A NON-PARAMETRIC METHOD FOR INCURRED BUT NOT REPORTED CLAIM RESERVE ESTIMATION

Hélio Lopes,[1,*] *Jocélia Barcellos,*[1] *Jessica Kubrusly,*[2] *&*
Cristiano Fernandes[3]

[1] *Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil*

[2] *Instituto de Matemática e Estatística, Universidade Federal Fluminense, Niterói, Rio de Janeiro, Brazil*

[3] *Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil*

Original Manuscript Submitted: 05/15/2011; Final Draft Received: 09/21/2011

The number and cost of claims that will arise from each policy of an insurance company's portfolio are unknown. In fact, there is a high degree of uncertainty on how much will ultimately be the cost of claims, not only during the period of inception but also after the contract termination, since there might be future, not yet reported, losses associated with past claims. Therefore, in practice, insurance companies have to protect themselves against the possibility of this ultimate cost by creating an additional reserve known as the incurred but not reported (IBNR) reserve. This work introduces new non-parametric models to IBNR estimation based on kernel methods; namely, support vector regression and Gaussian process regression. These are used to learn certain types of nonlinear structures present in claims data using the residuals produced by a benchmark IBNR estimation model, Mack's chain ladder. The proposed models are then compared to Mack's model using real data examples. Our results show that the three new proposed models are competitive when compared to Mack's benchmark model: they may produce the closest predictions of IBNR and also more accurate estimates, given that the variance for the reserve estimation, obtained through the bootstrap technique, is usually smaller than the one given by Mack's model.

KEY WORDS: *risk analysis, statistical learning, kernel methods, Monte Carlo*

1. INTRODUCTION

When an insurance policy is written it, typically will, cover a defined period from inception. In non-life insurance, claims incurred during this period due to physical damage or theft are often reported and settled quickly [1]. However, in other types of insurance, the time between a claim event and the determination of the amount to pay for this claim can be considerably large. So, on a portfolio of an insurance company the number and cost of claims that will arise from each of its policies are unknown. Indeed, at expiry of a policy there can be a high degree of uncertainty as to what the cost of claims will ultimately be since there might be future, not yet reported, losses associated with past claims. In practice, insurance companies have to protect themselves against the possibility of this ultimate cost by creating an additional reserve known as the incurred but not reported (IBNR) reserve. More precisely, this reserve corresponds to the total amount owed by the insurer to all valid claimants who have had a covered loss but have not yet reported it. A satisfactory estimate of such reserve can be made only through statistical techniques and in this article we propose a new non-parametric method to estimate it.

[*] Correspond to Hélio Lopes, E-mail: lopes@mat.puc-rio.br, URL: http://www.mat.puc-rio.br/ lopes

Several estimators for IBNR reserves have been proposed in the literature since the original work of Tarbell in 1934 [2], where the deterministic chain-ladder method was introduced. Mack derived a stochastic model to explain the chain-ladder method [3, 4] and several recent works have tried to reduce the variance of the reserve entailed by Mack's chain-ladder model (for example [5–7], just to cite a few). The majority of these models are constructed from the runoff triangle, which corresponds to an incomplete $n \times n$ matrix $\mathbf{C}$. In this matrix, only the elements $c_{i,j}$ with $j \leq n - i + 1$ are known, and they correspond to the cumulative paid out amounts with respect to the accident period (month, year,...) i, $i \in \{1, ..., n\}$, up to and including the development period j. To estimate the IBNR reserve, all these cited methods run algorithms to fill the unknown entries of $\mathbf{C}$ using the given data. Table 1 shows an example of a runoff triangle from the historical loss development study related to the automatic facultative general liability (AFG) [8]. The most popular method used to complete the runoff triangle is the chain-ladder, which estimates outstanding claims by projecting into the future a weighted average of past claim developments.

1.1 Contribution

The main contribution of our article is to present a new hybrid model for IBNR reserve estimation. It works in two stages. In the first stage, the method relies on the most popular method for IBNR estimation, the Mack's chain ladder, to obtain a prediction for IBNR reserve. In the second stage, it uses kernel-based regression methods to statistically learn from the residuals of the previous stage, assuming that further nonlinear structure can be estimated from them. The proposed method is very simple to implement and shows good results on real data.

1.2 Paper Outline

Section 2 gives an overview of the Mack's chain-ladder model. Section 3 gives a brief introduction to kernel-based methods for regression. Section 4 introduces the hybrid model. Section 5 shows the results and compares the proposed method to other classical methods. Section 6 proposes a bootstrap strategy to estimate the variance of the new method. Finally, Section 7 concludes the work and points out a future direction for research.

2. THE CHAIN-LADDER IBNR RESERVE ESTIMATOR

Mack was the first to propose a stochastic model for IBNR reserve estimation [3]. This section introduces Mack's chain-ladder (MCL) model, which is the most popular and practical model to solve this claim reserving problem. Mack's model is both very simple to implement and gives accurate results to IBNR estimation.

One can observe that the entries in each row of the runoff triangle $\mathbf{C}$ generally increase along the columns as time progresses. The chain-ladder method uses this observation to estimate the future payments of claims.

TABLE 1: Runoff triangle of the AFG data

Accident year	**Developing year *j***									
i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1981	5012	8269	10907	11805	13539	16181	18009	18608	18662	18834
1982	106	4285	5396	10666	13782	15599	15496	16169	16704	
1983	3410	8992	13873	16141	18735	22214	22863	23466		
1984	5655	11555	15766	21266	23425	26083	27067			
1985	1092	9565	15836	22169	25955	26180				
1986	1513	6445	11702	12935	15852					
1987	557	4020	10946	12314						
1988	1351	6947	13112							
1989	3133	5395								
1990	2063									

The MCL links successive cumulative claims to suitable link ratios. It is a distribution-free stochastic model built upon the following hypotheses:

- Cumulative claims $c_{i,j}$ of different accident period i are independent;
- A Markov chain is formed by $(c_{i,j})_{j\geq 1}$. There exist development factors $f_j > 0$, with $1 \leq j < n$, such that for all $1 \leq i \leq n$ and for all $1 \leq j \leq n$ the following moment conditions hold:

$$E[c_{i,j+1}|c_{i,1}, c_{i,2}, ..., c_{i,j}] = E[c_{i,j+1}|c_{i,j}] = c_{i,j} \cdot f_j,$$

$$\text{Var}[c_{i,j+1}|c_{i,1}, c_{i,2}, ..., c_{i,j}] = \text{Var}[c_{i,j+1}|c_{i,j}] = \sigma_j^2 \cdot c_{i,j}.$$

From the runoff triangle data, the MCL predicts the growing factor f_j from column j to column $j+1$ by use of the following estimator:

$$\hat{f}_j = \frac{\sum_{i=1}^{n-j} c_{i,j+1}}{\sum_{i=1}^{n-j} c_{i,j}} = \sum_{i=1}^{n-j} \frac{c_{i,j}}{\sum_{i=1}^{n-j} c_{i,j}} \times \frac{c_{i,j+1}}{c_{i,j}}.$$

Note that $\hat{f}_j$ is, in fact, a weighted average of the observed individual development factors $c_{i,j+1}/c_{i,j}$.

The variance parameters σ_j^2, for all $j \in \{1, \dots, n-2\}$, are estimated by the following unbiased estimator:

$$\hat{\sigma}_j^2 = \frac{1}{n-j-1} \sum_{i=1}^{n-j} c_{i,j} \left(\frac{c_{i,j+1}}{c_{i,j}} - \hat{f}_j \right)^2. \tag{1}$$

After computing $\hat{f}_j$, $j \in \{1, ..., n-1\}$, the IBNR total reserve can now be computed using the unbiased estimator given in the sequel:

$$\mathbf{IBNR}_{\text{MCL}} = \sum_{i=2}^{n} c_{i,n-i+1} \times (\hat{f}_{n-i+1} \cdot ... \times \hat{f}_{n-1} - 1).$$

Using the example given in Table 1, Table 2 illustrates how the MCL method predicts the unknown entries of the runoff triangle (in italic). Table 3 shows the reserve estimation by the use of the MCL method. For each accident period i, $i \in \{2, ..., n\}$, the reserve R_i is computed by $R_i = \hat{c}_{i,n} - c_{i,n-i-1}$.

To evaluate the uncertainty of the reserve estimation it is common to use the mean quadratic error (MQE). For the MCL IBNR estimator, the MQE is:

$$\text{MQE}(\hat{c}_{i,n}) = E[(\hat{c}_{i,n} - c_{i,n})^2|D], \quad \text{where} \quad D = \{c_{i,j}|i+j \leq n+1\}$$

TABLE 2: Reserve estimation using Mack's chain ladder model on the AFG runoff triangle data.

Accident year	**Development year *j***									
i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	5012	8269	10907	11805	13539	16181	18009	18608	18662	18834
2	106	4285	5396	10666	13782	15599	15496	16169	16704	*16857.95*
3	3410	8992	13873	16141	18735	22214	22863	23466	*23863.43*	*24083.37*
4	5655	11555	15766	21266	23425	26083	27067	*27967.34*	*28441.01*	*28703.14*
5	1092	9565	15836	22169	25955	26180	*27277.85*	*28185.21*	*28662.57*	*28926.74*
6	1513	6445	11702	2935	15852	*17649.38*	*18389.50*	*19001.20*	*19323.01*	*19501.10*
7	557	4020	10946	12314	*14428.00*	*16063.92*	*16737.55*	*17294.30*	*17587.21*	*17749.30*
8	1351	6947	13112	*16663.88*	*19524.65*	*21738.45*	*22650.05*	*23403.47*	*23799.84*	*24019.19*
9	3133	5395	*8758.90*	*11131.59*	*13042.60*	*14521.43*	*15130.38*	*15633.68*	*15898.45*	*16044.98*
10	2063	*6187.67*	*10045.83*	*12767.13*	*14958.92*	*16655.04*	*17353.46*	*17930.70*	*18234.38*	*18402.44*
$\hat{f}_j$	2.999	1.624	1.271	1.172	1.113	1.042	1.033	1.017	1.009	1.000

TABLE 3: IBNR reserve estimation using MCL model.

Accident year	Future payments	Realized payments	Reserve estimation
1	—	—	—
2	16857.95	16704	153.95
3	24083.37	23466	617.37
4	28703.14	27067	1636.14
5	28926.74	26180	2746.74
6	19501.10	15852	3649.10
7	17749.30	12314	5435.30
8	24019.19	13112	10907.19
9	16044.98	5395	10649.98
10	18402.44	2063	16339.44
Total	194288.2	142153	52135.2

In this expression, D represents the set of all known information. Similarly, the MQE for the IBNR reserve of each accident period $\hat{R}_i$ is:

$$\mathrm{MQE}(\hat{R}_i) = E[(\hat{R}_i - R_i)^2|D] = E[(\hat{c}_{i,n} - c_{i,n})^2|D] = \mathrm{MQE}(\hat{c}_{i,n}) \tag{2}$$

Thus, using the fact that $E[X-a]^2 = \mathrm{Var}[X] + (E[X]-a)^2$ and Eq. (2), the MQE of $\hat{R}_i$ can be rewritten as:

$$\mathrm{MQE}(\hat{R}_i) = \mathrm{Var}[R_{i,n}|D] + (E[\hat{c}_{i,n}|D] - c_{i,n})^2 \tag{3}$$

According to Mack [3], the conditional variance $\mathrm{Var}[R_{i,n}|D]$ is given by

$$\left.\begin{aligned} \mathrm{Var}[R_i] &\approx \hat{c}_{i,n}^2 \sum_{j=n-i+1}^{n-1} \frac{\sigma_{j+1}^2}{\hat{f}_{j+1}\hat{c}_{i,j}} \\ \mathrm{Var}[\hat{R}_i] &\approx \hat{c}_{i,n}^2 \sum_{j=n-i+1}^{n-1} \frac{\sigma_{j+1}^2}{\hat{f}_{j+1}\sum_{q=1}^{n-j}\hat{c}_{q,j}} \end{aligned}\right\} \tag{4}$$

Finally, using Eq. (4) and the hypothesis of the model, Mack shows in [3] that the expression for the MQE of $\hat{R}_i$ is given by

$$\mathrm{M\hat{Q}E}(\hat{R}_i) = \hat{c}_{i,n}^2 \sum_{j=n-i+1}^{n-1} \frac{\hat{\sigma}_j^2}{\hat{f}_j^2}\left(\frac{1}{\hat{c}_{i,j}} + \frac{1}{\sum_{t=1}^{n-j} c_{t,j}}\right) \tag{5}$$

where $\hat{c}_{i,j} = c_{i,n-i+1}\hat{f}_{n-i+1}\cdot\cdots\cdot\hat{f}_{j-1}$, with $j > n-i+1$, corresponds to the estimated value of future payments $c_{i,n-i+1}$. The standard error of $\hat{R}_i$ is from now on denoted by $\mathrm{se}(\hat{R}_i) = \sqrt{\mathrm{M\hat{Q}E}(\hat{R}_i)}$.

From this important result obtained by Mack, it is possible to obtain the MQE for the total IBNR reserve. Unfortunately, the standard error of the total IBNR reserve, $\hat{R}_{\text{Total}} = \hat{R}_2 + \cdots + \hat{R}_n$, cannot be obtained directly from the sum of the standard errors of each accident year i, $2 \le i \le n$, since they are correlated thanks to the common development factors $\hat{f}_j$ and $\hat{\sigma}_j^2$. It can be shown that the MQE of the total IBNR reserve is obtained by the following equation:

$$\mathrm{M\hat{Q}E}(\hat{R}_{\text{Total}}) = \sum_{i=2}^{n}\left\{[\mathrm{se}(\hat{R}_i)]^2 + \hat{c}_{i,n}\left(\sum_{k=i+1}^{n}\hat{c}_{k,n}\right)\sum_{j=n+1-i}^{n-1}\frac{\frac{2\hat{\sigma}_j^2}{\hat{f}_j^2}}{\sum_{t=1}^{n-j} c_{t,j}}\right\} \tag{6}$$

Mack's chain-ladder model can be found in the chain-ladder package [8] for the R platform (a public domain software for computational statistics and data analysis (for more details see http://www.r-project.org).

3. KERNEL-BASED METHOD FOR REGRESSION

Kernel-based methods for regression are universal learning machines for solving multidimensional scalar value prediction and estimation problems. These methods received a lot of attention in the machine-learning community since they are very well grounded on a statistical learning theory, called the Vapnik-Chervonenkis (VC) theory [9]. Its consistency conditions, convergence, generalization abilities, and implementation efficiency have been studied by several authors during the last four decades (see [9–11]). This section describes the two most used kernel-based methods for regression: the ε-Support Vector Regression (ε-SVR) and the Gaussian Process Regression (GPR).

3.1 The ε-Support Vector Regression Method

Let $\mathcal{D}$ be the training set, containing l samples $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ are the input data and $y_i \in \mathbb{R}$ are the target values. The SVR method first maps the data $\mathbf{x} \in \mathbf{R}^d$ into some a priori chosen Hilbert space $\mathcal{F}$, called the feature space, via a nonlinear function $\phi : \mathbf{R}^d \to \mathcal{F}$. In this feature space, the prediction function is formulated by the affine equation:

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \tag{7}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathcal{F}$, $\mathbf{w} \in \mathcal{F}$ and $b \in \mathbb{R}$.

In the ε-SVR method, the problem of learning is to find the best function; i.e., the values of $\mathbf{w}$ and b that minimize the following functional:

$$R(\mathbf{w}, b) = \frac{1}{l} \sum_{i=1}^{l} |y_i - f(\mathbf{x})|_\varepsilon + \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle$$

$$\text{where} \qquad |a|_\varepsilon = \begin{cases} 0 & \text{if } |a| < \varepsilon \\ |a| - \varepsilon & \text{otherwise.} \end{cases}$$

The solution $f(\mathbf{x})$ of this ε-SVR problem minimizes the deviation from $|f(\mathbf{x}_i) - y_i|$ for $i = 1, \ldots, l$, while being as flat as possible. Observe that the deviation is controlled by the loss function $| \cdot |_\varepsilon$, called ε-insensitive since it considers the loss equal to zero when the deviation is less than ε. The flatness is due to the second term of functional $R(\mathbf{w}, b)$, which penalizes the size of $\mathbf{w}$.

In the ε-SVR learning method, the values of $\mathbf{w}$ and b are determined by the following minimization problem:

$$\begin{array}{lll} \text{Minimize}_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 + P \cdot \sum_{i=1}^{l} (\xi_i + \hat{\xi}_i) & \\ \text{subject to:} & \xi_i, \hat{\xi}_i \geq 0 & i = 1, \ldots, l \\ & (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) - y_i \leq \varepsilon + \xi_i & i = 1, \ldots, l \\ & y_i - (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \leq \varepsilon + \hat{\xi}_i & i = 1, \ldots, l \end{array}$$

where P is a constant parameter that penalizes the ε-insensitive errors. The errors occurring if $f(\mathbf{x}_i)$ is above (respectively, below) y_i are represented by the ξ_i (respectively, $\hat{\xi}_i$) slack variables.

One can rewrite this optimization problem in its dual form by using Lagrange multipliers $\alpha_i, \hat{\alpha}_i$:

$$\begin{array}{ll} \text{Maximize}_{\alpha_i, \hat{\alpha}_i} & \sum_{i=1}^{l} (\hat{\alpha}_i - \alpha_i) y_i - \varepsilon \sum_{i=1}^{l} (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \\ \text{subject to:} & \sum_{i=1}^{l} (\hat{\alpha}_i - \alpha_i) = 0 \\ & 0 \leq \alpha_i, \hat{\alpha}_i \leq P \end{array} \tag{8}$$

This is a convex quadratic programming problem; therefore, it has a unique global solution. Let $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_l^*)$, $\hat{\alpha}^* = (\hat{\alpha}_1^*, \hat{\alpha}_2^*, \ldots, \hat{\alpha}_l^*)$ denote the optimal solution of the dual problem.

The complementary Karush Kuhn-Tucker conditions for this dual problem at the optimal solution are

$$\begin{aligned}
&\mathbf{w} - \sum_{i=1}^{l}(\hat{\alpha}_i^* - \alpha_i^*)\phi(\mathbf{x}_i) = 0 \\
&\alpha_i^*(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b - y_i - \varepsilon - \xi_i) = 0 \quad && i = 1, \ldots, l \\
&\hat{\alpha}_i^*(y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - b - \varepsilon - \hat{\xi}_i) = 0 \quad && i = 1, \ldots, l \\
&\hat{\alpha}_i^* \cdot \alpha_i^* = 0, \ \ \hat{\xi}_i \cdot \xi_i = 0 && i = 1, \ldots, l \\
&(\hat{\alpha}_i^* - P)\hat{\xi}_i = 0, \ (\alpha_i^* - P)\xi_i = 0 && i = 1, \ldots, l
\end{aligned} \tag{9}$$

These complementary condition formulas show several important and suitable properties of the ε-SVR learning method. The first equation means that at the optimal solution $\mathbf{w}^\star$ for the primal problem is a linear combination of the input points mapped to the feature space. Since

$$\mathbf{w}^\star = \sum_{i=1}^{l}(\hat{\alpha}_i^\star - \alpha_i^\star)\phi(\mathbf{x}_i),$$

then Eq. (7) can be rewritten as

$$f(\mathbf{x}) = \sum_{i=1}^{l}(\hat{\alpha}_i^\star - \alpha_i^\star)\langle \phi(\mathbf{x}_i), \phi(\mathbf{x})\rangle + b^\star. \tag{10}$$

where b^* is chosen so that $f(\mathbf{x}_i) - y_i = -\varepsilon$ for any i such that $\alpha_i^\star \in (0, P/l)$.

The other set of equations in Eq. (9) say that when $\alpha_i^\star$ and $\hat{\alpha}_i^\star$ are both equal to zero the scalar function prediction for the input point $\mathbf{x}_i$ distances from the target value y_i less than ε. The input points $\mathbf{x}_i$ in which one of the associated $\alpha_i^\star$ or $\hat{\alpha}_i^\star$ does not vanish are called the support vectors.

This method is available in the kernlab package [12] for the R platform. A more detailed presentation about ε-SVR can be found in [13].

3.2 Kernel Functions

The a priori chosen non-linear function ϕ, mapping the input point to the feature space, appears in two places: one is as the objective function of the ε-SVR dual optimization problem (8) as $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$, and the other is as the prediction function f in Eq. (10) as $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x})\rangle$. Notice that in both cases it is sufficient to know how to compute the inner-product of two points mapped to feature space by ϕ. Thus, a suitable and efficient way to do that is through the use of the so-called kernel functions.

Kernel functions have been recognized as important tools in several numerical analysis applications, including approximation, interpolation, meshless method for solving differential equations, and in machine learning [14].

A kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is defined as follows:

$$k(\mathbf{z}, \mathbf{w}) = \langle \phi(\mathbf{z}), \phi(\mathbf{w})\rangle.$$

Using this definition, the prediction function in Eq. (10) is better written as

$$f(\mathbf{x}) = \sum_{i=1}^{l}(\hat{\alpha}_i^\star - \alpha_i^\star)k(\mathbf{x}_i, \mathbf{x}) + b^\star. \tag{11}$$

In fact, kernel functions implicitly represent the mapping ϕ to the feature space $\mathcal{F}$. For example, consider that $\mathbf{z}$ and $\mathbf{w}$ are in $\mathbb{R}^2$. Also, consider the non-linear mapping $\phi : \mathbb{R}^2 \to \mathbb{R}^3$ as $\phi(\mathbf{z}) = (z_1^2, z_2^2, \sqrt{2}z_1 z_2)$. Then,

$$\langle \phi(\mathbf{z}), \phi(\mathbf{w})\rangle = z_1^2 w_1^2 + z_2^2 w_2^2 + 2z_1 z_2 w_1 w_2 = (\langle \mathbf{z}, \mathbf{w}\rangle)^2.$$

In conclusion, it is more efficient and more suitable to choose kernels rather than non-linear mappings ϕ. However, not all functions k represent an inner product in the feature space. Mercer's theorem characterizes these functions [9]. Some examples of kernel functions that satisfy Mercer's conditions are:

- Polynomial kernel [15]: $k(\mathbf{z}, \mathbf{w}) = (1 + \langle \mathbf{z}, \mathbf{w} \rangle)^d$.
- Gaussian kernel [15]: $k(\mathbf{z}, \mathbf{w}) = e^{-(\|\mathbf{z}-\mathbf{w}\|^2/2\sigma^2)}$.
- Wavelet kernel [16]: $k(\mathbf{z}, \mathbf{w}) = \prod_{i=1}^{n} h\left(\frac{z_i - w_i}{\sigma}\right)$, where $h(u) = \cos(1.75u)e^{-(u^2/2)}$.
- Fourier kernel [17]: $k(\mathbf{z}, \mathbf{w}) = \prod_{i=1}^{n} g\left(\frac{z_i - w_i}{\sigma}\right)$, where $g(u) = \frac{1-q^2}{2[1-2q\cos(u)]+q^2}$.

3.3 The Gaussian Process Regression Method

The GPR is another supervised statistical learning method for regression. Similar to the ε-SVR, it considers as an input the training data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,l}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Its objective also is to build a function $g : \mathbb{R}^n \to \mathbb{R}$ that approximates the input data. Under Gaussian process assumptions (see [10] for further details) the predictive mean value at a point $\mathbf{x}_\star$ is given by

$$g(\mathbf{x}_\star) = \mathbf{k}^T(\mathbf{x}_\star)(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

where $\mathbf{K}$ denotes the $d \times d$ matrix of covariances between the training points with entries $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$; $\mathbf{k}(\mathbf{x}_\star)$ is the vector of covariances such that the ith entry is $k(\mathbf{x}_i, \mathbf{x}_\star)$, where k is a kernel function; σ^2 is the noise variance on the observations; and $\mathbf{y}$ is the l-dimensional vector containing the training targets. Moreover, the predictive variance value at a point $\mathbf{x}_\star$ is given by

$$\mathrm{Var}[g(\mathbf{x}_\star)] = k(\mathbf{x}_\star, \mathbf{x}_\star) - \mathbf{v}^T\mathbf{v},$$

where $\mathbf{v}$ is the solution of the system $\mathbf{L}\mathbf{v} = \mathbf{k}(\mathbf{x}_\star)$, and $\mathbf{L}$ is the Cholesky decomposition of $(\mathbf{K} + \sigma^2\mathbf{I})$.

This method is also implemented in the `kernlab` package for `R` [12].

4. A HYBRID CHAIN-LADDER MODEL

In this section we present a hybrid model composed of a Mack's chain-ladder component and a kernel-based method component, to model, respectively, linear and nonlinear patterns contained in the runoff triangle. It might be the case that the behavior of the IBNR claims data is not best captured by a linear estimator such as Mack's chain-ladder model. So, a hybrid strategy that combines both linear and nonlinear structures present in the runoff triangle may be shown to be a good alternative.

Our model strategy acknowledges the fact that each known entry $c_{i,j+1}$ generally does not correspond to the value of $c_{i,j} \cdot \hat{f}_j$. The difference between these two values is what we call the residual at $c_{i,j+1}$.

The hybrid IBNR total reserve model $\mathbf{IBNR}_{\text{hybrid}}$ can be represented as follows:

$$\mathbf{IBNR}_{\text{hybrid}} = \mathbf{IBNR}_{\text{MCL}} + \mathbf{IBNR}_{\text{KBM}},$$

where $\mathbf{IBNR}_{\text{MCL}}$ is the reserve estimated by the linear Mack's chain-ladder model while $\mathbf{IBNR}_{\text{KBM}}$ is the reserve obtained by fitting the nonlinear statistical learning process. Both estimates are obtained by use of the runoff triangle data: Mack's on the original triangle data while $\mathbf{IBNR}_{\text{KBM}}$ is obtained by fitting the statistical learning process to the residuals of Mack's model.

The unknown entries $c_{i,j}$ of the runoff triangle $\mathbf{C}$ are then obtained by

$$\hat{\hat{c}}_{i,j} = \hat{c}_{i,j} + \hat{\psi}_{i,j},$$

where $\hat{c}_{i,j}$ is the prediction by the Mack's chain ladder and $\hat{\psi}_{i,j}$ is the residue learned by the kernel-based regression.

Next, we propose three different strategies to build the training data for the kernel-based regression method. These strategies differ on how the set $\mathcal{D}$ will be constructed, to be more specific, on how to build the $l \times n$ matrix $\mathbf{X}$ whose ith row corresponds to the transpose of an input vector $\mathbf{x}_i$, and the vector $\mathbf{y}$ whose i^{th} entry corresponds to the target value y_i. Each of these three strategies will define a new hybrid model. As we will see, the first model behaves as a multiplicative nonlinear correction to Mack's linear model, while the last two represent additive nonlinear corrections.

4.1 Hybrid Model 1

This first model uses kernel-based methods for regression to build an approximation function that models the residuals through the function $\Psi_1 : \mathbb{R}^2 \rightarrow \mathbb{R}$. In this model, the hybrid method tries to correct the estimation for $c_{i,j+1}$ by learning the ratio $f_{i,j}/\hat{f}_j$, where $f_{i,j} = c_{i,j+1}/c_{i,j}$.

Here, the ith row of the $l \times 2$ matrix $\mathbf{X}$ corresponds to the transposed bi-dimensional vector $\mathbf{x}^T = (f_{i-1,j}/\hat{f}_j, f_{i,j-1}/\hat{f}_{j-1})$, and each entry of the target l-dimensional vector $\mathbf{y}$ is defined by $\mathbf{y} = f_{i,j}/\hat{f}_j - 1$, where $2 \leq i \leq n-2$, $2 \leq j \leq n-i$ and $n \times n$ is the dimension of the runoff triangle. Since the training set should be constructed using only the known part of the triangle, it follows that the size l of the training set depends on n and for this strategy it corresponds to

$$l = \sum_{i=2}^{n-2} \sum_{j=2}^{n-i} 1 = \sum_{i=2}^{n-2} n - i - 1 = \frac{(n-2)(n-3)}{2} = \frac{n^2 - 5n + 6}{2}. \tag{12}$$

In fact, this choice for the training data tries to capture a nonlinear multiplicative correction for the estimated development factor $\hat{f}_j$ on each unknown entry of the runoff triangle, since the value of the unknown entry $c_{i,j}$ will be estimated by $\hat{\hat{c}}_{i,j+1} = \hat{c}_{i,j} + \Psi_1(f_{i-1,j}/\hat{f}_j, f_{i,j-1}/\hat{f}_{j-1}) \cdot \hat{c}_{i,j} \cdot \hat{f}_j$.

To illustrate the construction for this and the coming models, consider a small runoff triangle A given by

$$A = \left|\begin{array}{lllll} 5012 & 8269 & 10907 & 11805 & 13539 \\ \ 106 & 4285 & \ 5396 & 10666 & \\ 3410 & 8992 & 13873 & & \\ 5655 & 11555 & & & \\ 1092 & & & & \end{array}\right|. \tag{13}$$

Since $n = 5$, it follows that l is equal to 3. Mack's estimators for the growing factor are $\hat{f}_1 = 2,33$, $\hat{f}_2 = 1,40$, and $\hat{f}_3 = 1,37$; and from the triangle one can compute the values $f_{1,2} = 1,32$, $f_{2,1} = 40,42$, $f_{1,3} = 1,08$, $f_{2,2} = 1,26$, $f_{3,1} = 2,64$, $f_{2,3} = 1,98$, and $f_{3,2} = 1,54$. Therefore, the matrix $\mathbf{X}$ and the vector $\mathbf{y}$ are given by

$$\mathbf{X} = \begin{pmatrix} \dfrac{f_{1,2}}{\hat{f}_2} & \dfrac{f_{2,1}}{\hat{f}_1} \\ \dfrac{f_{1,3}}{\hat{f}_3} & \dfrac{f_{2,2}}{\hat{f}_2} \\ \dfrac{f_{2,2}}{\hat{f}_2} & \dfrac{f_{3,1}}{\hat{f}_1} \end{pmatrix} = \begin{pmatrix} 0,94 & 17,35 \\ 0,79 & 0,90 \\ 0,90 & 1,13 \end{pmatrix} \text{ and } \mathbf{y} = \begin{pmatrix} \dfrac{f_{2,2}}{\hat{f}_2} - 1 \\ \dfrac{f_{2,3}}{\hat{f}_3} - 1 \\ \dfrac{f_{3,2}}{\hat{f}_2} - 1 \end{pmatrix} = \begin{pmatrix} 0.90 - 1 \\ 1,45 - 1 \\ 1,10 - 1 \end{pmatrix}.$$

4.2 Hybrid Model 2

This second model uses kernel-based methods for regression to build an approximation function that models the residuals obtained from fitting Mack's model to the original runoff triangle. This function is defined by $\Psi_2 : \mathbb{R}^3 \rightarrow \mathbb{R}$. To build this approximation this new model has to fill the training matrix $\mathbf{X}_{l \times 3}$ and the target vector $\mathbf{y}_{3 \times 1}$ according to the following strategy:

- Each row of $\mathbf{X}$ is the transpose of a three-dimensional vector $\mathbf{x}^T = (c_{i-1,j}, c_{i-1,j+1}, c_{i,j})$, for $2 \leq i \leq n-1$ and $1 \leq j \leq n-i-1$.

- Each entry of $\mathbf{y}$ corresponds to $y = c_{i,j+1} - \hat{f}_j * c_{i,j}$, for $1 \leq i \leq n-2$ and $1 \leq j \leq n-i-1$.

Considering this, the number of training elements on $\mathcal{D}$ is given by

$$l = \sum_{i=2}^{n-1} \sum_{j=1}^{n-i-1} 1 = \sum_{i=1}^{n-2} (n-i-1) = \frac{(n-2)(n-1)}{2} = \frac{n^2-3n+2}{2}. \tag{14}$$

Notice that this model, by using three neighbors of $c_{i,j+1}$, tries to capture the nonlinearity not only between the columns but also between the lines. Therefore, it is an additive correction to the developing factor estimator $\hat{f}_j$.

For the runoff triangle A, the matrix $\mathbf{X}$ and the vector $\mathbf{y}$ correspond to

$$\mathbf{X} = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{2,1} \\ A_{1,2} & A_{1,3} & A_{2,2} \\ A_{1,3} & A_{1,4} & A_{2,3} \\ A_{2,1} & A_{2,2} & A_{3,1} \\ A_{2,2} & A_{2,3} & A_{3,2} \\ A_{3,1} & A_{3,2} & A_{4,1} \end{pmatrix} = \begin{pmatrix} 5012 & 8269 & 106 \\ 8269 & 10907 & 4285 \\ 10907 & 11805 & 5396 \\ 106 & 4285 & 3410 \\ 4285 & 5396 & 8992 \\ 3410 & 8992 & 5655 \end{pmatrix} \text{ and } \mathbf{y} = \begin{pmatrix} \hat{\psi}_{1,1} \\ \hat{\psi}_{1,2} \\ \hat{\psi}_{1,3} \\ \hat{\psi}_{2,1} \\ \hat{\psi}_{2,2} \\ \hat{\psi}_{3,1} \end{pmatrix} = \begin{pmatrix} 4038,02 \\ -603,00 \\ 3273,48 \\ 1046,70 \\ 984,20 \\ -1621,15 \end{pmatrix}.$$

4.3 Hybrid Model 3

This final model is very similar to the previous Model 2, and uses kernel-based methods for regression to build an approximation function that models the residuals through the function $\Psi_3 : \mathbb{R}^4 \to \mathbb{R}$. To build Ψ_3 it fills the training matrix $\mathbf{X}_{l\times 4}$ and the target vector $\mathbf{y}$ according to the following strategy:

- Each row of $\mathbf{X}$ is the transpose of a four-dimensional vector $\mathbf{x} = (c_{i,j}, c_{i,j+1}, c_{i+1,j}, \hat{f}_j)$, for $1 \leq i \leq n-2$ and $1 \leq j \leq n-i-1$.
- Each entry of $\mathbf{y}$ corresponds to $y = c_{i,j+1} - \hat{f}_j * c_{i,j}$, for $2 \leq i \leq n-1$ and $1 \leq j \leq n-i-1$

The number of training elements of $\mathcal{D}$ for this model also is given by Eq. (14).

It chooses three elements around $c_{i,j+1}$ and the factor $\hat{f}_j$ in order to capture data nonlinearity, not only between columns but also between lines. It is also an additive correction strategy for $\hat{f}_j$.

For the runoff triangle A, the matrix $\mathbf{X}$ and the vector $\mathbf{y}$ correspond to

$$\mathbf{X} = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{2,1} & \hat{f}_1 \\ A_{1,2} & A_{1,3} & A_{2,2} & \hat{f}_2 \\ A_{1,3} & A_{1,4} & A_{2,3} & \hat{f}_3 \\ A_{2,1} & A_{2,2} & A_{3,1} & \hat{f}_1 \\ A_{2,2} & A_{2,3} & A_{3,2} & \hat{f}_2 \\ A_{3,1} & A_{3,2} & A_{4,1} & \hat{f}_1 \end{pmatrix} = \begin{pmatrix} 5012 & 8269 & 106 & 2,33 \\ 8269 & 10907 & 4285 & 1,40 \\ 10907 & 11805 & 5396 & 1,37 \\ 106 & 4285 & 3410 & 2,33 \\ 4285 & 5396 & 8992 & 1,40 \\ 3410 & 8992 & 5655 & 2,33 \end{pmatrix} \text{ and } \mathbf{y} = \begin{pmatrix} \hat{\psi}_{1,1} \\ \hat{\psi}_{1,2} \\ \hat{\psi}_{1,3} \\ \hat{\psi}_{2,1} \\ \hat{\psi}_{2,2} \\ \hat{\psi}_{3,1} \end{pmatrix} = \begin{pmatrix} 4038,02 \\ -603,00 \\ 3273,48 \\ 1046,70 \\ 984,20 \\ -1621,15 \end{pmatrix}.$$

4.4 The Learning Process

After building the training set $\mathcal{D}$, the next step in the proposed methodology for IBNR estimation is to obtain the modeling function Ψ_1, Ψ_2, or Ψ_3. It is done by the use of the kernel-based methods for regression described in Section 3. From these three functions, the nonlinear part of the hybrid IBNR reserve estimator will be given by

$$\mathbf{IBNR}_{KBM} = \sum_{i=2}^{n} \sum_{j=n-i+1}^{n} \hat{\psi}_{i,j}, \tag{15}$$

where the value of $\hat{\psi}_{i,j+1}$ is calculated for each model as follows:

Hybrid Model 1: $\hat{\psi}_{i,j+1} = \Psi_1(f_{i-1,j}/\hat{f}_j, f_{i,j-1}/\hat{f}_{j-1}) \cdot \hat{f}_j \cdot \hat{c}_{i,j}$.

Hybrid Model 2: $\hat{\psi}_{i,j+1} = \Psi_2(\hat{c}_{i,j}, \hat{c}_{i,j+1}, \hat{c}_{i+1,j})$.

Hybrid Model 3: $\hat{\psi}_{i,j+1} = \Psi_3(\hat{c}_{i,j}, \hat{c}_{i,j+1}, \hat{c}_{i+1,j}, \hat{f}_j)$.

5. RESULTS

This section compares the results of fitting the three proposed hybrid models to Mack's chain-ladder model using some empirical data. It starts by considering the runoff triangle of the automatic factultative business in general liability (RAA) data set illustrated in Table 1 and also the ABC data set in Table 4, which presents a runoff triangle of a worker's compensation portfolio of a large company (both are available in the ChainLadder package). Tables 5–7 show the results for the three hybrid models using the ε-SVR and GPR methods to learn the nonlinear part of the IBNR total reserve and compare them to Mack's chain-ladder estimate.

One can notice that all proposed hybrid models produce similar results when compared to the MCL benchmark model for IBNR reserve estimation. In particular, in all data and specifications, they make corrections that result in

TABLE 4: Runoff triangle of the ABC data

Accident year	**Developing year *j***										
i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1977	153638	342050	476584	564040	624388	666792	698030	719282	735904	762544	762544
1978	178536	404948	563842	668528	739976	787966	823542	848360	871022	889022	
1979	210172	469340	657728	780802	864182	920268	958764	992532	1019932		
1980	211448	464930	648300	779340	858334	918566	964134	1002134			
1981	219810	486114	680764	800862	888444	951194	1002194				
1982	205654	458400	635906	765428	862214	944614					
1983	197716	453124	647772	790100	895700						
1984	239784	569026	833828	1024228							
1985	326304	798048	1173448								
1986	420778	1011178									
1987	496200										

TABLE 5: IBNR total estimation using the Hybrid Model 1

	IBNR total estimation using the Hybrid Model 1		
	GPR	ε-SVR	MCL
RAA	53615,38	53615,50	52135,23
ABC	5435057,00	5435057,00	5277760,36

TABLE 6: IBNR total estimation using the Hybrid Model 2

	IBNR total estimation using the Hybrid Model 2		
	GPR	ε-SVR	MCL
RAA	53693,25	53891,09	52135,23
ABC	5493195,00	5505984,00	5277760,36

TABLE 7: IBNR total estimation using Hybrid Model 3

	IBNR total estimation using Hybrid Model 1		
	GPR	ε-SVR	MCL
RAA	53815,45	52927,86	52135,23
ABC	5435057,00	5435057,00	5277760,36

a total IBNR reserve estimate higher than the one predicted by the MCL model. Since the observed IBNRs for these two triangles are not available, it is not possible to judge which one produces the best estimate.

Finally, as a final exercise in comparing the three proposed models to the MCL model, we will use a runoff triangle obtained from an insurance company in Brazil. This triangle has 48 lines, and in order to save some data to allow for out-of-sample comparisons, only the first 24 lines of this triangle will be used for model fitting. To improve comparisons among the different models these will be fitted to different sizes of the runoff triangle, and the observed loss values will be represented by the label "Loss," which corresponds to the the sum of the last column of the $n \times n$ runoff triangle minus the first element of the column. In Table 8 the three hybrid models and the MCL model are run using the n first columns of the runoff triangle, with n varying from 8 to 24. Notice that MCL gives the best approximation only in six out of the 17 runoff triangles and that in the majority of the cases the results of ε-SVR or GPR are very similar.

Since these three hybrid models are very easy to implement and have low computational complexity, they may be adopted by practitioners in the insurance industry as a complement to Mack's chain-ladder model.

For all results presented above, the ε-SVR and the GPR methods of the kernlab package were run using automatic parameter selection. In both cases, the kernel adopted was the Gaussian and the input data were normalized with the corresponding procedure.

6. BOOTSTRAPPING

Since it is very difficult to obtain a closed formula for the variance of each of the IBNR hybrid model estimators, we have chosen to use the boostrap method to obtain a numerical approximation of these variances. To implement this procedure several scenarios for the runoff triangle were generated by simulation according to the following steps:

1. The initial incremental triangle **I** is obtained from the cumulative runoff triangle **C** according to the following rule: $\mathbf{I}_{i,j} = c_{i,j+1} - c_{i,j}$, for $1 \leq i \leq n$ and $1 \leq j \leq n - i + 1$.

TABLE 8: Results for the Hybrid Models 1, 2, and 3, using GPR and ε-SVR, and MCL model. The estimated values in bold are those closest to the observed loss (true) values

	hybrid 1		**hybrid 2**		**hybrid 3**			
n	**GPR**	**SVR**	**GPR**	**SVR**	**GPR**	**SVR**	**MCL**	**Loss**
8	3252.29	3252.29	3484.07	3406.03	3496.95	3437.64	**2929.94**	2198.06
9	3121.79	3121.83	3225.91	3194.21	3236.97	3219.19	**2992.19**	2403.74
10	3036.09	3036.10	3165.02	3097.86	3155.12	3168.56	**2765.33**	2702.46
11	2079.09	2079.11	2107.38	2091.94	**2119.86**	2102.65	2035.77	3033.43
12	**2388.14**	2388.09	2312.32	2269.15	2311.39	2300.87	2371.65	3026.79
13	2786.47	2786.45	2879.77	**2892.97**	2871.61	2856.53	2554.99	3433.79
14	3217.77	3217.75	3199.19	3113.77	3144.23	3143.39	**3084.32**	2858.02
15	3341.42	3341.40	3364.38	3383.72	3348.54	3535.35	**3271.88**	2854.45
16	3408.84	3408.82	3511.09	3556.08	3231.53	3556.08	**3214.53**	3095.06
17	3241.46	**3241.48**	3227.45	3220.05	3227.45	3186.67	3191.01	3332.77
18	3215.45	**3215.46**	3201.01	3167.12	3212.84	3179.60	3126.63	3622.55
19	3537.56	3537.52	3547.58	3951.65	**3550.36**	3507.41	3512.36	4239.08
20	4003.22	**4003.25**	3979.25	4003.25	3989.96	3968.46	3931.84	4252.08
21	4137.46	4137.48	4133.45	4137.16	**4144.92**	4109.03	4020.38	4637.02
22	4771.62	4771.63	4693.65	**4690.57**	4694.25	4671.22	4740.65	4506.90
23	4035.20	4035.24	3994.76	**4051.30**	4006.66	4018.47	4012.00	4446.52
24	3714.77	**3714.78**	3642.43	3684.85	3668.85	3677.19	3591.91	4626.28

2. A bootstrap sample of the incremental triangle **I** is obtained by resampling with replacement the entries in each column, given the origin to a new cumulative runoff triangle. We assume that in each column the individual increments are independent from each other.

3. From all resampled incremental triangles **I**, a new cumulative triangle **C** is computed, and then a new estimative for the total IBNR reserve is computed.

This procedure is repeated several times. By doing, so it is possible to measure the accuracy of the hybrid model estimates and compare them to the MCL model, since it is now possible to numerically evaluate the MQE associated with the hybrid models. Table 9 shows the standard deviation estimation for the total IBNR reserve after simulating 999 runoff triangles on the 24×24 real data via bootstrapping. In Table 9 the three hybrid models are run with the GPR learning strategy using automatic choice for the parameters and normalized input data. In all cases the hybrid models gave better results when compared to the MCL model.

TABLE 9: Standard deviation estimate using bootstrapping

n	Hybrid 1	Hybrid 2	Hybrid 3	MCL
8	256.87	260.01	266.25	570.83
9	239.46	242.08	225.56	579.30
10	208.87	209.23	209.86	495.36
11	177.38	178.94	171.67	377.11
12	164.21	157.31	157.99	431.72
13	165.54	166.81	167.67	423.05
14	194.19	188.52	198.23	475.36
15	198.19	201.82	201.65	540.45
16	204.54	213.59	209.88	496.23
17	180.99	195.70	198.77	473.37
18	197.77	204.04	398.42	430.05
19	200.27	212.57	199.48	513.19
20	225.46	227.76	227.35	555.77
21	216.86	230.62	220.72	564.40
22	263.24	250.58	261.15	637.92
23	217.51	240.26	228.73	475.21
24	215.89	227.04	226.64	428.68

7. CONCLUSION

This work proposed a two-stage hybrid method for IBNR reserve estimation. In the first stage, the method uses Mack's chain-ladder model to obtain a reserve prediction, and in the second stage it uses kernel-based regression methods to statistically learn form the residuals obtained by fitting of Mack's model. Statistical learning from these residuals is accomplished by use of three different strategies combining the known entries of the runoff triangle. The method is very simple to implement and shows promising results based on the empirical exercises presented in this article. The authors plan to carry out further studies in order to obtain a theoretical expression for the IBNR variance estimator for the hybrid models.

ACKNOWLEDGMENTS

The authors would like to thank CAPES, CNPq, and FAPERJ for financial support.

REFERENCES

1. Boland, P. J., *Statistical and Probabilistic Methods in Actuarial Science*, CRC, Boca Raton, FL, 2006.

2. Tarbell, T. F., Incurred but not reported claim reserve, *Proc. of Casualty Actuarial Society Forum*, Parte II, 1934.
3. Mack, T., Distribution-free calculation of the standard error of chain ladder reserve estimates, *Austin Bull.*, 23(3)(76):443–518, 1993.
4. Mack, T., Which stocaschic model is underlying the chain ladder method?, *Insur. Math. Econ.*, 15(2-3):133–138, 1994.
5. England, P. D. and Verrall, R. J., Stocasthic claims reserving in general insurance, *Br. Actuar. J.*, 8:281–293, 2002.
6. Herbst, T., An application of randomly truncated data models in reserving IBNR claims, *Insur. Math. Econ.*, 25(2):123–131, 1999.
7. Doray, L. G., UMVUE of the IBNR reserve in a lognormal linear regression model, *Insur. Math. Econ.*, 18(1):43–57, 1996.
8. Gesmann, M., The ChainLadder package—insurance claims reserving in R, Proc. of the R User Conference, August 12–14, Dortmund, Germany, 2008.
9. Vapnik, V., *Statistical Learning Theory*, Wiley, New York, 1998.
10. Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.
11. Schölkopf, B. and Smola, A. J., *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
12. Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A., Kernlab—an S4 package for kernel methods in R, *J. Stat. Software*, 11(9):1–20, 2004.
13. Smola, A. J. and Schölkopf, B., A tutorial on support vector regression, *Stat. Comput.*, 14:199–222, 2004.
14. Schaback, R. and Wendland, H., Kernel techniques: From machine learning to meshless methods, *Acta Numerica*, 15:543–639, 2006.
15. Vapnik, V., Golowich, S., and Smola, A., Support vector method for function approximation, regression estimation, and signal processing, In *Advances in Neural Information Processing Systems 9*, pp. 281–287, MIT Press, Cambridge, MA, 1997.
16. Zhang, L., Zhou, W., and Jiao, L., Wavelet support vector machine, *IEEE Trans. Syst. Man Cybern.*, 34(1):34–39, 2004.
17. Stefan, R., SVM kernels for time series analysis, *Proc. of Tagungsband der GI Workshop-Woche*, pp. 43–50, 2001.