NEUROCOMPUTING, vol. 69, p. 2180 - 2192 (2006)

Building meaningful representations for nonlinear modeling of 1D- and 2D-signals: applications to biomedical signals

R. Dubois, B. Quenet*, Y. Faisandier**, G. Dreyfus**

** ESPCI-Paristech, Laboratoire d'Electronique,*
10 rue Vauquelin, 75005 Paris, France

*** ELA Medical*
C.A. La Boursidière,
92357 Le Plessis-Robinson Cedex, France

Abstract

The paper addresses two problems that are frequently encountered when modeling data by linear combinations of nonlinear parameterized functions. The first problem is *feature selection*, when features are sought as functions that are *nonlinear in their parameters* (e.g. Gaussians with adjustable centers and widths, wavelets with adjustable translations and dilations, etc.). The second problem is the design of an *intelligible representation* for 1D- and 2D- signals with peaks and troughs that have a definite meaning for experts.

To address the first problem, a generalization of the Orthogonal Forward Regression method is described. To address the second problem, a new family of nonlinear parameterized functions, termed *Gaussian mesa functions*, is defined. It allows the modeling of signals such that each significant peak or trough is modeled by a single, identifiable function. The resulting representation is sparse in terms of adjustable parameters, thereby lending itself easily to automatic analysis and classification, yet it is readily intelligible for the expert. An application of the methodology to the automatic analysis of electrocardiographic (Holter) recordings is described. Applications to the analysis of neurophysiological signals and EEG signals (early detection of Alzheimer's disease) are outlined.

Keywords: signal modeling; nonlinear regression; orthogonal forward regression; feature selection; Holter; electrocardiography; electroencephalography; EEG; ECG

1. Introduction

Modeling a signal by a family of parameterized functions is particularly useful in a variety of fields such as pattern recognition, feature extraction, classification or modeling. It is a straightforward way of performing information compression: the finite set of parameters of the modeling function may be a sparse representation of the signal of interest.

Typical families of parameterized functions used for modeling are polynomials, wavelets, radial basis functions, neural networks, etc. For a given modeling problem, the choice between those families is based on such criteria as implementation complexity, sparsity, number of variables of the quantity to be modeled, domain knowledge. The latter factor is actually the driving force behind the methodology described in the present paper.

More specifically, the scope of this article is twofold: first, we address the problem of *feature selection*, i.e. the problem of finding the most appropriate set of functions within a given family of functions that are *nonlinear in their parameters*; the solution that we describe here is generic. The second purpose is more application-specific: the design of a *meaningful representation* for 1-D or 2-D signals that exhibit bumps and/or troughs having specific meanings for the domain expert, i.e. the problem of finding a representation such that each bump or trough is modeled by a single, uniquely identifiable function. The intelligibility of the representation by the expert is especially important in the field of biological signal analysis: an application of our method to anomaly detection from electrocardiographic recordings is described (1D-signals), and an application to the modeling of time-frequency maps of electrophysiology and electro-encephalography recordings (2D-signals) is outlined.

The first part of the paper is devoted to the description of Generalized Orthogonal Regression (GOFR), an extension of the powerful Orthogonal Forward Regression (OFR) method of modeling by parameterized functions that are linear with respect to their parameters. We show that OFR can be extended to modeling by functions that are nonlinear with respect to their parameters. We show that GOFR overcomes some important limitations of traditional OFR.

In the second part of the paper, we define *Gaussian mesa functions,* which are shown to be especially appropriate for modeling signals that exhibit positive and negative peaks, in such a way that each peak can be appropriately modeled by a single mesa function.

Finally, we describe an application of the methodology to the automatic analysis of long-term electrocardiographic recordings (Holter recordings). We first show how each positive or negative peak can be modeled by a single mesa function. Then we show how each function

can be labeled, automatically and unambiguously, with the labels used routinely by experts, and how automatic discrimination between two types of heartbeats can be performed with that signal representation. As a final illustration, we outline an application of the methodology to time-frequency maps from electrophysiological and electroencephalographic recordings.

2. Orthogonal Forward Regression for feature selection

2.1. The feature selection problem

Let g_γ be a parameterized function and γ the vector of its parameters. Let $\Omega = \{g_\gamma\}_{\gamma \in \Gamma}$ be a family of such functions, where Γ is the set of the parameters. Note that the cardinality of Ω can be either finite or infinite.

Modeling a function f ($f \in L^2(\mathbb{R})$) with M functions, chosen from Ω, consists of finding a function $\tilde{f}$ that is a linear combination of M functions of Ω such that the discrepancy e_M between f and $\tilde{f}$ is as small as possible:

$$f = \sum_{\substack{i=1 \\ g_{\gamma_i} \in \Omega}}^{M} \alpha_i g_{\gamma_i} + e_M \qquad (1)$$

That problem amounts to estimating M parameter vectors $\{\gamma_i\}_{i=1..M}$ and M scalar parameters $\{\alpha_i\}_{i=1..M}$ to construct $\tilde{f}$. It can be solved in two steps:

- a *feature selection* step: in the set Ω, find the subset of M functions that are most relevant to the modeling of the signal of interest (see for instance [9], [16]),
- an *optimization* step: find the parameters of the functions selected as relevant features at the previous step.

2.1.1. Optimization

In the optimization step, $\{\gamma_i, \alpha_i\}_{i=1..M}$ are estimated from training data, i.e. specific values $\{x_k\}_{k=1..N}$ of the variable (or vector of variables), for which measurements f_k of the signal were performed; the measurements are assumed to have additive zero-mean noise ε_k: $f_k = f(x_k) + \varepsilon_k$. The set $\{(x_k, f_k)\}_{k=1..N}$ is called the training set.

The least squares cost function J is defined as:

$$J = \sum_{k=1}^{N} \left(f_k - \tilde{f}(x_k) \right)^2 = \sum_{k=1}^{N} \left(f_k - \sum_{\substack{i=1 \\ g_{\gamma_i} \in \Omega}}^{M} \alpha_i g_{\gamma_i}(x_k) \right)^2 \quad (2)$$

Equation (2) can be also written in the following form, highlighting the modeling error $e_M(x_k)$ and the measurement noise ε_k:

$$J = \sum_{k=1}^{N} \left(\left(f_k - f(x_k) \right) + \left(f(x_k) - \tilde{f}(x_k) \right) \right)^2 = \sum_{k=1}^{N} \left(\varepsilon_k + e_M(x_k) \right)^2 \quad (3)$$

The optimal model in the least squares sense $\tilde{f}$ is obtained by minimizing function J with respect to its parameters:

$$\tilde{f} = \sum_{\substack{i=1 \\ g_{\gamma_i} \in \Omega}}^{M} \alpha_i g_{\gamma_i} \quad (4)$$

with $J\left(\{\alpha_i, \gamma_i\}_{i=1,\ldots M} \right) = \min_{\alpha \in R, \gamma \in \Gamma} \left(J(\{\alpha, \gamma\}) \right)$.

2.1.2. Feature selection

The minimization of J is a multivariable nonlinear optimization problem, which is usually solved by iterative algorithms such as the BFGS algorithm or the Levenberg-Marquardt algorithm (see for instance [12], [15]). Being iterative, those algorithms require the choice of initial values of the parameters $\{\alpha_i, \gamma_i\}_{i=1,\ldots M}$. Therefore, prior to the optimization step, the number M of functions must be chosen, together with the initial values of the M parameter vectors $\{\gamma_i\}$ and of the parameters $\{\alpha_i\}$.

For functions that are local in space, such as Gaussians, random initialization of the parameters (centers and variances) is not recommended, because many random initializations and optimizations may be required in order to find a satisfactory model. In such a case, a frequent strategy consists in choosing one Gaussian per observation of the training set, centered on that point in input space, and with arbitrary variance [14]. The main shortcoming of the above initialization is the fact that the number of selected functions (M) is not optimal: it is related to the number of examples, which may have no relation whatsoever to the complexity of the data to be modelled. The Least-Squares Support Vector Machine (LS-SVM, also known as Ridge SVM) [5] starts with one function per example, and performs a selection

depending on the complexity of the margin boundary, but the parameters of the RBF functions are identical for all examples, and they are kept fixed during training.

The following three-step method, suggested in [4] for RBF functions and in [13] for wavelets, was designed to overcome those difficulties:

- generate a subset *D* of Ω, of finite size, called library,
- select *M* functions from *D* by an orthogonalization method based on the Gram-Schmidt algorithm [3]. This step is called the *selection* step; it is similar to Orthogonal Matching Pursuit ([11], [20]).
- initialize the optimization of *J* with the parameters $\{\gamma_i\}$ of those *M* selected functions, and the values $\{\alpha_i\}$ computed during the Gram-Schmidt selection step.

The final step consists in minimizing *J* with respect to the parameters thus initialized, as described in section 2.1.1.

If the model is linear in its parameters, the first two steps, called Orthogonal Forward Regression (OFR) or Orthogonal Matching Pursuit ([11]), are sufficient for constructing the model. OFR is described in detail in Appendix 1.

2.2. Limitation of the OFR procedure for feature selection

The OFR methodology presented above has been shown to be effective for modeling data with Radial Basis Functions [4] and wavelets [13]. However, the choice of the library *D* remains critical for a good optimization, and, in general, its size must be large in order to sample the whole space of parameters.

The main limitation of the algorithm can be illustrated as follows: assume that the function *f* to be modeled actually belongs to the set Ω of functions within which the model is sought (in such a case, the model is said to be "true"). Theoretically, only one function of Ω is sufficient for modeling *f*: function *f* itself. Further assume that *f* happens to belong to the library *D* of candidate functions. The Gram-Schmidt procedure will then select that function as the first function of the model, and the optimization of the parameters will be useless: one will have $\|e_1\| = 0$.

Conversely, if function *f* does not belong to the library *D*, the procedure will select *M* functions for modeling *f*, and the subsequent minimization of *J* will generate a model in which

the M functions will all be relevant: one will thus have built a model of a function of Ω with M functions of Ω, whereas a single function of Ω would have been sufficient for modeling f.

Traditionally, that problem is alleviated by building a very large library of candidate functions [6], so that, with high probability, the first selected function is very close to f in parameter space; then the optimization step brings that function even closer to f, and cancels the weights of the M-1 additional functions selected for the model. However, it has been shown [7] that selecting M functions in a library that contains N_d functions, with N examples, requires $O(M^3+M^2N_dN)$ operations: the computation time generated by large libraries hampers the efficiency of such a method to a large extent.

Actually, that problem can be traced to the fact that, in the procedure that was described in the previous section, the selection and optimization steps are distinct. In the following section, a procedure that merges those two steps, called GOFR (Generalized Orthogonal Forward Regression), is described: essentially, the method consists in "tuning" the function just after its selection, before any subsequent orthogonalization.

3. Generalized Orthogonal Forward Regression (GOFR)

Each iteration of the GOFR algorithm consists of 4 steps (Figure 1):

1. selection of the most relevant function g_γ of D;
2. "tuning" (optimization) of g_γ (this step is the main difference between OFR and GOFR);
3. orthogonalization of f;
4. orthogonalization of the library D.

3.1. Iteration $n = 1$

1) Selection of the most relevant function g_{γ_1} in ${}^1D = D$

As in the OFR procedure (see appendix 1), the function g_{γ_1} that has the smallest angle with f in observation space is selected:

$$\cos^2\left(f, g_{\gamma_1}\right) = \max_{g_\gamma \in D}\left(\cos^2\left(f, g_\gamma\right)\right) \quad \text{where} \quad \cos^2(f, g_\gamma) = \left(\frac{\langle f, g_\gamma \rangle}{\|f\|\|g_\gamma\|}\right)^2 \tag{5}$$

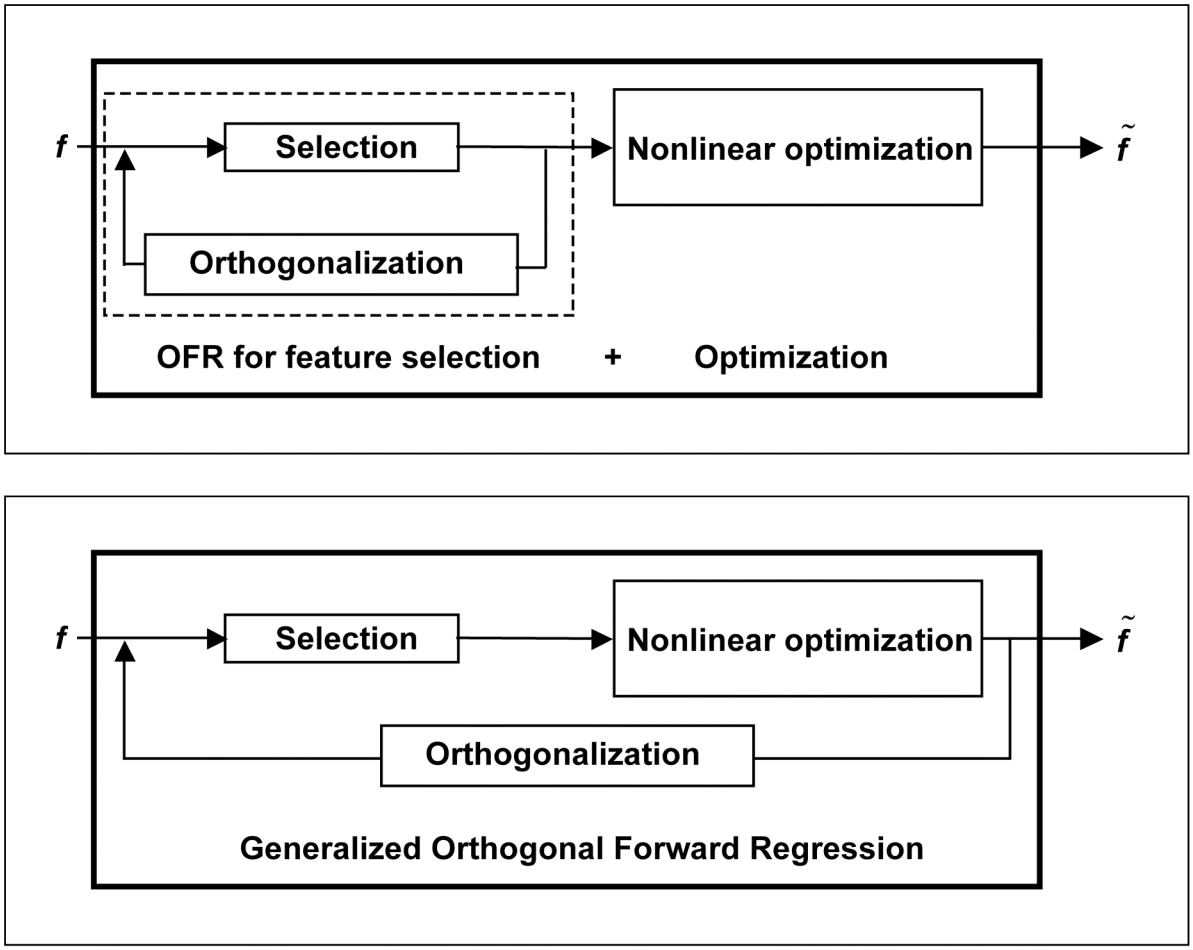


Figure 1

Top: conventional OFR for feature selection followed by nonlinear optimization;

bottom: Generalized Orthogonal Forward Regression (GOFR)

The model $\tilde{f}$ is built:

$$\tilde{f} = \alpha_1 g_{\gamma_1} \quad \text{where } \alpha_1 = \left\langle f, g_{\gamma_1} \right\rangle \tag{6}$$

2) Tuning of the selected function g_{γ_1}

Tuning g_{γ_1} consists in estimating its parameters γ_1 in order to minimize the modeling error e_1. That estimate is computed on the training set by minimizing the mean square error J:

$$J(\gamma_1, \alpha_1) = \sum_{i=1}^{N} \left(f_i - \tilde{f}(x_i) \right)^2 = \sum_{i=1}^{N} \left(f_i - \alpha_1 g_{\gamma_1}(x_i) \right)^2 \tag{7}$$

Note that this optimization problem involves only the parameters pertaining to g_{γ_1}, and α_1, so that a solution is found with a small amount of computation. Let (γ_1^*, α_1^*) be that solution. The first function of the model is thus $g_{\gamma_1^*}$ and the first parameter is α_1^*. The model at the first iteration is thus:

$$\tilde{f} = \alpha_1^* g_{\gamma_1^*} \tag{8}$$

We denote $u_1 = g_{\gamma_1^*}$

3) Orthogonalization of f (Figure 2)

As in the OFR algorithm, orthogonal projections onto the null subspace of the first selected function $u_1 = g_{\gamma_1^*}$ are performed:

$$r_2 = f - \langle f, u_1 \rangle u_1 \tag{9}$$

r_2 is thus in the null space of $g_{\gamma_1^*}$.

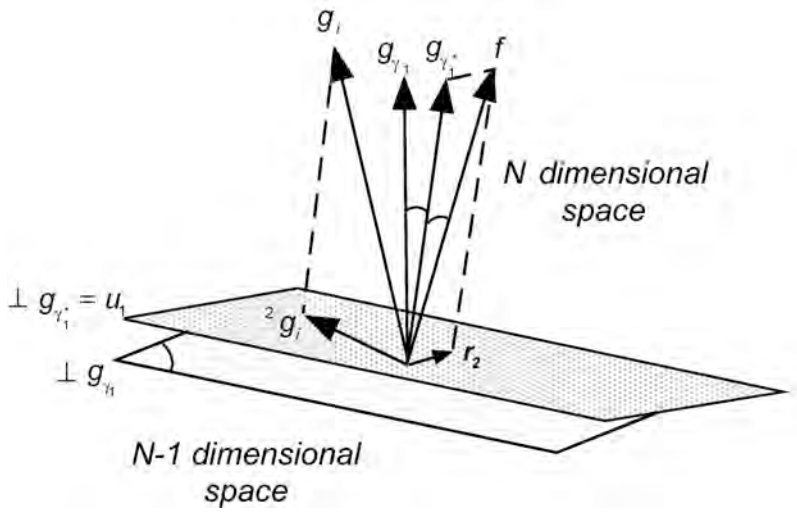


Figure 2

Orthogonalization with respect to $u_1 = g_{\gamma_1^*}$

4) Orthogonalization of $^1D = D$

A new set ^2D is computed, in the null space of u_1:

$$^2D = \left\{ \; ^2g_\gamma = {}^1g_\gamma - \langle {}^1g_\gamma, u_1 \rangle u_1 \;\; , \;\; {}^1g_\gamma \in {}^1D \right\} \tag{10}$$

3.2. Iteration *n*

When iteration *n* starts, functions $\left(g_{\gamma_1^*}, g_{\gamma_2^*}, ..., g_{\gamma_{n-1}^*}\right)$ are selected, and the orthogonal family $\left(u_1, u_2, ..., u_{n-1}\right)$ is built such that $\text{span}(g_{\gamma_1^*}, g_{\gamma_2^*}, ..., g_{\gamma_{n-1}^*}) = \text{span}(u_1, u_2, ..., u_{n-1})$.

Function r_n is in the null space of the space generated by $\text{span}(u_1, u_2, ..., u_{n-1})$, and the set ${}^nD$ is available, built as follows:

$${}^nD = \left\{ {}^ng_\gamma = {}^{n-1}g_\gamma - \left\langle {}^{n-1}g_\gamma, u_{n-1} \right\rangle u_{n-1},\ {}^{n-1}g_\gamma \in {}^{n-1}D \right\} \tag{11}$$

That guarantees that the elements of ${}^nD$ are orthogonal to the space generated by $\left(g_{\gamma_1^*}, g_{\gamma_2^*}, ..., g_{\gamma_{n-1}^*}\right)$.

1) Selection of g_{γ_n}

The element g_{γ_n} of ${}^nD$ that has the smallest angle with r_n in observation space is selected:

$$\cos^2\left(r_n, {}^ng_{\gamma_n}\right) = \max_{{}^ng_\gamma \in {}^nD} \left(\cos^2\left(r_n, {}^ng_\gamma\right)\right) \tag{12}$$

Thus, the model built from *n* functions can be written as:

$$\tilde{f} = \sum_{i=1}^{n-1} \alpha_i^* g_{\gamma_i}^* + \alpha_n g_{\gamma_n} \qquad \text{and} \quad \alpha_n = \left\langle f, g_{\gamma_n} \right\rangle \tag{13}$$

2) Tuning of g_{γ_n}

The tuning of g_{γ_n} is performed by minimizing the function $J\left(\gamma_n, \alpha_n\right)$:

$$J\left(\gamma_n, \alpha_n\right) = \sum_{i=1}^{N} \left(f_i - \tilde{f}\left(x_i\right)\right)^2 = \sum_{i=1}^{N} \left(f_i - \sum_{i=1}^{n-1} \alpha_i^* g_{\gamma_i}^*\left(x_i\right) - \alpha_n g_{\gamma_n}\left(x_i\right) \right)^2 \tag{14}$$

Let $\left(\gamma_n^*, \alpha_n^*\right)$ be the result of the optimization. As mentioned above, optimization is fast because the only variables of *J* are γ_n and α_n. The *n*-th function of the model is thus $g_{\gamma_n^*}$, and its coefficient is α_n^*.

Therefore the model is:

$$\tilde{f} = \sum_{i=1}^{n} \alpha_i^* g_{\gamma_i}^* \tag{15}$$

u_n is defined as:

$$u_n = g_{\gamma_n^*} - \sum_{i=1}^{n-1} \left\langle g_{\gamma_n^*}, u_i \right\rangle u_i \tag{16}$$

Thus one has

$$\text{span}(g_{\gamma_1^*}, g_{\gamma_2^*}, ..., g_{\gamma_n^*}) = \text{span}(u_1, u_2, ..., u_n) \tag{17}$$

and

$$\left\langle u_j, u_i \right\rangle = \delta_i^j \qquad \text{where } \delta_i^j \text{ is the Kronecker symbol} \tag{18}$$

which guarantees the orthogonality of the basis $(u_1, u_2, ..., u_n)$.

3) Orthogonalization of r_n

In order to compute the new residual r_{n+1}, which is the part of f in the null space of the space spanned by the $(g_{\gamma_1^*}, g_{\gamma_2^*}, ..., g_{\gamma_{n-1}^*}, g_{\gamma_n^*})$, one can write r_{n+1} as:

$$r_{n+1} = r_n - \left\langle r_n, u_n \right\rangle u_n$$

4) Orthogonalization of ${}^nD$

The set ${}^{n+1}D$ is computed as in (30):

$$\begin{aligned} {}^{n+1}D &= \left\{ {}^{n+1}g_\gamma = {}^n g_\gamma - \left\langle {}^n g_\gamma, u_n \right\rangle u_n, {}^n g_\gamma \in {}^n D \right\} \\ &= \left\{ {}^{n+1}g_\gamma = g_\gamma - \sum_{i=1}^{n} \left\langle g_{\gamma_i}, u_i \right\rangle u_i, g_\gamma \in D \right\} \end{aligned} \tag{19}$$

3.3. Iteration *n* = *M*

After M iterations, the family $\left(g_{\gamma_1^*}, g_{\gamma_2^*}, ..., g_{\gamma_M^*}\right)$ of waveforms from Ω and the family $\left(\alpha_1^*, \alpha_2^*, ..., \alpha_M^*\right)$ are built. Therefore, the model of the function f can be written as:

$$\tilde{f} = \sum_{i=1}^{M} \alpha_i^* g_{\gamma_i}^* \tag{20}$$

As in the OFR algorithm, one can, in principle, perform a final minimization of the mean square error by adjusting the whole set of parameters $\left(\alpha_1^*, \alpha_2^*, ..., \alpha_M^*\right)$ and $\left(\gamma_1^*, \gamma_2^*, ..., \gamma_M^*\right)$; it turns out, however, that the overall improvement is usually slight, and may not be worth the computation time.

Hence, the model of f that will be retained is the model described by equation (20).

3.4. Summary: GOFR *vs.* (OFR + optimization)

Regression with functions that are nonlinear in their parameters can be performed by feature selection from a large library of functions, followed by nonlinear optimization of a cost function with respect to all parameters initialized in the selection step. Thus, if n_f functions with p parameters have been selected by OFR, the process involves a nonlinear optimization in a space of dimension $n_f p$.

In GOFR, each selected function is optimized prior to orthogonalization, so that modeling by n_f functions with p parameters involves n_f nonlinear optimizations in a space of dimension p.

Therefore, if the number of parameters is small and the number of functions is large, GOFR may be expected to be much less computer intensive than OFR followed by simultaneous optimization of all parameters. That will be exemplified in section 4.4.

4. Application to the detection of the characteristic waveforms in ECG recordings

The above procedure is particularly efficient for the extraction of characteristic waveforms, as shown in the present section on the modeling of ECG signals. The ECG recording of a normal heartbeat is made of 5 characteristic peaks (Figure 3), termed “waves”, traditionally denoted as P, Q, R, S and T waves (see for instance [10]). The shape and position of the waves are the basis of the experts’ diagnosis. In order to design an effective diagnosis aid system, based on an automatic labeling of the waves, it is essential to accurately (i) locate those waves, and (ii) extract their shape.

To that effect, we used the GOFR algorithm described above, with a particular type of function specially designed to fit the cardiac waves that we will refer to as “Gaussian mesa”.

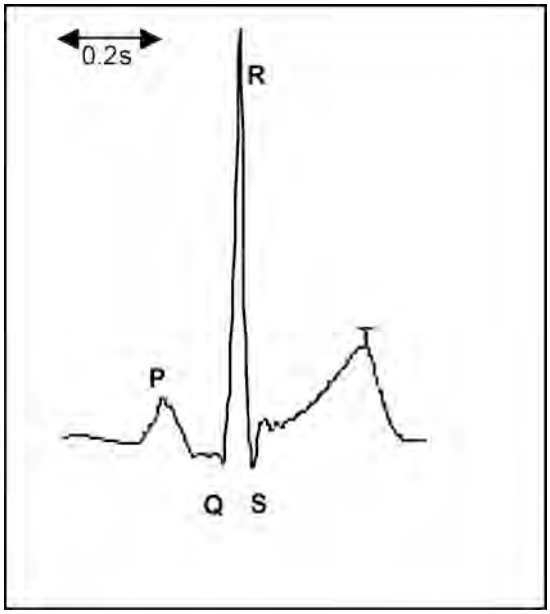
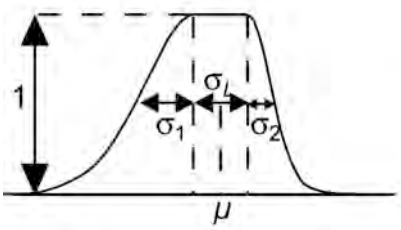


Figure 3

Typical heart beat with P, Q, R, S and T waves

4.1. Gaussian mesa function

The cardiac waves P, Q, R, S and T can be seen as positive or negative peaks below and above a baseline. The T wave is generally asymmetric, and, in some pathological cases, some waves exhibit a plateau. The Gaussian mesa waveform defined here makes it possible to fit exactly that kind of signal. A Gaussian mesa is an asymmetric function with 4 parameters and unit amplitude; it is made of two half-Gaussian functions connected with a linear, horizontal part (Figure 4). This function is continuous, differentiable, and all its derivatives with respect to its parameters are continuous, which is essential when applying standard optimization algorithms.



$$B(x,\mu,\sigma_1,\sigma_L,\sigma_2)=\begin{cases}\exp\left(-\frac{1}{2}\left(\frac{x-(\mu-\sigma_L/2)}{\sigma_1}\right)\right)^2 & \text{if } x\le\mu-\sigma_L/2\\ 1 & \text{if } \mu-\sigma_L/2\le x\le\mu+\sigma_L/2\\ \exp\left(-\frac{1}{2}\left(\frac{x-(\mu+\sigma_L/2)}{\sigma_2}\right)\right)^2 & \text{if } x\ge\mu+\sigma_L/2\end{cases}$$

Figure 4

Definition of the Gaussian mesa function

The 4 parameters are thus $\gamma = \{\mu, \sigma_1, \sigma_2, \sigma_L\}$:

μ: location in time,

σ_1: standard deviation of the 1st Gaussian function,

σ_2: standard deviation of the 2nd Gaussian function,

σ_L: length of the horizontal part,

The following conditions must be complied with:

$\sigma_1, \sigma_2 > 0, \sigma_L \geq 0$

In the following, we show how the GOFR algorithm was successfully applied to the modeling of heartbeat recordings by Gaussian mesa functions.

4.2. Library of Gaussian mesas

As mentioned in section 1, the library is constructed by sampling the set Γ of the parameters. That sampling requires a tradeoff: the sampling step must be small enough for fast convergence of training, but it must not be so small that it would increase the computation time during the subsequent orthogonalization step. Since the goal of the method is to provide a representation that matches the expert's representation of the signal, expert knowledge must be used at this point: in the present case, the narrowest peak to be modeled is at least 20 msec long [10], so that there is no point, in using library functions of width below 20 msec: hence one should have $\sigma_1 + \sigma_2 + \sigma_L > 20$ ms. Moreover, in order to decrease the number of functions (which is desirable, as shown in section 2.2), the library can be built from symmetrical mesas only, with horizontal part of length zero: $\sigma_1 = \sigma_2$ and $\sigma_L = 0$ (Figure 5). Since the GOFR algorithm performs a tuning of the parameters of the selected waveform just after its selection, the discretization of Γ may be coarse: in the present application, the library has only 132 symmetric Gaussian mesas.

4.3. Application of the GOFR procedure to Gaussian mesas for ECG modeling

The GOFR algorithm is run with M=6, since there are 5 characteristic peaks in a normal ECG heartbeat recording, and we allow for one extra function for modeling a possible spurious "bump" due to noise. Therefore, the following four steps were iterated 6 times:

- selection of the most relevant function g of D,
- tuning of g,
- orthogonalization of the ECG signal f,
- orthogonalization of library D.

The first selected function is shown on Figure 6.

During the tuning step, the parameters of the selected function are estimated; the result of that step is shown on Figure 7. Note that, in that case, constrained optimization is performed since σ_1, σ_2 and σ_L must be positive. In all numerical experiments reported here, optimization was performed by the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm [15] with appropriate modification to accommodate the constraints.

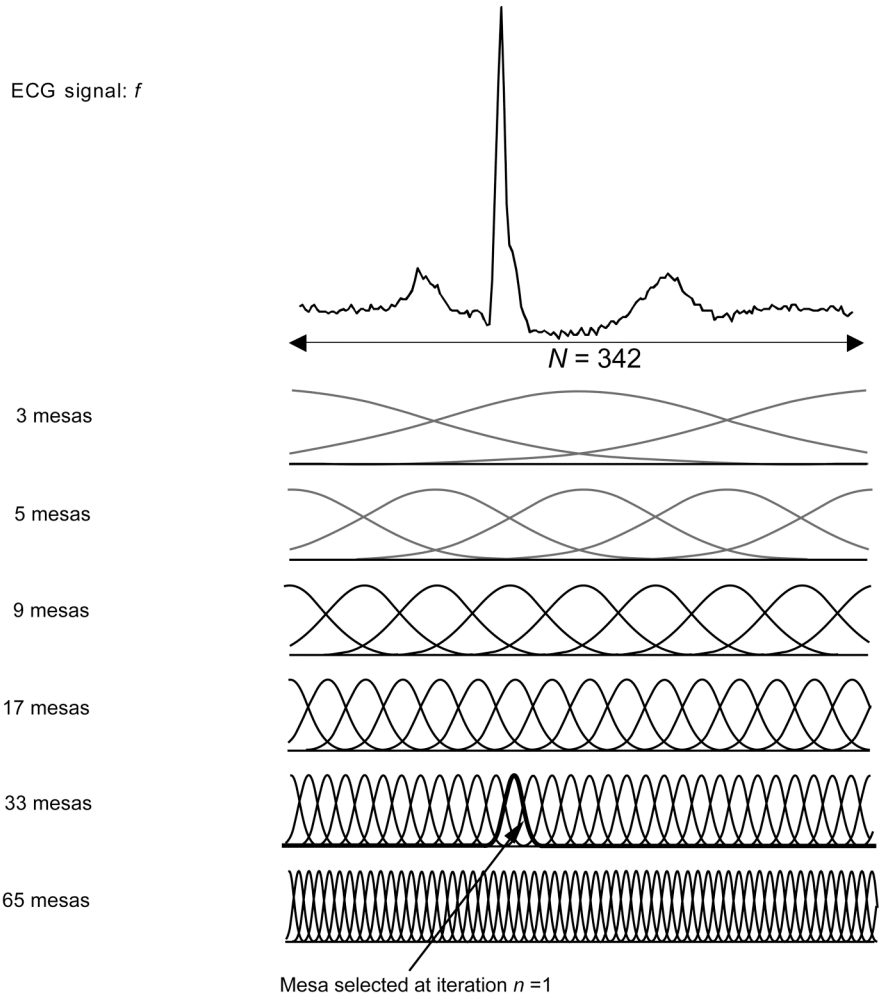


Figure 5

Library D is made of symmetric Gaussian mesas, with different locations and different standard deviations.

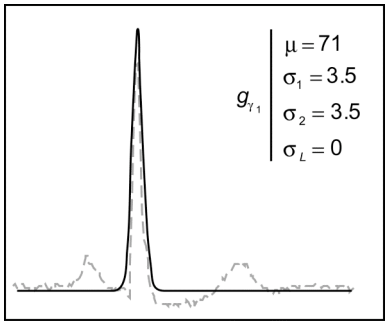


Figure 6

First Gaussian mesa function selected and signal *f*

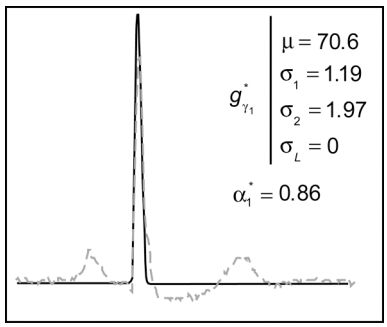


Figure 7

Tuned Gaussian mesa function

Then the part of the ECG that remains to be explained (Figure 8) is computed as shown previously (9), and the new library derived from the initial one is also computed (10).

After 6 iterations of that 4-step algorithm, the ECG has been broken up into 6 Gaussian Mesa functions (Figure 9).

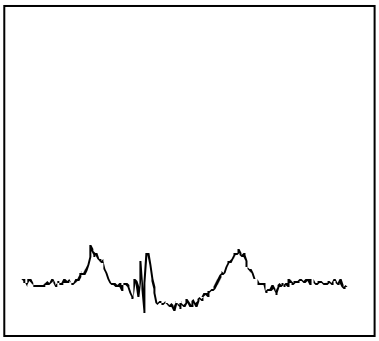


Figure 8

Part of the ECG that remains for modeling after iteration 1

Note that each characteristic cardiac wave is fitted by exactly one mesa function, which was the purpose of combining GOFR and mesa functions. The benefits of that property are illustrated in the next sections.

4.4. Comparison between GOFR and (OFR + optimization)

In order to provide a comparison between GOFR and OFR followed by optimization, on a non-academic example, we apply those algorithms to ECG heartbeats. Since the same parameters are optimized by the methods, the same library of functions (described in section 4.2) was used for both methods.

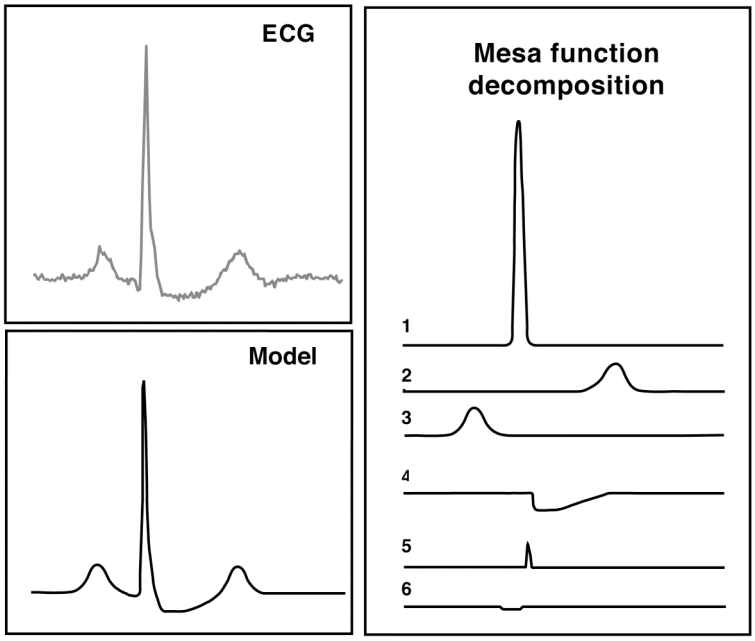


Figure 9

A normal heartbeat broken up into Gaussian mesa functions (shown in space *D*). Functions 1 to 5 will be assigned one of the "medical" labels P, Q, R, S, T as described in section 4.5, function 6 will be labeled as "noise".

As a first test, 100 heartbeats were modeled, from the MIT-BIH Arrhythmia database[I]. Table 1 shows the computation time per heartbeat and the mean square modeling error. In the present case, 6 functions are selected with 5 parameters each, so that the comparison is

[I] Available from http://ecg.mit.edu/

between 6 nonlinear optimizations in a 5-dimensional space and 1 nonlinear optimization in a 30-dimensional space. Note that the computation times include selection and orthogonalization, in addition to nonlinear optimization (see Figure 1). Clearly, the GOFR procedure is both more accurate and faster than OFR followed by optimization.

In addition, we discuss below the results obtained on 3 examples.

Example 1 (Figure 10) is a biphasic heartbeat[II]: the Q waves and the R waves have the same amplitude.

	Computation time[III]	Mean Square Error
OFR + optimization	29 msec	$1.41 \cdot 10^{-3}$
GOFR	18 msec	$0.17 \cdot 10^{-3}$

Table 1

Comparison of computation time and accuracy of OFR followed by simultaneous optimization of all parameters, and GOFR

Example 2 (Figure 11) is a ventricular ectopic heartbeat: this type of anomalous beat is very frequent; one of its specific features is that the width of the R wave is larger than 0.8 ms.

Example 3 (Figure 12) is an atrial ectopic beat: the upside-down P wave is typical of that anomaly.

It is clear from those examples that, if each wave is modeled by a single function (as shown in the present section), and if each function is subsequently assigned automatically a label P, Q, R, S or T (as described in section 4.5), automatic discrimination between such heartbeats can easily be performed from the parameters of the mesa functions that model each wave.

[II] Examples 1 and 2 are sampled from records #1001 and #1005 from the AHA database (American Heart Association database) [1]. Example 3 is sampled from the Ela Medical database (not available publicly).

[III] C program running under Windows XP on a Pentium IV-m, 2.8 Ghz.

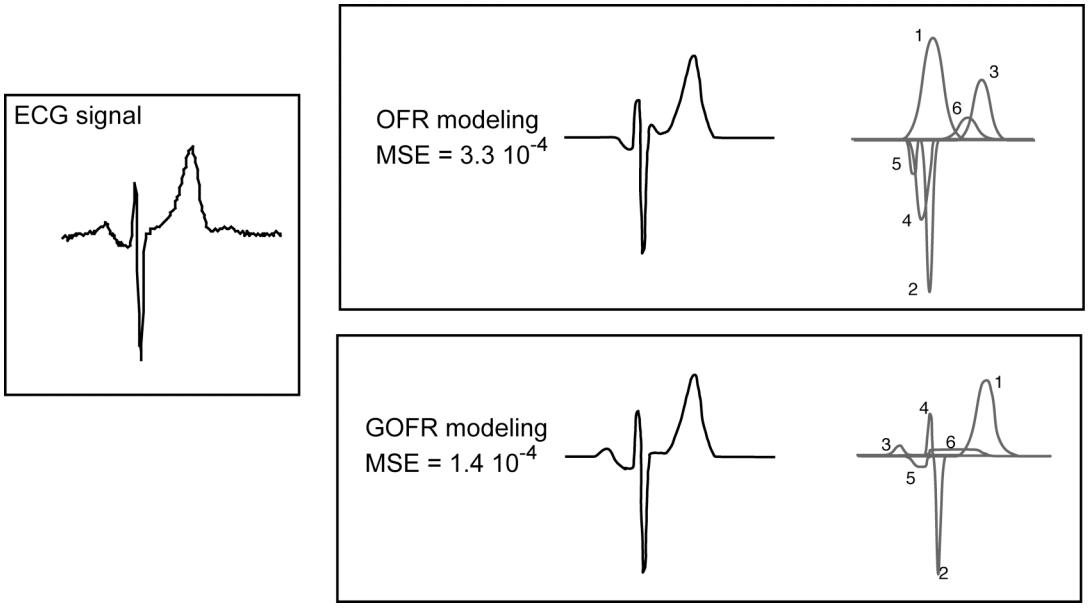


Figure 10

Comparison between OFR and GOFR models on a biphasic normal beat. *MSE* denotes the mean square modeling error.

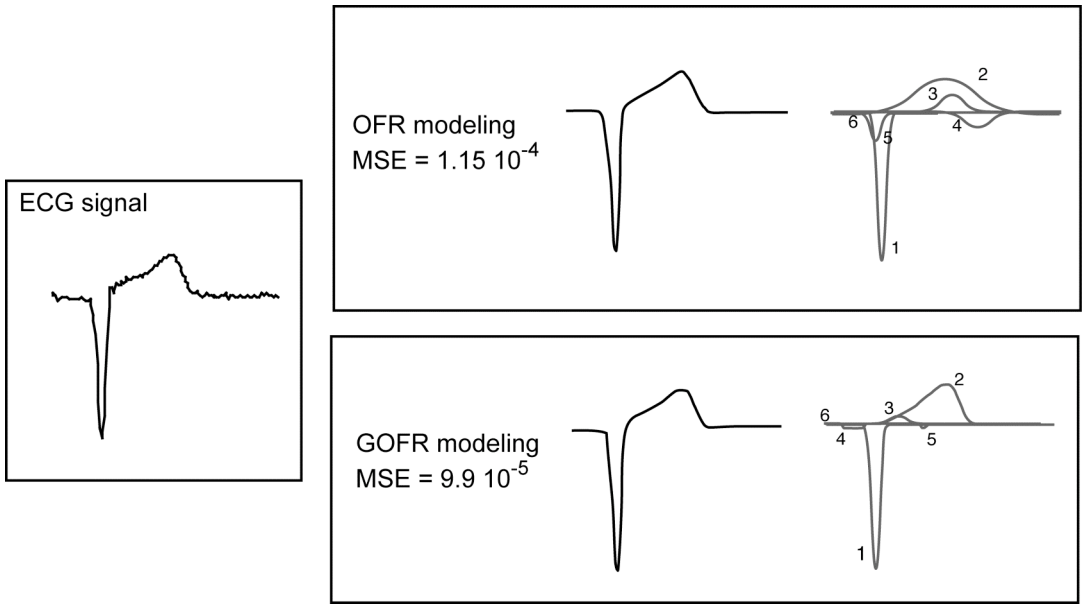


Figure 11

Comparison between OFR and GOFR on a ventricular ectopic beat. *MSE* denotes the mean square modeling error.

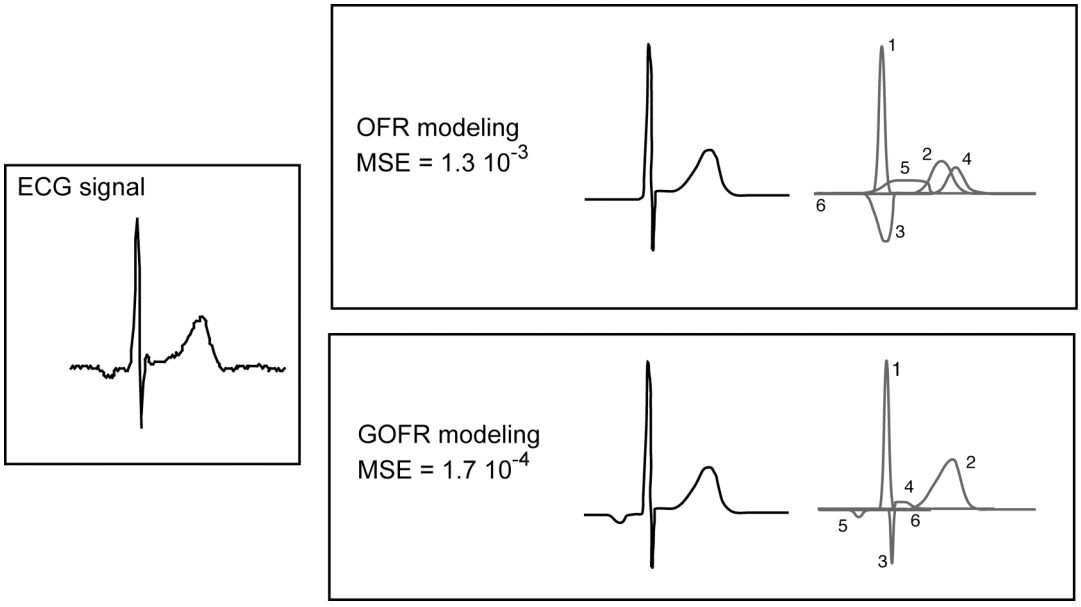


Figure 12

Comparison between OFR and GOFR on an atrial ectopic beat. *MSE* denotes the mean square modeling error.

In all those examples, the MSE (mean square error) is smaller for GOFR than for OFR. In addition, and more importantly, the representation of the characteristic cardiac waves is much more meaningful when obtained by the GOFR decomposition: each mesa function selected and tuned with the GOFR algorithm has a medical meaning, and, conversely each wave is modeled by a single mesa function. For example, in the atrial ectopic beat shown on Figure 12, the main information of the heartbeat (the upside down P wave) is not modeled with the OFR algorithm, while Gaussian mesa function number 4 models this anomaly by application of the GOFR algorithm.

The above examples are samples from a very large database. For a complete description of the application of the GOFR to the automatic analysis of Holter recordings (ECG recordings of 24-h duration), and its application to standard international ECG databases, the interested reader is referred to [8].

4.5. Application of the mesa function representation to heartbeat discrimination

The benefit of the modeling methodology described above is clearly illustrated in the final step of the process, which consists in assigning a “medical label” to each mesa function. Since

each wave is modeled by a single mesa function, a vector in 4-dimensional space describes each wave present in the database; therefore, classical discrimination methods can be used for assigning a label to each mesa function.

The task is performed in two steps (Figure 13): first, the R waves are labeled, in order to discriminate two different kinds of heartbeats, namely, the "normal"[IV] beats and the ventricular beats; the P, Q, S, T waves of non-ventricular beats are subsequently labeled.

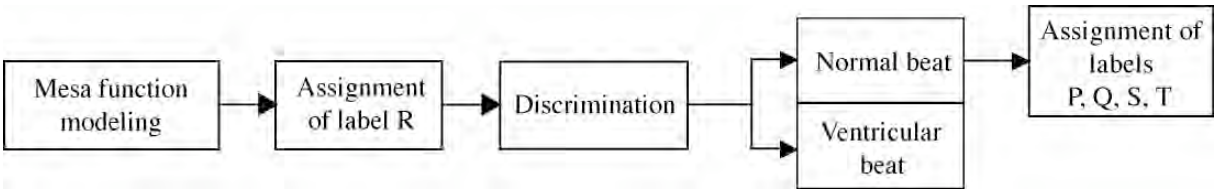


Figure 13

Assignment of medical labels (P, Q, R, S & T) to the mesa functions that model the heartbeats

4.5.1. Labeling the R waves

The labeling of the R waves is performed by discriminating R waves from non-R waves. A database of 960 mesa functions that model R waves and 960 mesa functions that model non-R waves was used for training and validation of a neural classifier. Testing was performed on a database with 960 mesa functions that model R waves and 7117 mesa functions that model non-R waves. The components of the input vector were the 5 parameters of the mesa functions, and the output was an estimate of the probability $\Pr(C_R \mid \mathbf{x}_i)$ of mesa function i being a R wave given the vector $\mathbf{x}_i$ of its parameters (Figure 14). For each heartbeat, the posterior probability was computed for each mesa function, and the mesa function with highest probability was assigned the label R. Finally, given the R wave (width, amplitude) and information about the context of the heart beat (rhythm, amplitude ratio with previous/next beat…), a knowledge-based decision tree was used for deciding whether this heart beat was a normal beat or a ventricular beat.

[IV] "Normal" beats should be more accurately termed "non-ventricular", since they can exhibit anomalies. However, we will follow the accepted terminology in cardiography.

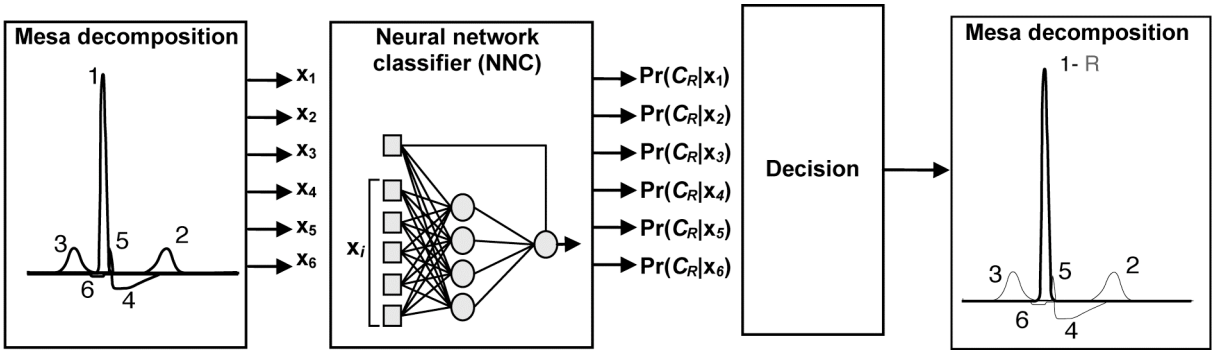


Figure 14

Procedure for assigning the R label

The labeling procedure was tested on two international databases, the AHA database and the MIT database; the results are shown on Table 2. They are better than results obtained by state-of-the-art published methods [2], and they provide a substantial improvement over results obtained by commercially available programs on the same databases [8].

	Normal Beats		**Ventricular Beats**	
	MIT Database	AHA Database	MIT Database	AHA Database
Number of normal beats	86,071	131,949	4,771	11,407
Sensitivity (%)	99.80	99.68	91.72	87.77
Positive predictivity (%)	99.47	98.95	95.46	95.93

Table 2

Result of R wave assignment and heart beat labelling on MIT and AHA database.

Sensitivity: $S = \frac{TP}{TP + FN}$ where TP is the number of true positives and FN he number of false negatives; Positive predictivity: $P = \frac{TP}{TP + FP}$ where FP is the number of false positives.

4.5.2. Labeling P, Q, S and T waves of non-ventricular heart beats

A similar procedure was applied to the labeling of the P, Q, S and T waves of non-ventricular beats. Four classifiers computed an estimate of the probability for each mesa function to belong to one of the classes (Figure 15). The label of the most probable class was assigned to the mesa function. Table 3 summarizes the data pertaining to each classifier. To the best of our knowledge, no algorithm performing the automatic labeling of the P, Q, S, T waves has ever been published.

The validation of this last part of the algorithm could not be performed on different databases because no database with P, Q, S, T labels is publicly available at present. Nevertheless, these results, obtained on the private database of Ela Medical, are very satisfactory.

	Hidden neurons	Training set size	Test set size	Misclassification rate on the training set (%)	Misclassification rate on the test set (%)
P wave classifier	3	1464	1710	0.3	0.5
Q wave classifier	3	600	290	2.8	2
S wave classifier	3	956	824	2	1.5
T wave classifier	5	2238	2506	0.5	0.8

Table 3

Architecture of each classifier for labeling P, Q, S and T waves.

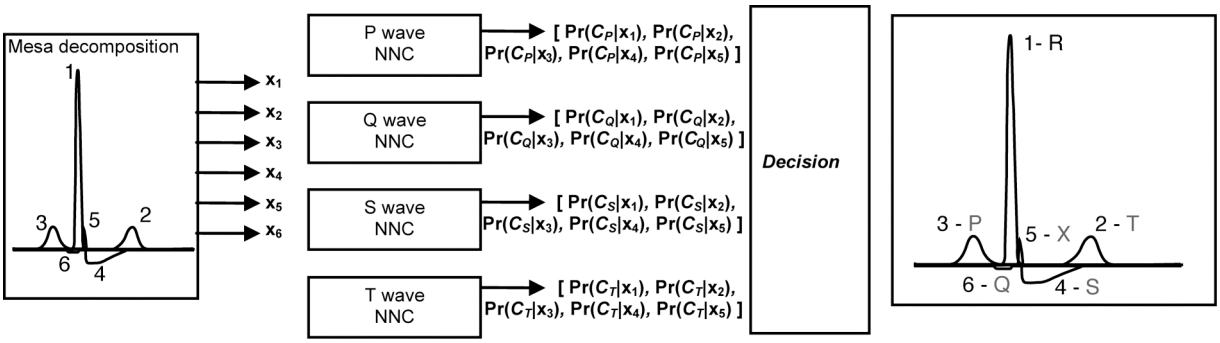


Figure 15

Procedure for labeling the P, Q, S and T waves. Since the heartbeat is modeled with 6 mesa functions, one of them is rejected by the classifier, hence assigned the label X

4.6. Application to 2-D data

The analysis of electrophysiological signals or electroencephalographic signals is more and more frequently performed in the time-frequency domain. Signals are wavelet-transformed, and the resulting map is analyzed in terms of time-frequency patterns of activity, arising in the form of localized “bumps” in the 2D-space of the map, which experts relate to the cognitive task being performed, or to the mental state of the patient. Thus, those “bumps” are the 2D-equivalents of the “waves” described in the present paper. The time-frequency maps arising from electrophysiological recordings in the olfactory bulb of rats while they were trained to

recognize odors were modeled as described in the present paper ([17], [18]); the modeling provided a very sparse representation of the areas of interest on the map, from which automatic discrimination between rats that had been trained to recognize an odor and "naïve" rats was performed.

In a completely different context, EEG recordings of patients who developed Alzheimer's disease one year and a half after the recording, and EEG of control subjects, were modeled by our technique [19]; the resulting representation allowed automatic discrimination between the two groups of recordings, thereby opening new opportunities for early detection of the disease. The detailed description of these applications is far beyond the scope of the present paper.

5. Conclusion

Signals are frequently modeled as parameterized linear combinations of functions such as wavelets, radial basis functions, sigmoids, etc. Orthogonal Forward Regression performs that task efficiently when the parameters of the functions are not adjustable, so that the model is linear in its parameters. In the present paper, we addressed the problem of designing models that are nonlinear with respect to their parameters, i.e. models where both the parameters of the functions that are combined linearly, and the parameters of that linear combination, are adjusted from data. Moreover, an additional constraint was taken into account, namely, the intelligibility of the model in terms of the (biomedical) significance of the functions that build up the model, for 1D- and 2D signals that exhibit peaks and troughs that have a definite meaning. We described a generalization of Orthogonal Forward Regression, for efficient nonlinear feature selection, and we defined a new family of very flexible parameterized functions, called Gaussian mesa functions. We illustrated the method by modeling long-duration electrocardiographic signals, where each wave of a heartbeat recording was successfully modeled by a single function, allowing the subsequent assignment of a medically meaningful label to each function. The method has been applied to the modeling of time-frequency maps of electrophysiology and electro-encephalography data; in the latter case, early detection of Alzheimer's disease was performed successfully.

.

APPENDIX 1

Orthogonal Forward Regression (OFR)

Since the paper describes a generalization of Orthogonal Forward Regression, readers may find a description of the latter useful.

As mentioned above, OFR is a three-step method (Figure 1):

- generation of a library D of feature functions from Ω,
- selection of M functions $\{g_i\}_{i=1,..M}$ chosen from D for the modeling of f,
- estimation of the parameters $\{\gamma_i, \alpha_i\}_{i=1..M}$ by minimization of the least squares modeling error J computed on the training set.

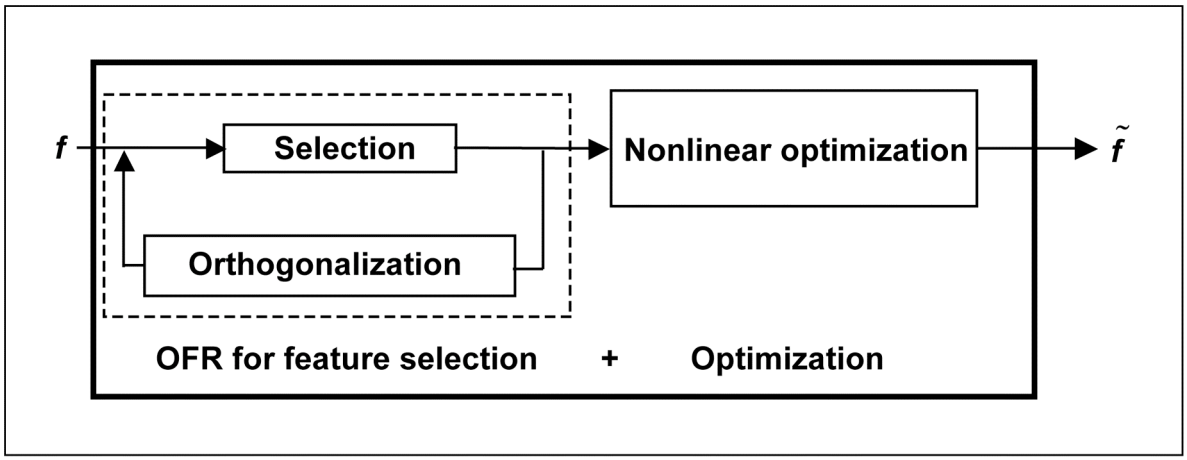


Figure 16

Graphical representation of the OFR algorithm followed by nonlinear optimization

1. Library construction

The construction of the library D of candidate features is performed by discretizing the space Ω, which amounts to discretizing the set of the parameters Γ. To that effect, it is necessary to choose a discretization step that is as small as possible in order to accurately represent Γ, albeit limited by the computational complexity that results from the number of candidate functions N_b of the library.

2. Gram-Schmidt orthogonalization for feature selection

During the selection step, the parameters $\{\gamma_i\}_{i=1..M}$ of the candidate functions are fixed. The model is thus linear in its adjustable parameters, which are the $\{\alpha_i\}_{i=1..M}$:

$$\tilde{f} = \sum_{i=1}^{M} \alpha_i g_{\gamma_i} \tag{21}$$

One can thus rank the N_b candidate features of the library D in order of decreasing relevance, given the data of the training set, and select only the M most relevant functions. That requires M iterations of the following Gram-Schmidt orthogonalization algorithm:

1. select the most relevant waveform g_γ from D,
2. orthogonalize the function f with respect to g_γ,
3. orthogonalize the library D with respect to g_γ.

2.1. Iteration $n = 1$

1) Selection of g_{γ_1}

The function g_{γ_1} is selected from the library ${}^1D = D$ as follows: g_{γ_1} is the function that has the smallest angle with the function f in observation space, i.e. in the N-dimensional space where the components of vector f are the N observed values f_k of f present in the training set:

$$g_{\gamma_1} = \arg \max_{g_\gamma \in D} \left(\cos^2 \left(f, g_\gamma \right) \right) \quad \text{where} \quad \cos^2 \left(f, g_\gamma \right) = \frac{\left\langle f, g_\gamma \right\rangle^2}{\|f\|^2 \left\| g_\gamma \right\|^2} \tag{22}$$

with

$$\left\langle f, g_\gamma \right\rangle = \sum_{k=1}^{N} f_k g_\gamma \left(x_k \right) \quad \text{and} \quad \|f\|^2 = \langle f, f \rangle = \sum_{k=1}^{N} {f_k}^2 \tag{23}$$

The function g_{γ_1} is the first feature of the model; in the following, it is denoted as $u_1 = g_{\gamma_1}$.

The information present in f that is still to be modeled (the residual) lies in the null space of u_1. Therefore the next two steps consist in projecting the function f and the library ${}^1D$ onto the null space of u_1.

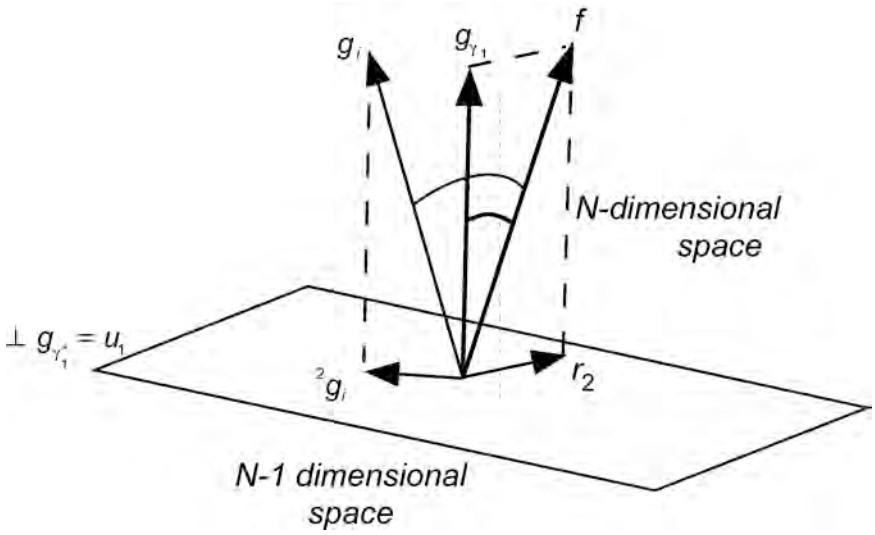


Figure 17

Orthogonalization with respect to u_1

2) Orthogonalization of f with respect to u_1 (Figure 17)

f is the sum of a vector of the space generated by u_1 and the residual vector r_2 in the null space of u_1:

$$f = \langle f, u_1 \rangle u_1 + r_2 \tag{24}$$

3) Orthogonalization of D

The new set ^2D is computed as the set of functions $^2g_\gamma$, where $^2g_\gamma$ is the part of the functions $^1g_\gamma \in {}^1D$ that lies in the null space of the selected function $g_{\gamma_1} = u_1$:

$$^2D = \left\{ \; ^2g_\gamma \; / \quad ^2g_\gamma = {}^1g_\gamma - \langle {}^1g_\gamma, u_1 \rangle u_1 \quad , \; {}^1g_\gamma \in {}^1D \; \right\} \tag{25}$$

At that point, r_2 must be expressed as functions of ^2D. To that effect, the same steps 1) 2) and 3) will be applied with the function r_2 and the library ^2D.

2.2. Iteration *n*

When iteration n starts, functions $\left(g_{\gamma_1},...,g_{\gamma_{n-1}}\right)$ of D are selected, and the orthogonal basis (u_1, u_2, …,u_{n-1}) is built so that the vectors $\left(g_{\gamma_1},...,g_{\gamma_{n-1}}\right)$ are in the subspace generated by the vectors (u_1, u_2, …, u_{n-1}):

$$\text{span}(u_1,...,u_{n-1}) = \text{span}(g_{\gamma_1},...,g_{\gamma_{n-1}}) \tag{26}$$

The functions that belong to ${}^nD$ lie in the null space of $\text{span}(u_1,...,u_{n-1})$.

$${}^nD = \left\{ {}^ng_\gamma \,/\, {}^ng_\gamma = {}^{n-1}g_\gamma - \left\langle {}^{n-1}g_\gamma, u_{n-1} \right\rangle u_{n-1}, {}^{n-1}g_\gamma \in {}^{n-1}D \right\} \tag{27}$$

The procedure at iteration n is as above:

1) Selection of g_{γ_n}

The element of ${}^nD$ that has the smallest angle with function r_n is selected:

$$g_{\gamma_n} \,/\, \cos^2\left(r_n, {}^ng_{\gamma_n}\right) = \max_{{}^ng_\gamma \in D_n} \left(\cos^2\left(r_n, {}^ng_\gamma\right)\right) \tag{28}$$

Denoting $u_n = {}^ng_{\gamma_n}$, the set of functions (u_1, u_2, …, u_n) is an orthogonal basis of the space generated by $\left(g_{\gamma_1},...,g_{\gamma_n}\right)$.

2) Orthogonalization of r_n

The part of the function to be modeled that remains to be explained is r_{n+1}, located in the null space of $\text{span}(u_1,...,u_n)$; r_{n+1} is computed as:

$$r_{n+1} = r_n - \left\langle r_n, u_n \right\rangle u_n \tag{29}$$

3) Orthogonalization of ${}^nD$

Similarly, the set ${}^{n+1}D$ is computed as the part of the elements of D located in the null space of $\text{span}(u_1,...,u_n)$:

$$\begin{aligned} {}^{n+1}D &= \left\{ {}^{n+1}g_\gamma \,/\, {}^{n+1}g_\gamma = {}^ng_\gamma - \left\langle {}^ng_\gamma, u_n \right\rangle u_n, {}^ng_\gamma \in {}^nD \right\} \\ &= \left\{ {}^{n+1}g_\gamma \,/\, {}^{n+1}g_\gamma = g_\gamma - \sum_{i=1}^{n} \left\langle g_{\gamma_i}, u_i \right\rangle u_i, g_\gamma \in D \right\} \end{aligned} \tag{30}$$

2.3. Termination $n = M$

The algorithm terminates when all N_b functions are ranked. However, it is not necessary to rank all candidate functions, since the only relevant functions are functions whose contributions to the model are larger than the noise present in the measurement of the signal to be modeled; based on that criterion, an efficient termination condition was proposed in [13], which stops the process after a number of iterations $M \le N_b$.

Whatever the termination criterion, at the end of the algorithm (iteration $n = M$), M functions of D $(g_{\gamma_1},...,g_{\gamma_M})$ are selected, and the orthogonal basis (u_1, u_2, …u_M) is generated.

One can write:

$$r_{M+1} = r_M - \langle r_M, u_M \rangle u_M \tag{31}$$

By summing over the M equations (29) the following relation is obtained:

$$f = \sum_{n=1}^{M} \langle r_n, u_n \rangle u_n + r_{M+1} \tag{32}$$

Since the set of vectors was constructed such that $\text{span}(u_1,...,u_M) = \text{span}(g_{\gamma_1},...,g_M)$, there is a single family $\{\alpha_i\}_{i=1,...M} \in \mathbb{R}$ such that:

$$f = \sum_{i=1}^{M} \alpha_i g_{\gamma_i} + r_{M+1} \tag{33}$$

One writes:

$$\tilde{f} = \sum_{i=1}^{M} \alpha_i g_{\gamma_i} \tag{34}$$

Thus, the model $\tilde{f}$ is built from the M most relevant waveforms $\left(g_{\gamma_1},...,g_{\gamma_M}\right)$ with the M parameters $\left(\alpha_1,\alpha_2,...,\alpha_M\right)$.

3. Optimization

The final step of the OFR procedure consists in estimating the parameters $\{\gamma_i^*, \alpha_i^*\}_{i=1,...M}$ that minimize the least squares cost function:

$$\{\gamma_i^*, \alpha_i^*\}_{i=1,...M} = \underset{\substack{\alpha \in \mathbb{R} \\ \gamma \in \Gamma}}{\arg\min} \left(J(\{\alpha_i, \gamma_i\}_{i=1,...M} \right) \tag{35}$$

with

$$J = \sum_{k=1}^{N} \left(f_k - \tilde{f}(x_k) \right)^2 = \sum_{k=1}^{N} \left(f_k - \sum_{\substack{i=1 \\ g_{\gamma_i} \in \Omega}}^{M} \alpha_i g_{\gamma_i}(x_k) \right)^2 \tag{36}$$

Therefore the model obtained with OFR algorithm is:

$$\tilde{f} = \sum_{i=1..M} \alpha_i^* g_{\gamma_i^*} \tag{37}$$

References

[1] AHA-DB, *AHA Database Series 1*: The American Heart Association Electrocardiographic - ECRI, 1997

[2] P. de Chazal, M. O'Dwyer, and R. Reilly, Automatic Detection of Heartbeats using ECG Morphology and Heartbeat Interval Features, IEEE Transactions on Biomedical Engineering 51 (2004) 1196-1206.

[3] S. Chen, S. A. Billings, and W. Luo, Orthogonal least squares methods and their application to non-linear system, Int. J. Control 50 (198) 1873-1896.

[4] S. Chen, C. F. N. Cowan, and P. M. Grant, Orthogonal Least Square Learning Algorithm for Radial Basis Function Networks, IEEE Transactions on Neural Networks 2 (1991) 302-309.

[5] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines. Cambridge: University Press, 2000.

[6] I. Daubechies, Ten Lectures on Wavelets, CBMS-NSF Series in Appl. Math. Vol. 61 (SIAM, Philadelphia, 1991).

[7] G. Davis, S. Mallat, and M. Avellaneda, Adaptive Greedy Approximations, Journal of Constructive Approximation 13 (1997), 57-98.

[8] R. Dubois, "Application de nouvelles méthodes d'apprentissage à la détection précoce d'anomalies en électrocardiographie", Thèse de Doctorat, Université Pierre et Marie Curie, Paris, 2003 (available from: http://www.neurones.espci.fr/Francais.Docs/dossier_recherche/bibliographie/theses.htm)

[9] I. Guyon and A. Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 3 (2003) 1157-1182.

[10] J. W. Hurst, Ventricular Electrocadiography (Lippincott Williams & Wilkins Publishers, 1990).

[11] S. Mallat and Z. Zhang, Matching Pursuits with Time-Frequency Dictionaries, IEEE Transactions on Signal Processing 41 (1993) 3397-3415.

[12] M. Minoux, Programmation Mathématique vol. 1 (Dunod, Paris, 1983).

[13] Y. Oussar and G. Dreyfus, Initialization by selection for wavelet network training, Neurocomputing 34 (2000) 131-143.

[14] T. Poggio and F. Girosi, "Networks for Approximation and Learning", Proceedings of the IEEE, 78 (1990) 1481-1497.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in C (Cambridge University Press, 1992).

[16] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar, Ranking a Random Feature for Variable and Feature Selection, Journal of Machine Learning Research, 3 (2003), 1399-1414.

[17] F. Vialatte, C. Martin, N. Ravel, B. Quenet, G. Dreyfus, and R. Gervais, Oscillatory activity, behaviour and memory, new approaches for LFP signal analysis, 35 th Annual General Meeting of the European Brain and Behaviour Society, Barcelona, Spain, 2003.

[18] F. Vialatte, Modélisation en bosses pour l'analyse des motifs oscillatoires reproductibles dans l'activité de populations neuronales: applications à l'apprentissage olfactif chez l'animal et à la détection précoce de la maladie d'Alzheimer. Thèse de doctorat de l'Université Pierre et Marie Curie, Paris. Available from http://www.neurones.espci.fr/Francais.Docs/dossier_recherche/bibliographie/theses.htm

[19] F. Vialatte, A. Cichocki, G. Dreyfus, T. Musha, R. Gervais, Early Detection of Alzheimer's Disease by Blind Source Separation, Time-frequency Transformation, and Bump Modeling of EEG Signals, Lecture Notes in Computer Science, 3696 (2005) 683-692, Springer.

[20] P. Vincent and Y. Bengio, Kernel Matching Pursuit, Machine Learning 48 (2002) 165-187.