Title
Regression analysis for peak designation in pulsatile pressure signals

Permalink
https://escholarship.org/uc/item/3vv1t2bj

Journal
Medical & Biological Engineering & Computing, 47(9)

ISSN
1741-0444

Authors
Scalzo, Fabien
Xu, Peng
Asgari, Shadnaz
et al.

Publication Date
2009-09-01

DOI
10.1007/s11517-009-0505-5

Peer reviewed

ORIGINAL ARTICLE

Regression analysis for peak designation in pulsatile pressure signals

Fabien Scalzo · Peng Xu · Shadnaz Asgari ·
Marvin Bergsneider · Xiao Hu

Abstract Following recent studies, the automatic analysis of intracranial pressure (ICP) pulses appears to be a promising tool for forecasting critical intracranial and cerebrovascular pathophysiological variations during the management of many disorders. A pulse analysis framework has been recently developed to automatically extract morphological features of ICP pulses. The algorithm is able to enhance the quality of ICP signals, to segment ICP pulses, and to designate the locations of the three ICP subpeaks in a pulse. This paper extends this algorithm by utilizing machine learning techniques to replace Gaussian priors used in the peak designation process with more versatile regression models. The experimental evaluations are conducted on a database of ICP signals built from 700 h of recordings from 64 neurosurgical patients. A comparative analysis of different state-of-the-art regression analysis methods is conducted and the best approach is then compared to the original pulse analysis algorithm. The results demonstrate a significant improvement in terms of accuracy in favor of our regression-based recognition framework. It reaches an average peak designation accuracy of 99% using a kernel spectral regression against 93% for the original algorithm.

F. Scalzo (✉) · P. Xu · S. Asgari · M. Bergsneider · X. Hu
Department of Neurosurgery, Geffen School of Medicine,
University of California, Los Angeles, USA
e-mail: fabien.scalzo@gmail.com

1 Introduction

The management of many neurological disorders relies on the continuous measurement of intracranial pressure (ICP). Dynamics of ICP reflect the brain's compensatory capability to intracranial volumetric changes and pathophysiological changes of the cerebral vasculature. Previous works have shown that variations of the ICP pulse morphology are linked to the development of intracranial hypertension [10, 20] and cerebral vasospasm [5], acute changes in the cerebral blood carbon dioxide (CO_2) levels [6, 19], and changes in the craniospinal compliance [9]. More generally, several studies [7, 8] have established a link between the morphology of the ICP pulse and the outcome head injured patients. Therefore, the automatic and continuous analysis of ICP morphological features appears to be promising for a better monitoring of pathophysiological intracranial and cerebrovascular changes.

An ICP pulse is typically triphasic [6] (i.e. three sub-peaks in each ICP pulse). Therefore, locating these three peaks in an ICP pulse would be an essential step to conduct a thorough analysis of ICP pulse morphological features because the calculation of the amplitude and timing of each peak can be readily carried out after knowing the designation of these peaks. A promising technique [1] was developed to detect the first peak of ICP pulses. Handcrafted features were extracted and used by different decision functions to determine if an incoming peak is a true component based on a threshold. However, processing an ICP signal to extract the three peaks in a continuous and robust

way is very challenging and beyond most of state-of-the-art ICP analysis methods [2, 9].

MOCAIP algorithm [16] (**Mo**rphological **C**lustering *and* **A**nalysis of **I**CP **P**ulse) has recently been developed for this purpose. In contrast with Morphologram [13] that analyzes the ICP offline, the framework is capable of extracting morphological changes of ICP pulse in real time. The MOCAIP algorithm offers several interesting properties; it is able to enhance ICP signal quality, to recognize legitimate ICP pulses and to detect the three sub-peaks in an ICP pulse. During the peak designation process, MOCAIP relies on a Gaussian model to represent the prior knowledge about the position of each peak in the pulse. The assignment is chosen such that it maximizes the probability of observing the peaks given the prior distributions. These priors have been previously learned from a training set of annotated data. This can be problematic because the position of the peaks within the pulse presents a large variation which is translated into large variance priors. This weakens the effectiveness of the peak designation step. Moreover, the ICP pulse itself, which contains potentially informative values, is not exploited directly during this step.

The current paper addresses these problems in the original MOCAIP algorithm [16] by utilizing machine learning techniques. The key idea is to exploit a regression model instead of using unimodal priors during the peak designation to improve the accuracy of the peak designation process. The regression model $y = f(x)$ is able to predict the most likely position of the three peaks, $y = (p_1, p_2, p_3)$, given a segmented ICP pulse discretized as a vector x. Different regression methods to predict the location of the peaks are compared in our experiments. The methods considered in this work include Multi-Linear Regression [12], Support vector machine (SVM) algorithm [7], recently developed spectral regression (SR) analysis [4], and extremely randomized decision trees (Extra-Trees) [14].

2 Methods

2.1 MOCAIP algorithm

MOCAIP is a recently developed algorithm to identify the three peaks that occur in an ICP signal. The recognition of ICP peaks is achieved through three major tasks. The first task consists of robustly segmenting a continuous ICP signal into a sequence of individual ICP pulses (i.e. heartbeats). This is done by using a pulse detection algorithm, a clustering algorithm, and a filtering process that identifies valid pulses. The second task is to detect all the candidate peaks in each ICP pulse. Finally, the third task relates to the designation of the three peaks among the detected candidates.

2.1.1 Detection, clustering, and validation

MOCAIP starts by segmenting the continuous ICP into a sequence of individual ICP pulses. To this end, MOCAIP combines an ICP pulse extraction technique [15] with the ECG QRS detection [2] that finds each ECG beat.

ICP recordings collected from the bedside monitors can be contaminated by several types of noise and artifacts. Instead of applying ICP morphology analysis using individual pulses separately, a representative cleaner pulse is extracted from a sequence of consecutive ICP pulses. A hierarchical clustering approach [17] is used, and the the centroid of the main cluster is extracted. We refer to it as the dominant pulse, and we denote it as S_i.

When the signal is heavily contaminated by artifacts, a dominant pulse extracted from a signal sequence might not correspond to a valid pulse. To identify valid ICP pulses automatically, MOCAIP exploits a reference library containing validated ICP pulses. A pulse is judged to be valid if it belongs to a cluster whose average pulse correlates with any of the reference ICP pulses.

2.1.2 Detection of candidate peaks

Once a valid ICP pulse S_i has been extracted, MOCAIP detects a set of peak candidates (or curve inflections). Each of them is potentially one of the three peaks. The extraction of these candidates relies on the segmentation of the ICP pulse into concave and convex regions. This is done using the second derivative of the pulse. Typically, a peak corresponds to the intersection of a convex to a concave region on a rising edge of ICP pulse or to the intersection of a concave to a convex region on the descending edge of the pulse. This detection process produces a pool of N peak candidates $(a_1, a_2, ..., a_N)$.

2.1.3 Assignment of detected peaks

The last task of the MOCAIP algorithm is to identify the three ICP peaks (p_1, p_2, p_3) from the set of candidate peaks. Given $P_i(a_j)$, $i = 1, 2, 3$ to denote the probability density functions (PDF) of assigning a_j to the i-th peak (each PDF is a Gaussian distribution estimated from peak locations previously detected on a set of reference ICP pulses). The peak assignment amounts to searching for the maximum of the following objective function

$$J(i, j, k) = P_1(a_i) + P_2(a_j) + P_3(a_k) \qquad (1)$$

In order to deal with missing peaks, an empty designation a_0 is added to the pool of candidates. In addition, to avoid false designation, MOCAIP uses a threshold ρ such that $P_i(a_k) = 0$, $i \in \{1, 2, 3\}$, $k \in \{1, 2,..., N\}$ if the probability of assigning a_k to p_i is less than ρ.

2.2 Regression analysis for peak designation

During the peak assignment, the MOCAIP algorithm exploits Gaussian priors to infer the position of the three peaks from a set of peak candidates. Because there exist large variations in the pulse morphology of the ICP signals, the actual position of each of the three peaks is extremely variable. The strategy employed by MOCAIP limits its ability to cope with the complexity of data and therefore may lead to wrong or missed assignments.

This section focuses on this problem and introduces an extension of the MOCAIP algorithm. The innovative idea is to consider the position (p_1, p_2, p_3) of the peaks as a function f of the pulse signal. To this end, a regression model (previously learned) is exploited instead of the Gaussian priors during the peak designation to improve the accuracy of the process. The strength of using this model is that it exploits the values of the pulse itself during the peak assignment. Another advantage is the ability of the framework to exploit powerful machine learning algorithms (Sect. 2.2.3).
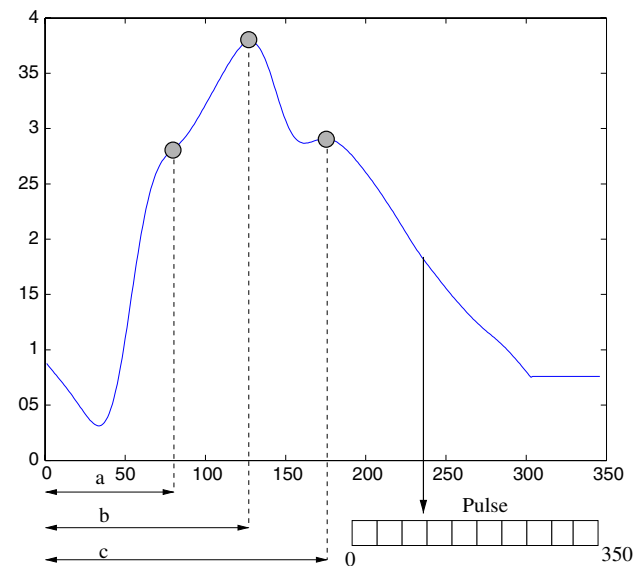
Fig. 1 A regression model $f(x)$ is used to predict the positions a, b and c, of the three peaks. The pulse is discretized and normalized into a vector x

During the learning phase, a regression model $y_i = f(x_i)$ is estimated from a set $S = \{ S_{i=1 \dots n} \}$ of training pulses (i.e. inputs) labelled with the locations of the peaks $y_i = (p_1, p_2, p_3)$ (i.e. outputs) within the pulse. For simplicity, each pulse S_i is resized to a vector $x_i \in \mathbb{R}^s$ of length s following the procedure described in Sect. 2.2.1 and illustrated in Fig. 1.

For recognition on a previously unseen pulse x_j, the regression model $y_j = f(x_j)$ predicts the most likely position of the three peaks $y_j = (p_1, p_2, p_3)$. In parallel, a set of candidate peaks $a_1, a_2, ..., a_N$ is extracted at curve inflections. Then a matching algorithm Sect. 2.2.2 is used to assign the closest peak candidates to the predictions of the regression model.

2.2.1 ICP pulse pre-processing

In order to be processed by the regression analysis, each ICP pulse S_i (sampled at 400 Hz) has to be represented as a vector $x_i \in \mathbb{R}^s$. Because the length of the pulse may vary, it is resized to a vector of fixed length $s \in \mathbb{N}^+$, such that it is set proportional to the average pulse length on the training data,

$$s = \alpha \frac{\sum_i^n \text{length}(S_i)}{n} \qquad (2)$$

where α was empirically set to 1.7 during our experiments. The feature vector x_i corresponds to the normalized pulse S_i if it has a length of s. If the length of the pulse S_i is larger than s, the extra-values at the end of the pulse are discarded. If the pulse is smaller than s, the last value $S_i(\text{last})$ is repeated to fill the vector, as written formally

$$x_i(j = 1 \dots s) = \begin{cases} S_i(j) & \text{if } (j \leq \text{length}(S_i)) \\ S_i(\text{last}) & \text{otherwise} \end{cases} \qquad (3)$$

In addition, each pulse vector x_i is then normalized such that the minimum and maximum values of the vector are respectively 0 and 1.

2.2.2 Prediction assignment algorithm

As a final step, the locations (p_1, p_2, p_3) predicted by the regression model are used to identify the peaks in the set of candidates $(a_1, a_2, ..., a_N)$. As summarized in the Algorithm 2, the closest candidate peaks to any prediction is assigned to the peak label corresponding to the matched prediction. After the assignment, the candidate peak is removed from the set. Therefore, each candidate peak is assigned to maximum one label. This is repeated three times. A threshold λ is used to avoid the assignment of candidate peaks whose distance to its closet prediction is too large.

Algorithm 1 Prediction Assignment

```
1:  // detect peak candidates
2:  a ← candidates(xᵢ)
3:  // detect peak predictions
4:  {t₁, t₂, t₃} ← predict(xᵢ)
5:  for each j < 3 do
6:      // compute the distance between the
            predictions and each candidate
7:      {d₁, i₁} ← min |a − t₁|
8:      {d₂, i₂} ← min |a − t₂|
9:      {d₃, i₃} ← min |a − t₃|
10:     {dⱼ, k} ← min(d₁, d₂, d₃)
11:     if dⱼ <= λ then
12:         // assign the peak candidate
13:         pⱼ ← a(iₖ)
14:         a(iₖ) ← ∅
15:     else
16:         pⱼ ← ∅
17:     end if
18: end for
```

2.2.3 Regression analysis

Regression analysis is a statistical technique used for the numerical analysis between an input variable $x_i \in X$ and an output variable $y_i \in Y$. This section provides an overview on different regression analysis methods that will be used within our framework; these includes Multi-Linear Regression [8], support vector machine (SVM) algorithm [7], Spectral regression (SR) analysis [4], and extremely randomized decision trees (Extra-Trees) [14].

2.2.4 Multiple linear regression

A common way to obtain a regression model is to perform a multiple linear regression (MLR) analysis [8]. The intuition behind this technique is to fit a model such that the sum-of-squares error (SSE) between the observed and the predicted values is minimized.

Let X be a set of n input variables $x_i \in \mathbb{R}^s$ (i.e. normalized pulse values), Y of set of observations $y_i \in \mathbb{R}^3$ (i.e. peak positions) and β a $s \times 3$ matrix of parameters, the multiple linear regression model is expressed as follow:

$$Y = \beta X + \epsilon \tag{4}$$

$$\Leftrightarrow y_i = \beta_1 x_i(1) + \beta_2 x_i(2) + \cdots + \beta_s x_i(s) + \epsilon_i \tag{5}$$

where $i = 1...n$ and $\varepsilon_i = N(0, \sigma^2)$ denotes a set of noise variables.

The goal of the multiple linear regression (MLR) analysis is to find estimates $\hat{\beta}$ to the coefficients β such that they minimize the sum-of-squares error (SSE),

$$\hat{\beta} = \operatorname{argmin}_\beta \sum_{i=1}^{n} (\beta x_i - y_i)^2 \tag{6}$$

The optimal $\hat{\beta}$ can be expressed as

$$\hat{\beta} = (XX^T)^{-1} X^T Y \tag{7}$$

We used a QR factorization to obtain $\hat{\beta}$. The estimated regression coefficients $\hat{\beta}$ can then be used to predict the output values $\hat{y}'_i \in \hat{Y}'$ from a set of previously unseen data $x'_i \in X'$,

$$\hat{Y}' = \hat{\beta}^T X' \tag{8}$$

2.2.5 Spectral regression analysis

The spectral regression analysis (SR) [4] is a recent method which combines spectral graph analysis and standard linear regression. The goal consists of finding a regression model which has similar predictions $\hat{y}_i \in \hat{Y}$ for data samples $x_i \in X$ that are close (i.e. that are nearest neighbors in a graph representation), such that the following measure ϕ is minimized:

$$\phi = \sum_{i,j=1}^{n} (\hat{y}_i - \hat{y}_j)^2 W_{i,j} \tag{9}$$

where $W \in \mathbb{R}^{n \times n}$ is the affinity (i.e. item-item similarity) matrix that associates a positive value to $W_{i,j}$ if the samples x_i, x_j belong to the same class.

More precisely, this is done by first using the eigenvectors of the affinity matrix W,

$$We = \lambda De \tag{10}$$

where D is a diagonal matrix whose entries are column sums of W, $D_{i,i} = \sum_j W_{j,i}$, and e_0, e_1, ..., e_d denote the $d + 1$ eigenvectors with respect to the $d + 1$ largest eigenvalues $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_d$.

Then Spectral Regression finds d vectors $\{\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_d\}$ that minimize the residual sum of square error (SSE),

$$\hat{\beta}_j = \operatorname{argmin}_\beta \sum_{i=1}^{n} (\beta^T x_i - y_i^j)^2 \tag{11}$$

where y_i^j is the i-th element of e_j.

Spectral Regression is a linear regression algorithm. However, it can easily be extended to nonlinear problems by using a kernel projection (i.e. "kernel trick"). This technique allows to use a linear regression analysis to solve a nonlinear problem by mapping the observations into a higher-dimensional space, where the linear regression is subsequently used. In our framework, a radial basis function (RBF) kernel is used as a projection matrix,

$$K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}, \quad \gamma > 0 \tag{12}$$

We further refer to this technique as the Kernel spectral regression (KSR).

2.2.6 Support vector machine regression

A support vector machine (SVM) [7] is a supervised learning technique that has been used extensively in a wide range of pattern recognition applications.

When used in a regression framework, SVM aims at finding a function f that maps any input x_i to its output label y_i. This is done using the optimization presented in Eq. 13. Intuitively, it allows errors as long as they are less than ε, while trying to make f as "flat" as possible.

$$\text{argmin}_{w,b,\xi^{+,-}} \frac{1}{2}\langle w, w\rangle + C \sum_{i=1}^{N} \xi_i^- + \xi_i^+$$
$$\text{subject to}(\langle w, \phi(x_i)\rangle + b) - y_i \leq \epsilon + \xi_i^+ \quad (13)$$
$$y_i - (\langle w, \phi(x_i)\rangle + b) \leq \epsilon + \xi_i^-$$
$$\xi_i^-, \xi_i^+ > 0$$

where $\phi(x_i)$ corresponds to the projection of x_i into a higher dimensional feature space, $< .,. >$ denotes the inner product between two vectors. Vectors ξ_i^- and ξ_i^+ correspond to the lower and upper parameters in which the estimated function $g(x_i) = \ < w, \phi(x_i) > \ + b$ is allowed to vary for a given error ε and cost C.

In practice, this optimization is solved using the dual maximization problem (more details can be found in the reference paper [7]):

$$\text{argmin}_{\alpha^+, \alpha^-} \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) K(x_i, x_j)$$
$$+ \epsilon \sum_{i=1}^{n} (\alpha_i^+ + \alpha_i^-) + \sum_{i=1}^{n} y_i (\alpha_i^+ - \alpha_i^-) \quad (14)$$

$$\text{subject to} \sum_{i=0}^{n} (\alpha_i^+ - \alpha_i^-) = 0 \text{ and } 0 \leq \alpha_i^{+,-} \leq C$$

which has the following objective function,

$$g(x) = \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-) K(x, x_i) + b \quad (15)$$

where x is a new input vector, and α^+, α^- correspond to the upper and lower error boundaries.

In our framework, $K(x_i, x_j)$ is a RBF kernel function (Eq. 12) that maps input features into another space in which the samples are hopefully linearly separable.

2.2.7 Extremely randomized decision trees

Extremely randomized decision trees (Extra-Trees) [14] is a machine learning method that extends classical decision trees by introducing stochasticness during the induction process. Extra-Trees consists of an ensemble of randomized binary decision trees.

The induction algorithm of a tree takes the form a top-down process that successively splits the leaves where the output varies. Each internal node is annotated with a threshold λ_j that is defined on an input attribute $j \in [1, 2,..., m]$ randomly selected. The algorithm sets the value λ_j to a pseudo-random value depending on a Gaussian distribution $\mathcal{N}(\mu_j, \sigma_j)$ (estimated from the training samples), where μ_j, σ_j are the mean and standard deviation of this attribute j. The construction stops at a given node when its output values are constant for all the training samples.

To obtain output predictions $\hat{y}_i$ from a new, previously unseen input vector x_i, the vector is independently processed by each of the $k \in K$ trees ($K = 50$ in our implementation). The predicted values $\hat{y}_k^i = f_k(x_i)$ originating from the trees are collected, and the final prediction is computed as a weighted average,

$$\hat{y}_i = \sum_{k=1}^{K} w_k \hat{y}_k^i \quad (16)$$

such that $\hat{y}_k^i = f_k(x_i)$ and $\sum_k w_k = 1$. The weights w_k are set proportional to the accuracy obtained by the tree f_k on the training set.

3 Results

The dataset used in our experiments originates from the UCLA ADULT HYDROCEPHALUS CENTER and has previously been used to evaluate MOCAIP [16]. The usage of this archived dataset in the present work was approved by the UCLA Internal Review Board.

The ICP and ECG data were collected from 64 inpatients treated for various intracranial pressure related conditions. Data belonging to two patients, which were included in the original MOCAIP study had to be removed in the present study because a closer scrutiny revealed that no valid ICP recordings were obtained for these patients because of device malfunctions. ICP was monitored continuously using Codman intraparenchymal microsensors (Codman and Schurtleff, Raynaud, MA) placed in the right frontal lobe. ICP signals were recorded from bedside monitors using corporate data acquisition systems.

The sampling rate during the recordings was either 240 or 400 Hz. A total of 153 signal segments of approximately 5-h long were extracted every 12 h. These ICP and ECG signal segments were subsequently processed by MOCAIP to produce a set of 14230 raw dominant pulses. Among these, 13,611 were considered as valid pulses and constitutes the dataset $\Theta = \{X, Y\}$ that will be used in our experiments. Each pulse was labelled manually by experienced researchers with the position of the three peaks.

Whenever a peak was missing (i.e. no curve inflection), it was labelled with the empty set. The dataset is particularly challenging because there exists a large variability in the ICP signals that may arise from the individual condition of each patient. Moreover, among the set of pulses, 1717 have missing p_1, 265 have missing p_2 and 34 have missing p_3.

3.1 Comparative analysis of regression methods

The different regression techniques that have been described in Sect. 2.2.3 are now compared. Their strength is quantified on an ICP peak detection task by measuring their precision, generalization power on new patients, cost in complexity, and sensitivity to the number of training samples. The technique that performs the best across these different evaluations will be chosen to be included in the extension of the MOCAIP extension framework Sect. (2.2. The following acronyms LIN, SVM, SR, KSR, and EXTRA-Trees will be used in this section to refer to Multiple Linear, Support Vector Machine, Spectral Regression, Kernel Spectral Regression, Extremely Randomized Decision trees methods respectively.

3.1.1 Prediction accuracy

This experiment evaluates the accuracy of each regression method to detect the three peaks within ICP pulses. The error e is measured in terms of average prediction error, in milliseconds (ms), between the actual position of the peaks $y_i = (p_1, p_2, p_3)$ and the prediction $\hat{y}_i = (\hat{p}_1, \hat{p}_2, \hat{p}_3)$ of the regression method,

$$\{e_{p_1}, e_{p_2}, e_{p_3}\} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \qquad (17)$$

and $e = (e_{p_1} + e_{p_2} + e_{p_3})/3$, where e_{p_i} denotes the prediction error for the i-th peak.

A five-fold crossvalidation is performed on a dataset $\Theta' \subset \Theta$ consisting of 11080 ICP pulses. These pulses were randomly selected from the dataset Θ and they were required to contain three peaks. For each fold, 8,864 pulses are used for training and 2,216 are used as a test set. The partitioning was made randomly such that each pulse appeared at least once in the test set. The results obtained by each method are reported in Fig. 2 as light blue bars. KSR, $e^G_{KSR} = 1.32$ ms ± 0.053, and Extra-Trees $e^G_{XT} = 2.64$ ms ± 0.21 offer the lowest prediction error. They are closely followed by SVM $e^G_{SVM} = 3.77$ ms ± 0.06. Linear regression approaches $e^G_{SR} = 7.98$ ms ± 0.13 and $e^G_{LIN} = 8.01$ ms ± 0.14 does not perform as well. This suggests that the relation between the pulse x and the position y of the peaks is non-linear and cannot be captured precisely by linear approaches.
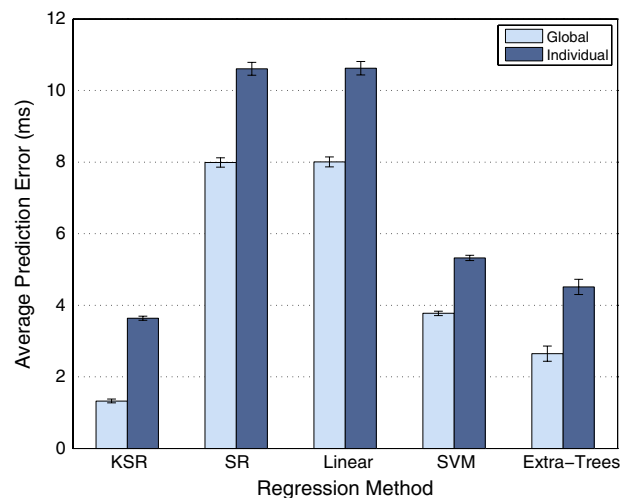
Fig. 2 Average error between the predicted and the actual position of each peak (Eq. 17). Results are reported for different regression models (KSR, SR, Linear, SVM, Extra-Trees). For each technique, the results are compared between globally (*light blue*) and individually (*dark blue*) trained models. For global models, a standard five-fold cross-validation is used to evaluate the models on the all dataset. In contrast, Individual models are evaluated on each patient by excluding the data of that particular patient from the training set

3.1.2 Prediction accuracy on new patients

The regression models used in the previous experiment were trained globally. Such an approach does not take into account that the data from the same patients probably appear both in the training and test set. In order to measure the generalization ability of the model, it is more realistic to perform the evaluation on previously unseen patients only. This is referred to as the *individual* approach.
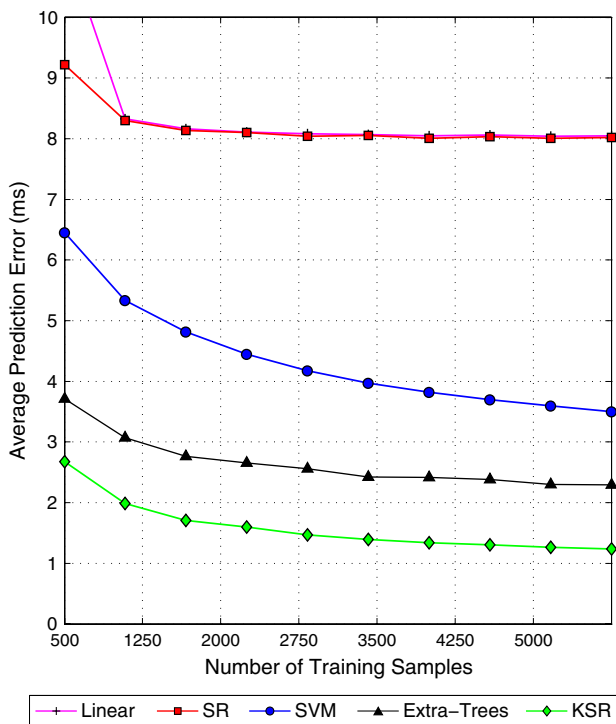
The protocol in this experiment is to train a regression model separately for each patient such that its data are not included in the training set, but used as a test set. This can be seen as a 64-fold crossvalidation (one fold for each of the 64 patients) such that the training set of each fold consists of 4,000 randomly selected pulses from all the patients excluding one. Each test set is made of ICP pulses of the patient data that were excluded of the training set. The prediction error (Eq. 17) averaged over the 64 test sets is shown in Fig. 2 as dark blue bars. The error of these individual models is compared, for each regression technique, to the results of the *global* model trained in the previous paragraph. The prediction on new patients is obviously more challenging. For KSR the error changes from $e^G_{KSR} = 1.32$ ms for the global model to $e^I_{KSR} = 3.63$ ms for the individual models, SVM from $e^G_{SVM} = 3.77$ ms to $e^I_{SVM} = 5.32$ ms, SR from $e^G_{SR} = 7.98$ ms to $e^I_{SR} = 10.60$ ms and linear from $e^G_{LIN} = 8.01$ ms to $e^I_{LIN} = 10.62$ ms. Similarly to the previous experiment, KSR and Extra-Trees perform the best. They offer the best

generalization on new patients and obtain the lowest average prediction error.
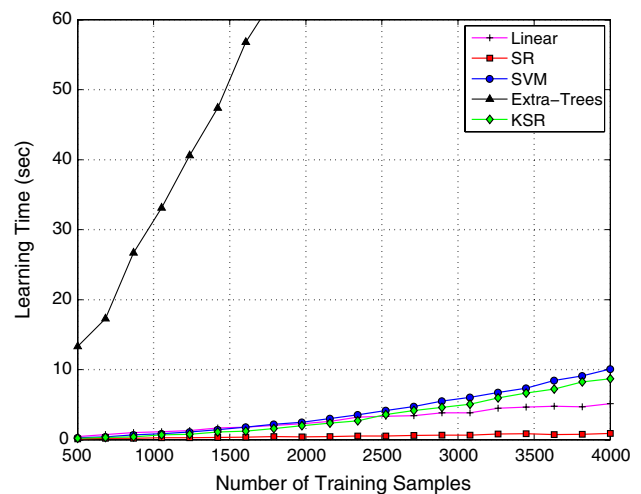
3.1.3 Number of training samples

The estimation of the regression models relies on a set of training ICP pulses. The number of these training samples has generally two conspicuous impacts. On the one hand, the accuracy is expected to be better as the number of labelled examples increases. On the other hand, it tends to make the learning process more costly in terms of memory uses and running time. The current experiment focuses on these two aspects.

The experimental protocol is similar to the one used for the global evaluation. The average prediction error of the peak position (Eq. 17) is measured using a five-fold cross-validation that is performed on the 11080 ICP pulses. The number of training samples varies from 500 to 6,000. The average prediction error of the peaks is shown against the number of training samples in Fig. 3. It can be seen that, for KSR and SR, the error quickly decreases from 2.67 to 1.99 ms between $n = 500$ and $n = 1,050$. Then it decreases much slower, to finally reach an average error of 1.24 ms.

Fig. 4 Effect of the number of training samples on the average training time and standard deviation for different regression models (KSR, SR, Linear, SVM, Extra-Trees). Extra-Trees seem to be the most costly technique in terms of computational complexity

The great performances of KSR, SVM and especially Extra-Trees, come with a greater cost in computational complexity. Using the same protocol, but this time the number of training samples varies from $n = 500$ to $n = 4,000$. Figure 4 shows to what extent this is true. It can be noticed that the training time is about the same for a small number of samples ($n = 500$), KSR = 0.19; SR = 0.12; LIN = 0.41; SVM = 0.25, however some methods become prohibitive in terms of time consumption for a larger number of samples ($n = 4,000$) [KSR = 8.69; SR = 0.89; LIN:5.13; SVM:10.07] sec.

3.2 Peak recognition

In the following, we evaluate the proposed extension of MOCAIP based on the Kernel spectral regression (KSR). The recognition results are compared to the original MOCAIP.

In these experiments, a predicted position $\hat{y}_i$ is considered as correct if it does not differ by more than 30 ms from the actual position y_i. Based on this definition, it is possible to say if a prediction is a true positive (TP), false positive (FP), true negative (TN), or false negative (FN),

$$
\text{A predition } \hat{y}_i \text{ of } y_i \text{ is a } \begin{cases} \text{TP,} & \text{if } (|\hat{y}_i - y_i| \leq 30 \wedge \hat{y}_i \neq \emptyset) \\ \text{FP,} & \text{if } (|\hat{y}_i - y_i| > 30 \wedge \hat{y}_i \neq \emptyset) \\ \text{TN,} & \text{if } (\hat{y}_i = \emptyset \wedge y_i = \emptyset) \\ \text{FN,} & \text{if } (\hat{y}_i = \emptyset \wedge y_i \neq \emptyset) \end{cases}
\tag{18}
$$

Table 1 summarizes these results (TP, FP, TN, FN) for each peak (p_1, p_2, and p_3) using the *individual* approach presented earlier.

Fig. 3 Effect of the number of training samples on the average prediction error (Eq. 17) and standard deviation for different regression models (KSR, SR, Linear, SVM, Extra-Trees) using a fivefold crossvalidation

Table 1 Confusion matrix for individually trained KSR models: true positive (TP), false negative (FN) false positive (FP) and true negative (TN) for the three peaks (P1, P2, and P3)

Actual value	Prediction P1		Prediction P2		Prediction P3	
	Peak	No peak	Peak	No peak	Peak	No peak
Peak	11,223	23	12,542	109	12,663	129
No peak	86	1,482	8	155	6	16

Table 2 Peak recognition accuracy (Eq. 19) obtained by the original MOCAIP algorithm (Gaussian priors) and by the kernel spectral regression (KSR) extension proposed in this paper on a dataset of 13,611 pulses

	P_1 (%)	P_2 (%)	P_3 (%)
KSR (global)	99.37	99.42	99.59
KSR (individual)	99.15	99.09	98.95
Gaussian Priors	97.26	92.84	90.83

The regression models trained globally perform best because they might contain data from the patient to be predicted

Based on these results and the ones obtained by globally trained models, we report the accuracy in Table 2, which is defined as:

$$ACC = (TP + TN)/(TP + FP + TN + FN) \qquad (19)$$

Figure 5 shows the Receiver operating characteristic (ROC) for MOCAIP, individual Kernel Spectral Regression (KSR) models. The plots show the true positive rate (TPR) (sensitivity) against the false positive rate (FPR),

$$TPR = TP/(TP + FN) \qquad (20)$$

$$FPR = FP/(FP + TN) \qquad (21)$$

We can observe that our MOCAIP extension (based on KSR) achieves a very high true positive rate for correctly designating the three peaks. The significant improvement in terms of True Positive and False Positive rate is confirmed by the combined accuracy. Whereas MOCAIP obtains 97.26, 92.84, and 90.83% for each peak, the results of the proposed extension are 99.15, 99.09 and 98.95% on new patients only.

These results are comparable to those obtained by the global regression model: 99.37, 99.42 and 99.59% (Table 2). This indicates that our regression model has a good generalization power and it does not seem to be subject by over-fitting problems. These results were obtained after an optimization of the optimal parameters such that the threshold $t = 44$ and the temperature parameter γ of the RBF kernel is equal to 2.

Figure 6 illustrates successful detection results on four different pulses. We can observe that the detection is robust given the large shape variability of the ICP signal.

4 Discussion

MOCAIP is a recent paradigm for analyzing intracranial pressure pulses in terms of their morphological characteristics. The main contribution of the present work is to significantly improve the performance of the peak designation component of MOCAIP. Peak designation is a challenging and important task that MOCAIP needs to solve because correct morphological feature extraction cannot be achieved without properly designating each peak. The improvement was achieved by using nonlinear regression techniques, particularly, the Kernel Spectral Regression (KSR). The improved MOCAIP algorithm will facilitate towards automatic and robust characterization of ICP pulse morphological changes.

A comparative analysis of different regression methods was described in Sect. 3.1. kernel spectral regression (KSR) and extremely randomized decision Trees (Extra-Trees) yielded the lowest error rates in predicting the position of the peaks. The main drawback of Extra-Trees is the high computational cost during the learning phase. It becomes prohibitive as the number of training samples increases. For these reasons, we decided to select KSR as the regression method employed in the MOCAIP extension.

Furthermore, experimental results presented in Sect. 3.2 validate the method by showing a significant increase of the peak designation accuracy in comparison with the original MOCAIP. The first possible reason for the better performance using a regression approach is that the amplitude information of ICP pulse is incorporated into the peak designation process while such information was not exploited in the original framework. The second possible reason is that the peak designation process in the original MOCAIP algorithm makes an assumption of independence between the positions of the ICP sub-peaks while the regression based approach does not depend on this assumption.

Although the clinical value of the morphological properties extracted by MOCAIP needs further study, it could be argued that these metrics definitely provide more information than the mean ICP, which is prevalently used in current clinical practice. ICP pulses originate from blood pressure along the cerebral vasculature. A particular configuration of sub-peaks in an ICP pulse is influenced by arterial, capillary, and venous blood pressure pulses, as well as their interactions with the three major intracranial compartments, including the cerebral vasculature, the brain tissue, and the cerebrospinal fluid circulatory system. Therefore, it is plausible that ICP pulse morphological changes may provide good indications of changes in any of these three compartments. These can be caused by a variety of pathological events such as the narrowing cerebral
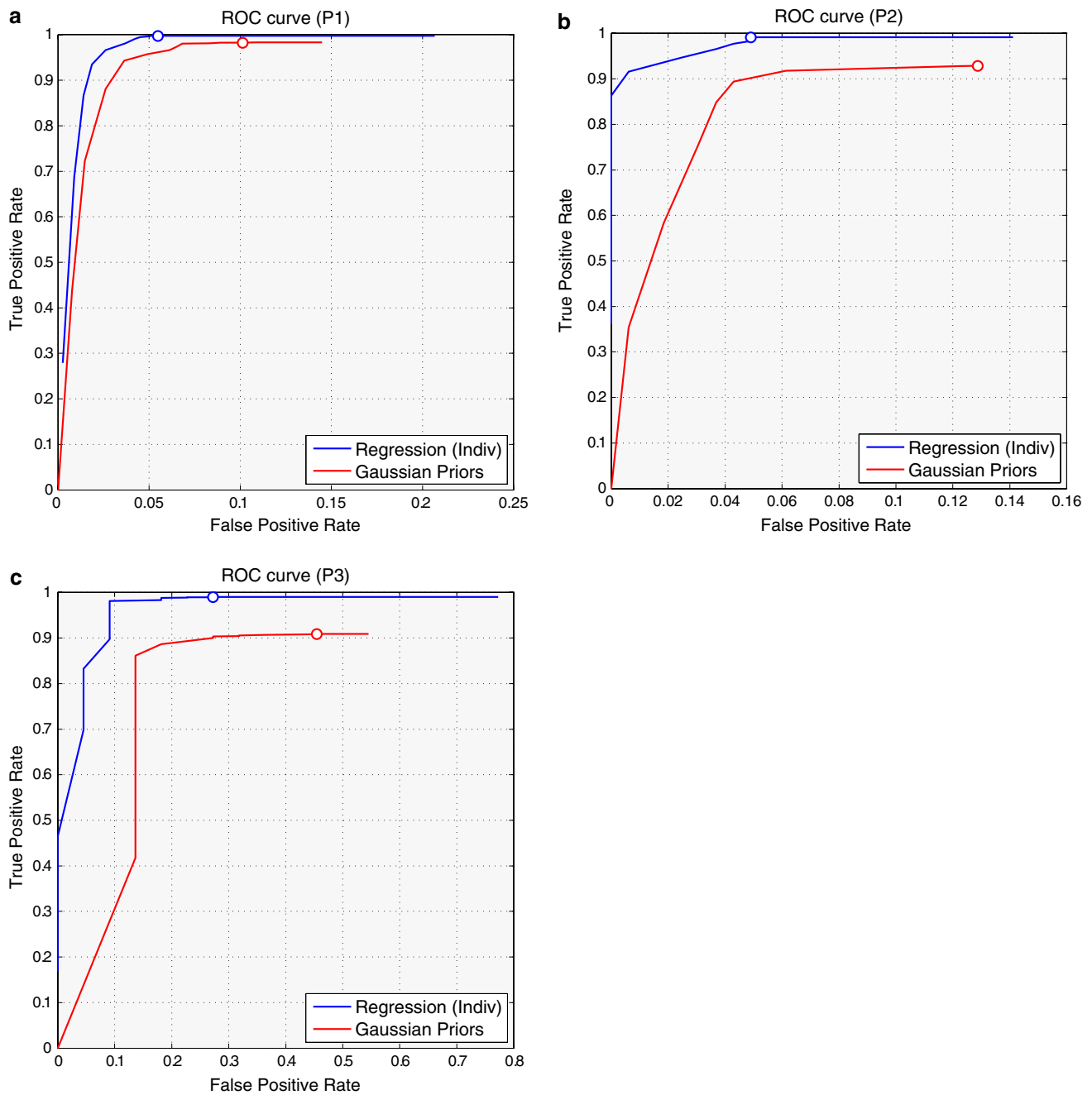
Fig. 5 For each of the three peaks, Receiver operating characteristic (ROC) curves ((**a**), (**b**), and (**c**)) are reported for the original MOCAIP (Gaussian Priors) and the individual trained regression models. We can observe that the use of our regression technique offers a significant improvement in performance for each peak. *Circles* correspond to the true positive and false positive rates obtained by the threshold ($t = 44$ for the regression model) that minimizes the combined accuracy (Eq. 19). Their values are reported in Table 2

arteries (vasospasm) after subarachnoid hemorrhage and development of mass-occupying lesions after a brain injury. Given these considerations, it seems promising for future clinical studies to investigate whether tracking ICP pulse morphological changes in a near real-time fashion can lead to forecasting intracranial pathological changes.

Finally, it can be pointed out that morphological analysis of pulsatile signals is a popular technique for extracting useful information. In addition to intracranial pressure, the morphology of arterial blood pressure pulses has been widely used to extract parameters [12] such as the augmentation index [18] (for characterizing the stiffness of
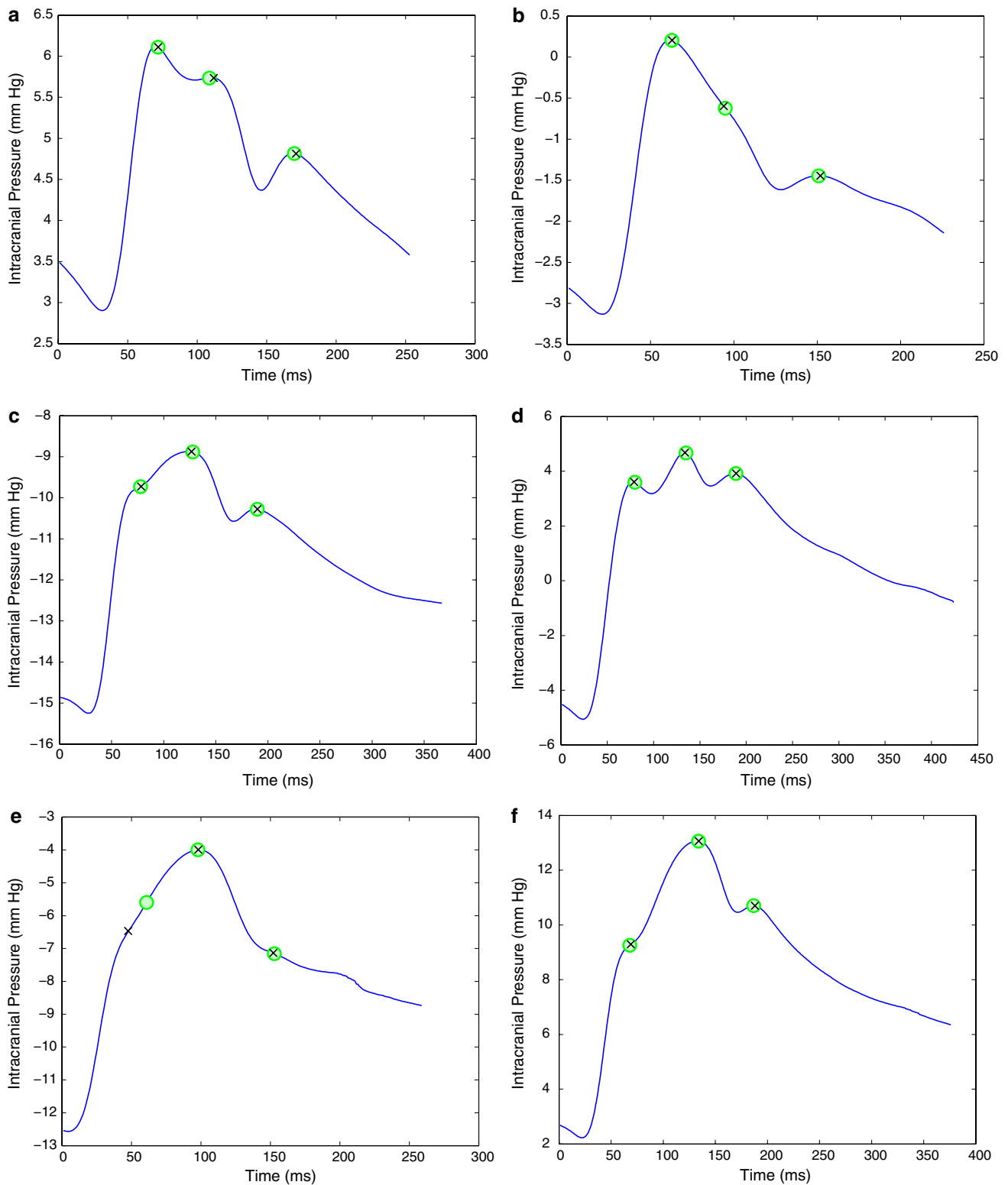
Fig. 6 Detection of peaks on six different ICP pulses (**a**), (**b**), (**c**), (**d**), (**e**), and (**f**). The ground truth is marked as a black cross and the prediction of the MOCAIP regression algorithm is depicted as a green dot

large arteries). It is quite possible that the MOCAIP algorithm may be applicable to analyze arterial blood pressure pulse and other similar pulsatile hemodynamic signals.

References

1. Aboy M, McNames J, Thong T, Tsunami D, Ellenby MS, Goldstein B (2005) An automatic beat detection algorithm for pressure signals. IEEE Trans Biomed Eng 52(10):1662–1670

2. Afonso VX, Tompkins WJ, Nguyen TQ, Luo S (1999) Ecg beat detection using filter banks. IEEE Trans Biomed Eng 46(2):192–202

3. Balestreri M, Czosnyka M, Steiner LA, Schmidt E, Smielewski P, Matta B, Pickard JD (2004) Intracranial hypertension: what additional information can be derived from ICP waveform after head injury?. Acta Neurochir (Wien) 146(2):131–141

4. Cai D, He X, Han J (2008) SRDA: an efficient algorithm for large-scale discriminant analysis. IEEE Trans Knowl Data Eng 20(1):1–12

5. Cardoso ER, Reddy K, Bose D (1988) Effect of subarachnoid hemorrhage on intracranial pulse waves in cats. J Neurosurg 69(5):712–718

6. Cardoso ER, Rowan JO, Galbraith S (1983) Analysis of the cerebrospinal fluid pulse wave in intracranial pressure. J Neurosurg 59(5):817–821

7. Chang C-C, Lin C-J (2001) LIBSVM: a library for support vector machines, 2001, Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm

8. Chatterjee S, Hadi AS (1986) Influential observations, high leverage points and outliers in linear regression. Stat Sci 1:379–393

9. Chopp M, Portnoy HD (1980) Systems analysis of intracranial pressure: comparison with volume–pressure test and csf-pulse amplitude analysis. J Neurosurg 53(4):516–527

10. Contant CF, Robertson CS, Crouch J, Gopinath SP, Narayan RK, Grossman RG (1995) Intracranial pressure waveform indices in transient and refractory intracranial hypertension. J Neurosci Methods 57(1):15–25

11. Czosnyka M, Guazzo E, Whitehouse M, Smielewski P, Czosnyka Z, Kirkpatrick P, Piechnik S, Pickard J (1996) Significance of intracranial pressure waveform analysis after head injury. Acta Neurochir (Wien) 138(5):531–541

12. De Melis M, Morbiducci U, Rietzschel ER, De Buyzere M, Qasem A, Van Bortel L, Claessens T, Montevecchi FM, Avolio A, Segers P (2009) Blood pressure waveform analysis by means of wavelet transform. Med Biol Eng Comput 47(2):165–173

13. Ellis T, McNames J, Aboy M (2007) Pulse morphology visualization and analysis with applications in cardiovascular pressure signals. IEEE Trans Biomed Eng 54(9):1552–1559

14. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63(1):3–42

15. Hu X, Xu P, Lee D, Vespa P, Bergsneider M (2008) An algorithm of extracting intracranial pressure latency relative to electrocardiogram r wave. Physiol Meas 29:459–471

16. Hu X, Xu P, Scalzo F, Vespa P, Bergsneider M (2009) Morphological clustering and analysis of continuous intracranial pressure. IEEE Trans Biomed Eng 56(3):696–705

17. Kaufman L, Rousseeuw PJ (2005) Finding groups in data: an introduction to cluster analysis. Wiley series in probability and mathematical statistics. Wiley, Hoboken, N.J., Leonard Kaufman, Peter J. Rousseeuw

18. Nichols WW, O'Rourke MF (2005) McDonald's blood flow in arteries, 5th edn. Oxford University Press

19. Portnoy HD, Chopp M (1981) Cerebrospinal fluid pulse wave form analysis during hypercapnia and hypoxia. Neurosurgery 9(1):14–27

20. Takizawa H, Gabra-Sanders T, Miller JD (1987) Changes in the cerebrospinal fluid pulse wave spectrum associated with raised intracranial pressure. Neurosurgery 20(3):355–361