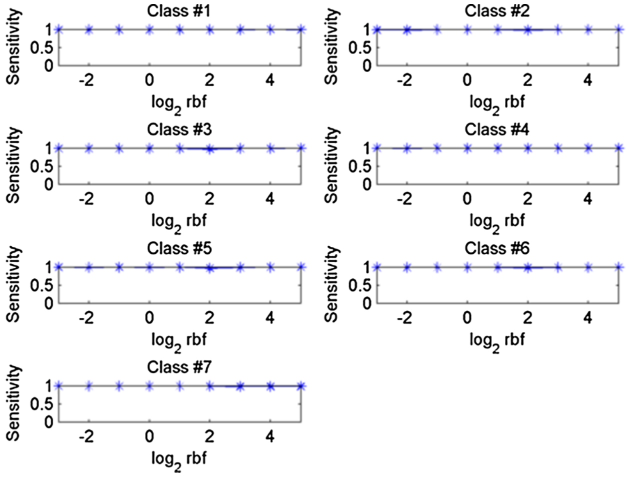

Fig. 6 Variation of sensitivity per class as a function of the kernel parameter for RVM using trial #1 and feature vector extracted through the Katz's method

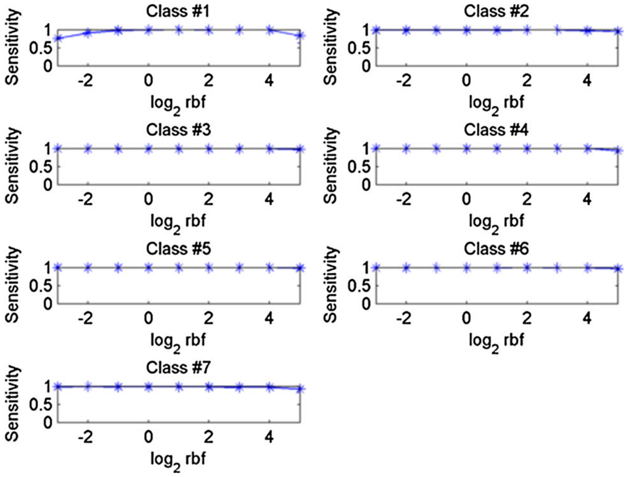

Fig. 8 Variation of sensitivity per class as a function of the kernel parameter for mRVM2 using trial #1 and feature vector extracted through the Katz's method

the Katz's method. For this feature type, the performance levels delivered by SVM, RVM, and mRVM1 were rather comparable, since the null hypothesis could not be rejected in five out of six trials. In half of the trials, mRVM1 has provided the best average results, whereas in all cases, mRVM2 was overmatched by the best kernel machine. It is also worth mentioning that the standard deviation values of the error rates obtained with the Katz's method were usually smaller for all machines, evidencing the robustness of the induced models to the variability of training/test data in the cross-validation process.

In what concerns the efficiency of FD-based RVM and its variations in terms of computational time, the results shown in Tables 6 and 7 reveal that the training of these models is usually more expensive than that of SVM. However, in the testing phase, this time reduced from 2 s on average for SVM to circa 1.5 s on average for RVM and to about 0.4 s on average for mRVM1 and mRVM2. This suggests that the FD-based multiclass RVM can yield more sparse solutions, which means a better data reduction ability. Anyway, regardless of the FD estimation technique used, the time taken to obtain the final classification outputs from the induced RVM models is usually small, which ensures their practical deployment in real-world settings.

By looking at the values shown in Tables 4 and 5, one can perceive that the use of the Katz's method as feature extractor has endowed all classifiers with the capability to provide a good balance between specificity and sensitivity

Table 6 Average CPU time (in seconds) spent in the training phase for each combination of FD estimation method, classifier type, and experimental trial

Model	FD	Trial #1	Trial #2	Trial #3	Trial #4	Trial #5	Trial #6	Average #
SVM	**BC**	83.874	78.230	75.913	82.057	82.651	80.030	80.459
	HG	70.342	59.462	62.795	61.826	61.448	61.623	62.916
	KT	59.030	56.776	59.107	58.316	59.723	58.199	58.525
	SV	76.657	77.458	75.644	78.977	80.005	79.341	78.014
RVM	**BC**	395.237	389.891	382.572	381.816	394.904	388.657	388.846
	HG	372.106	387.296	367.190	376.485	374.384	372.156	374.936
	KT	291.536	319.270	258.482	256.289	239.642	244.999	268.370
	SV	395.939	404.915	386.840	397.688	401.969	396.744	397.349
mRVM1	**BC**	364.851	389.985	316.299	360.709	318.953	346.452	349.542
	HG	328.064	353.138	298.344	325.205	285.865	307.457	316.346
	KT	319.234	384.032	480.383	379.395	358.734	395.872	386.275
	SV	377.182	371.101	323.581	398.207	340.964	335.092	357.688
mRVM2	**BC**	383.436	374.228	385.177	379.760	356.889	342.905	370.399
	HG	383.326	420.473	420.152	403.707	385.867	412.833	404.393
	KT	524.064	498.941	509.644	563.858	529.767	491.003	519.546
	SV	357.942	388.900	361.419	419.416	362.607	382.724	378.835

Table 7 Average CPU time (in seconds) spent in the test phase for each combination of FD estimation method, classifier type, and experimental trial

Model	FD	Trial #1	Trial #2	Trial #3	Trial #4	Trial #5	Trial #6	Average #
SVM	**BC**	2.397	2.395	2.404	2.418	2.395	2.386	2.399
	HG	2.390	2.385	2.388	2.388	2.386	2.386	2.387
	KT	2.393	2.386	2.396	2.391	2.393	2.395	2.392
	SV	2.395	2.396	2.393	2.399	2.397	2.390	2.395
RVM	**BC**	1.508	1.503	1.519	1.505	1.516	1.510	1.510
	HG	1.519	1.514	1.525	1.521	1.518	1.526	1.521
	KT	1.535	1.528	1.535	1.523	1.534	1.527	1.530
	SV	1.511	1.501	1.523	1.507	1.506	1.510	1.510
mRVM1	**BC**	0.402	0.403	0.403	0.401	0.401	0.401	0.402
	HG	0.418	0.416	0.416	0.415	0.414	0.415	0.416
	KT	0.400	0.401	0.404	0.399	0.400	0.402	0.401
	SV	0.403	0.402	0.402	0.404	0.403	0.401	0.402
mRVM2	**BC**	0.406	0.404	0.404	0.405	0.402	0.403	0.404
	HG	0.424	0.420	0.423	0.427	0.421	0.422	0.423
	KT	0.428	0.425	0.424	0.422	0.426	0.419	0.424
	SV	0.406	0.404	0.407	0.405	0.414	0.405	0.407

of the classes. In fact, for all FD methods but Katz's, the specificity values were usually significantly lower than the sensitivity values. Besides, as evidenced in Figs. 5, 6, 7, and 8, very high sensitivity values could be obtained for all seven classes, irrespective of the value used for the kernel parameter. This behavior could not be reproduced by the other FD methods.

The choice of the kernel parameter value was not a crucial factor to distinguish between the overall best error rates exhibited by the models, even though for each kernel machine, there are some values of σ that appear more frequently in Table 2, such as $\sigma = 8$ for SVM and $\sigma = \{2, 4\}$ for RVM. As depicted in Figs. 1, 2, 3, and 4, there is usually a range of values for the kernel parameter yielding quite interesting results, although there is no optimal value yielding 100 % of correct classification for all classes. Interestingly, the best values of σ for the combination of mRVM1 and Katz's method were always the same, namely $\sigma = 32$, the highest value of the studied range. Maybe higher values of this parameter could yield even better results for the mRVM1. In terms of stability, RVM models were usually more robust to the choice of σ, considering the mean accuracy over all classes altogether.

Finally, in what regards the complexity of the induced models, the number of support vectors and relevance vectors of the best calibrated SVM and RVM models was usually significantly higher than the number of RV associated with mRVM models—refer to Table 2. An exception occurs for the combination of mRVM1 and Katz's method. In this case, the number of RV was much higher than those obtained by using the other methods for calculating the FD. On the other hand, the models induced by mRVM2 were always the less complex ones, regardless of the FD method. So, when the sparsity of the induced model is a key aspect to take into account, the use of mRVM2 seems to be much recommended.

5 Concluding remarks

In this paper, we investigated the potentials of using relevance vector machines (both in the standard and multiclass formulations) to cope with the task of EMG signal classification. In this study, we have considered different methods for calculating the fractal dimension of 1D signals as feature extractors.

Through experiments conducted on a publicly available dataset involving different types of limb movements (seven classes in total), we have empirically confirmed that the deployment of the kernel machines equipped with the FD feature values can be useful for achieving good levels of classification performance. In particular, the combination of SVM, RVM, and mRVM1 with Katz's method was the best, across the different experiment trials, in terms of accuracy and generalization. In what concerns the complexity issue, however, mRVM2 has consistently produced more sparse models, implying higher efficiency when classifying large batches of novel samples.

As ongoing work, we are currently extending the scope of investigation by considering other nonlinear dynamics methods to extract the hidden information in the EMG signals, such as the Lyapunov exponent, and Hurst exponent [2]. As future work, we plan to investigate the impact of using EMG sub-segments of different sizes and also of using different feature selection methods, since feature selection is a preprocessing step that can bring about gains in terms of classifier accuracy [23, 45]. Finally, the combination of different kernel machines in heterogeneous committee machines will also be researched in the context of EMG signal classification.

Acknowledgments The first and second authors acknowledge the sponsorship from the Brazilian National Council for Research and Development (CNPq) via grants #475406/2010-9, #308816/2012-9, and #304603/2012-0. The third author thanks the financial support of São Paulo Research Foundation (FAPESP/ Brazil)—process number 2011/04608-8.

References

1. Abry P, Gonçalves P, Véhel JL (eds) (2013) Scaling, fractals and wavelets. Wiley, New York
2. Acharya UR, Ng EYK, Swapna G, Michelle YSL (2011) Classification of normal, neuropathic, and myopathic electromyograph signals using nonlinear dynamics method. J Med Imag Health Inform 1:375–380
3. Ancillao A, Galli M, Rigoldi C, Albertini G (2014) Linear correlation between fractal dimension of surface EMG signal from rectus femoris and height of vertical jump. Chaos Solitons Fractals 66:120–126
4. Barnsley M (1988) Fractals everywhere. Academic Press, New York
5. Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin
6. Chan AD, Green GC (2007) Myoelectric control development toolbox. In: Proceedings of 30th conference of the Canadian medical & biological engineering society
7. Chan FH, Yang YS, Lam FK, Zhang YT, Parker PA (2000) Fuzzy EMG classification for prosthesis control. IEEE Trans Rehabil Eng 8:305–311
8. Chang GC, Kang WJ, Luh JJ, Cheng CK, Lai JS, Chen JJJ, Kuo TS (1996) Real-time implementation of electromyogram pattern recognition as a control command of man–machine interface. Med Eng Phys 18(7):529–537
9. Chu JU, Moon I, Lee YJ, Kim SK, Mun MS (2007) A supervised feature-projection-based real-time EMG pattern recognition for multifunction myoelectric hand control. IEEE-ASME Trans Mechatron 12:282–290
10. Damoulas T, Girolami M, Ying Y, Campbell C (2008) Inferring sparse kernel combinations and relevance vectors: An application to subcellular localization of proteins. In: Proceedings of the 7th International Conference in Machine Learning Applications, pp 577–582
11. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
12. Dobrowolski AP, Wierzbowski M, Tomczykiewicz K (2012) Multiresolution MUAPs decomposition and SVM-based analysis in the classification of neuromuscular disorders. Comput Meth Prog Bio 107:393–403
13. Easwaramoorthy D, Uthayakumar R (2011) Improved generalized fractal dimensions in the discrimination between healthy and epileptic EEG signals. J Comput Sci 2:31–38
14. Eke A, Herman P, Kocsis L, Kozak LR (2002) Fractal characterization of complexity in temporal physiological signals. Physiol Meas 23:1–38
15. Englehart K, Hudgins B, Chan ADC (2003) Continuous multifunction myoelectric control using pattern recognition. Technol Disabil 15(2):95–103
16. Englehart K, Hudgins B, Parker P (2001) A wavelet-based continuous classification scheme for multifunction myoelectric control. IEEE Trans Biomed Eng 48(3):302–311
17. Goge A, Chan A (2004) Investigating classification parameters for continuous myoelectrically controlled prostheses. In: Proceedings of the 28th conference of the Canadian medical & biological engineering society, pp 141–144
18. He W, Yow KC, Guo Y (2012) Recognition of human activities using a multiclass relevance vector machine. Opt Eng 51:017,202
19. Higuchi T (1988) Approach to irregular time series on the basis of the fractal theory. Phys D 31(2):277–283
20. Hollander M, Wolfe DA (1999) Nonparametric statistical methods. Wiley, New York
21. Hu X, Wang Z, Ren X (2005) Classification of surface EMG signal using relative wavelet packet energy. Comput Meth Prog Biomed 79:189–195

22. Hu X, Wang ZZ, Ren XM (2005) Classification of surface EMG signal with fractal dimension. J Zhejiang Univ Sci B 6:844–848
23. Huang H, Xie HB, Guo JY, Chen HJ (2012) Ant colony optimization-based feature selection method for surface electromyography signals classification. Comput Biol Med 42:30–38
24. Hudgins B, Parker P, Scott RN (1993) A new strategy for multifunction myoelectric control. IEEE Trans Biomed Eng 40(1):82–94
25. Janjarasjitt S (2014) Examination of the wavelet-based approach for measuring self-similarity of epileptic electroencephalogram data. J Zhejiang Univ Sci C 15:1147–1153
26. Kang WJ, Cheng CK, Lai JS, Shiu JR, Kuo TS (1996) A comparative analysis of various EMG pattern recognition methods. Med Eng Phys 18(5):390–395
27. Katz M (1988) Fractals and the analysis of waveforms. Comput Biol Med 18(3):145–156
28. Khokhar ZO, Xiao ZG, Menon C (2010) Surface EMG pattern recognition for real-time control of a wrist exoskeleton. Biomed Eng Online 9:41
29. Lima CAM, Coelho ALV (2011) Kernel machines for epilepsy diagnosis via EEG signal classification: a comparative study. Artif Intell Med 53:83–95
30. Lima CAM, Coelho ALV, Chagas S (2009) Automatic EEG signal classification for epilepsy diagnosis with relevance vector machines. Expert Syst Appl 36:10054–10059
31. Lucas MF, Gaufriau A, Pascual S, Doncarli C, Farina D (2008) Multi-channel surface EMG classification using support vector machines and signal-based wavelet optimization. Biomed Signal Proces Control 3:169–174
32. Najarian K, Splinter R (2012) Biomedical signal and image processing, 2nd edn. CRC Press, Boca Raton
33. Nussbaum M A, Yassierli (2003) Assessment of localized muscle fatigue furing low-moderate static contractions using the fractal dimension of EMG. In: Proceedings of the XVth triennial congress of the international ergonomics association, Seoul, Korea, August 25–29
34. Phinyomark A, Phukpattaranont P, Limsakul C (2012) Fractal analysis features for weak and single-channel upper-limb EMG signals. Expert Syst Appl 39:11156–11163
35. Phinyomark A, Quaine F, Charbonnier S, Serviere C, Tarpin-Bernard F, Laurillau Y (2014) Feature extraction of the first difference of EMG time series for EMG pattern recognition. Comput Methods Programs Biomed 117:247–256
36. Psorakis I, Damoulas T, Girolami MA (2010) Multiclass relevance vector machines: sparsity and accuracy. IEEE Trans Neural Netw 21(10):1588–1598
37. Riillo F, Quitadamo L, Cavrinia F, Gruppioni E, Pinto C, Pastò NC, Sbernini L, Albero L, Saggio G (2014) Optimization of EMG-based hand gesture recognition: supervised vs. unsupervised data preprocessing on healthy subjects and transradial amputees. Biomed Signal Process Control 14:117–125
38. Sarkar M, Leong TY (2003) Characterization of medical time series using fuzzy similarity-based fractal dimensions. Artif Intell Med 27:201–222
39. Scholköpf B, Smola A (2002) Learning with kernels. MIT Press, Cambridge
40. Sevcik C (1998) A procedure to estimate the fractal dimension of waveforms. Complex Int 5:1–19
41. Subasi A (2013) Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. Comput Biol Med 43:576–586
42. Tipping M (2001) Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 1:211–244
43. Tipping M, Faul A (2003) Fast marginal likelihood maximisation for sparse bayesian models. In: Proceedings of 9th AISTATS workshop, pp 3–6
44. Tricot C (1995) Curves and Fractal Dimension. Springer, New York
45. Yana Z, Wanga Z, Xieb H (2008) The application of mutual information-based feature selection and fuzzy LS-SVM-based classifier in motion classification. Comput Meth Prog Biomed 90:275–284
46. Yousefi J, Hamilton-Wright A (2014) Characterizing EMG data using machine-learning tools. Comput Biol Med 51:1–13