J. Mod. Power Syst. Clean Energy (2018) 6(2):268–280
https://doi.org/10.1007/s40565-018-0397-1

General noise support vector regression with non-constant uncertainty intervals for solar radiation prediction

J. PRADA[1], J. R. DORRONSORO[1]

Abstract General noise cost functions have been recently proposed for support vector regression (SVR). When applied to tasks whose underlying noise distribution is similar to the one assumed for the cost function, these models should perform better than classical ϵ-SVR. On the other hand, uncertainty estimates for SVR have received a somewhat limited attention in the literature until now and still have unaddressed problems. Keeping this in mind, three main goals are addressed here. First, we propose a framework that uses a combination of general noise SVR models with naive online R minimization algorithm (NORMA) as optimization method, and then gives non-constant error intervals dependent upon input data aided by the use of clustering techniques. We give theoretical details required to implement this framework for Laplace, Gaussian, Beta, Weibull and Marshall–Olkin generalized exponential distributions. Second, we test the proposed framework in two real-world regression problems using data of two public competitions about solar energy. Results show the validity of our models and an improvement over classical ϵ-SVR. Finally, in accordance with the principle of reproducible research, we make sure that data and model implementations used for the experiments are easily and publicly accessible.

CrossCheck date: 19 January 2018

Received: 30 June 2017 / Accepted: 19 January 2018 / Published online: 7 March 2018

✉ J. PRADA
jesus.prada@estudiante.uam.es

J. R. DORRONSORO
jose.dorronsoro@uam.es

[1] Universidad Autónoma de Madrid, Madrid, Spain

Keywords Support vector regression, General noise model, Naive online R minimization algorithm (NORMA), Uncertainty intervals, Clustering, Solar energy, Reproducible research

1 Introduction

Support vector machines (SVMs) and their regression counterpart, support vector regression (SVR) models have proved to perform well in many real-world situations, such as solar radiation [1], time series [2] or healthcare [3].

The classical version of this branch of regression models is called ϵ-SVR. This name comes from the cost function used in their optimization problem, the ϵ-insensitive loss function (ILF). The ILF ignores the errors within a certain distance ϵ to the target value, giving these points a value of zero cost and linear cost to errors outside the interval $(-\epsilon, \epsilon)$. According to [4], use of the ILF is justified under the assumption that noise in the data is additive and Gaussian, with its variance and mean being random variables. However, it has been proved that in several real-world problems noise distribution does not belong to the Gaussian family, with noise following instead significantly different distributions. This is the case, for example, for wind data [5]. For this reason, this paper presents a framework to build general noise SVR models, suited for any noise distribution assumption.

These proposed general noise SVR models use a different cost function and hence a different formulation of the optimization problem from the one applied in classical ϵ-SVR. A consequence of the use of the new formulation is that the standard optimization method employed in ϵ-SVR, sequential minimal optimization (SMO), becomes unfeasible to apply in these new models. The optimization

method chosen for the purpose of training the proposed models is naive online R minimization algorithm (NORMA); several reasons are behind the choice of NORMA above other SVM optimization methods for the purpose of this paper, such as simplicity, generalization capacity, and easy extension to regression and non-linear kernels. These factors will be discussed more in-depth later in this paper.

Furthermore, classical ϵ-SVR models do not provide any error interval estimates for their predictions. Therefore, a method to compute these intervals is also provided. The computation of these intervals is based on the work we previously carried out in [6], but the problem of constant intervals is solved in a general way in this paper.

In our previous work, we relied on problem-dependent techniques, based on expertise on the specific area that comprises the task we wanted to tackle, to cluster data into different groups and then we applied the proposed technique on each group. Here, we propose a general method to address this problem, based on the use of standard clustering methods, such as *k*-means or *k*-prototypes. This addition is a highly relevant one as intervals with the same width for each test instance could suppose a critical drawback for data whose distribution strongly depends on the input features, and the need of expertise to develop clustering methods to solve this problem entails a strong limitation to the application of these methods to general regression tasks.

This uncertainty interval computation method is used in combination with our general noise SVRs to provide a framework which can give predictions adapted to any noise distribution assumption and, at the same time, supply uncertainty estimates for these forecasts that also depend on the distribution assumed to be present in the noise. If we are able to make an accurate noise distribution assumption for a particular regression problem, the proposed framework should give optimized predictions and error intervals and, in particular, surpass classical ϵ-SVR accuracy.

Theoretical details and code implementations for this framework, that we call general noise SVRs with non-constant uncertainty intervals, are developed and made publicly available. We focus on a particular set of noise distributions, namely Laplace, Gaussian, Beta, Weibull, and the Marshall–Olkin generalized exponential (MOGE), but the framework is prepared to be easily applicable to other distributions.

To test the usefulness of this new framework, experiments using datasets from the American Meteorological Society (AMS), solar radiation prediction contest [7] and from the 2014 Global Energy Forecasting Competition (GEFCom2014) [8] are carried out. The goal of these contests is to achieve the best short term predictions and the best probabilistic distribution, respectively. Our results show that the proposed models can outperform classical ϵ-SVR models over two real-world regression tasks, and also that problem-independent techniques applied to tackle the problem of constant uncertainty intervals contribute significantly to improve the intervals accuracy with respect to constant width estimates and are competitive with clustering techniques based on expert analysis. Furthermore, Weibull and, specially, Beta distributions seem to be the best noise distribution assumptions for these solar energy problems, in accordance with previous results such as [9].

The main contributions of this paper can be summarized as follows:

1) The problem of constant width in the uncertainty intervals formulations described in [6] is addressed using general techniques, based on standard clustering methods such as *k*-means and *k*-prototypes.
2) A framework for general noise SVRs with non-constant uncertainty intervals is proposed, combining the use of general cost functions, NORMA optimization, clustering techniques and non-fixed error interval estimates for a particular choice of noise distribution assumption. All theoretical background, formulations and implementation for this framework are given for several probability distributions, with just some easy computations required to adapt our method to other choices of distribution assumption.
3) The proposed framework implementation is easily accessible as libraries for the R programming language via the comprehensive R archive network (CRAN). Availability of data sets used in the experiments is also guaranteed as they come from public competitions.
4) Experiments are carried out to tackle two real-world regression problems related to solar energy prediction. The following conclusions can be drawn from these experiments. First, the proposed models give better forecasts than classical SVR when a suitable noise distribution is assumed. Proposed techniques to avoid constant width in the uncertainty intervals described in [6] improve the accuracy of error estimates. Finally, Weibull and Beta distributions seem to capture best the underlying noise distribution in solar radiation prediction tasks.

The rest of this paper is organized as follows. A briefly review of prior theoretical background for classical ϵ-SVR formulation, general noise SVR models, NORMA optimization, uncertainty intervals for SVR and clustering methods such as *k*-means and *k*-prototypes is presented in Section 2. Section 3 gives an in-depth description of the proposed general noise SVRs with non-constant uncertainty intervals framework. Section 4 contains an explanation of implementation details and experiments over two

real-world solar datasets are described in Section 5. Section 6 analyzes the results obtained in these experiments. The paper ends with a short section on conclusions and possible lines of further work.

2 Prior theoretical background

2.1 Classical ϵ-SVR

For SVR, the loss function to be minimized is called the ϵ-ILF:

$$l_\epsilon(\delta_i) = \begin{cases} -\delta_i - \epsilon & \delta_i < -\epsilon \\ 0 & \delta_i \in [-\epsilon, \epsilon] \\ \delta_i - \epsilon & \delta_i > \epsilon \end{cases} \tag{1}$$

where $\delta_i = f(\boldsymbol{x}_i) - y_i$, $\boldsymbol{x}_i (i = 1, 2, \ldots, N)$ is the feature vector and $y_i (i = 1, 2, \ldots, N)$ is the target value we want to predict.

Adding the ridge regression regularization term we obtain the following optimization problem:

$$\min_{\boldsymbol{\beta}, \beta_0} H(\boldsymbol{\beta}, \beta_0) = \sum_{i=1}^{N} l_\epsilon(y_i - f(\boldsymbol{x}_i)) + \frac{\lambda}{2} ||\boldsymbol{\beta}||^2 \tag{2}$$

where $f(\boldsymbol{x}_i) = \boldsymbol{\beta} \boldsymbol{x}_i^{\mathrm{T}} + \beta_0$; $\lambda \geq 0$ is the regularization parameter; $\boldsymbol{\beta}$ and β_0 are the model weights and the bias term.

Reference [10] shows that this problem is equivalent to the following convex constrained optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{\beta}, \beta_0, \xi_i, \hat{\xi}_i} \left[\frac{1}{2} ||\boldsymbol{\beta}||^2 + C \sum\limits_{i=1}^{N} (\xi_i + \hat{\xi}_i) \right] \\ \text{s.t. } \xi_i, \hat{\xi}_i \geq 0 \\ \quad f(\boldsymbol{x}_i) - y_i \leq \epsilon + \xi_i \\ \quad y_i - f(\boldsymbol{x}_i) \leq \epsilon + \hat{\xi}_i \end{cases} \tag{3}$$

where $i = 1, 2, \ldots, N$; ξ_i and $\hat{\xi}_i$ are quantify errors above and below the ϵ-band, respectively; C is the cost hyper-parameter used to regulate model complexity and has an analogous purpose to λ in (2).

In practice, the problem solved is the dual formulation derived using standard Lagrangian techniques [11]:

$$\begin{cases} \max\limits_{\alpha_i, \alpha_i^*} L_D(\alpha_i, \alpha_i^*) = \sum\limits_{i=1}^{N} y_i(\alpha_i^* - \alpha_i) - \\ \qquad \frac{1}{2} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j - \\ \qquad \epsilon \sum\limits_{i=1}^{N} (\alpha_i^* + \alpha_i) \\ \text{s.t. } 0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1, 2, \ldots, N \\ \quad \sum\limits_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0 \end{cases} \tag{4}$$

Solutions satisfy the Karush–Kuhn–Tucker conditions:

$$\begin{cases} \alpha_i (y_i - f(\boldsymbol{x}_i) + \epsilon + \xi_i) = 0 \\ \alpha_i^* (f(\boldsymbol{x}_i) - y_i + \epsilon + \hat{\xi}_i) = 0 \\ (C - \alpha_i) \xi_i = 0 \\ (C - \alpha_i^*) \hat{\xi}_i = 0 \end{cases} \tag{5}$$

It can be shown [10] that:

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) \boldsymbol{x}_i \tag{6}$$

Therefore, solution functions to the SVR problem have the following form:

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}_i + \hat{\beta}_0 \tag{7}$$

Finally, using the kernel trick and a kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ satisfying Mercer's condition [12], we can get the following analogous formulations of (4) and (7).

$$\begin{aligned} L_D = & \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i) - \\ & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \\ & \epsilon \sum_{i=1}^{N} (\alpha_i^* + \alpha_i) \end{aligned} \tag{8}$$

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) k(\boldsymbol{x}, \boldsymbol{x}_i) + \hat{\beta}_0 \tag{9}$$

That allow us to extend the previous linear version of the SVR problem to a non-linear one with no need of explicitly computing a set of basis functions $\{h_m(\boldsymbol{x}), m = 1, 2, \ldots, M\}$.

2.2 General noise SVR

In 2002, an SVR formulation to obtain a general noise version of the model was proposed in [13]. This general noise SVR can be used with any particular loss function $c(\boldsymbol{x}_i, y_i, f(\boldsymbol{x}))$. Its optimization problem is described as:

$$\begin{cases} \min\limits_{\boldsymbol{\beta},\beta_0,\xi_i,\hat{\xi}_i} \left[\frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum\limits_{i=1}^{N}(c_i(\xi_i) + c_i(\hat{\xi}_i))\right] \\ \text{s.t. } \xi_i, \hat{\xi}_i \geq 0 \\ \quad f(\boldsymbol{x}_i) - y_i \leq \epsilon_i + \xi_i \\ \quad y_i - f(\boldsymbol{x}_i) \leq \epsilon_i^* + \hat{\xi}_i \end{cases} \tag{10}$$

where $i = 1, 2, \ldots, N$; $c_i(\xi_i) = c(\boldsymbol{x}_i, y_i, y_i + \epsilon_i + \xi)$; $c_i(\hat{\xi}_i) = c(\boldsymbol{x}_i, y_i, y_i - \epsilon_i^* - \hat{\xi})$; ϵ_i and ϵ_i^* are chosen such that $c(\boldsymbol{x}_i, y_i, y_i + \xi) = 0\ ,\ \forall\ \xi \in [-\epsilon_i^*, \epsilon_i]$.

2.3 NORMA optimization

In [14], an optimization method suitable for SVRs in an online setting was proposed. This method focuses on the so-called instantaneous regularized risk:

$$R_{inst,\lambda}[f_t, \boldsymbol{x}_t, \boldsymbol{y}_t] := l(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t) + \frac{\lambda}{2}||f_t||^2_{\mathcal{H}} \tag{11}$$

where l is a given loss function; f_t is a function where $f_t \in \mathcal{H}$ with $\mathcal{H}$ a reproducing kernel Hilbert space and k the corresponding kernel; $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are the feature vectors and targets available at instant t, respectively.

NORMA performs gradient descent with respect to $R_{inst,\lambda}$, i.e., it uses as update rule:

$$f_{t+1} = f_t - \eta_t \partial_f R_{inst,\lambda}[f_t, \boldsymbol{x}_t, \boldsymbol{y}_t] \tag{12}$$

where $\eta_t > 0$ is the learning rate, which usually is constant, i.e., $\eta_t = \eta$.

We can split the derivative $\partial_f R_{inst,\lambda}[f_t, \boldsymbol{x}_t, \boldsymbol{y}_t]$ into two factors: $\partial_f l(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t)$ and $\partial_f \lambda||f_t||^2_{\mathcal{H}}/2$. As stated in [14], the following equations for these factors hold true.

$$\partial_f l(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t) = l'(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t)k(\boldsymbol{x}_t, \cdot) \tag{13}$$

$$\partial_f \frac{\lambda}{2}||f_t||^2_{\mathcal{H}} = \lambda f_t \tag{14}$$

where $l'(\boldsymbol{x}, \boldsymbol{y}) = \partial_{\boldsymbol{x}} l(\boldsymbol{x}, \boldsymbol{y})$.

Substituting (13) and (14) into (12) we get:

$$\begin{aligned} f_{t+1} &= f_t - \eta_t(l'(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t)k(\boldsymbol{x}_t, \cdot) + \lambda f_t) \\ &= (1 - \eta_t\lambda)f_t - \eta_t l'(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t)k(\boldsymbol{x}_t, \cdot) \end{aligned} \tag{15}$$

In (14), it is needed for the algorithm to work properly that $\eta_t < 1/\lambda$ holds.

Reformulating f_t in (15) in the form of kernel expansions as described in [14] we get:

$$f_{t+1}(\boldsymbol{x}) = \sum_{i=1}^{t} \hat{\alpha}_i k(\boldsymbol{x}_i, \boldsymbol{x}) \tag{16}$$

$$\hat{\alpha}_i := (1 - \eta_t\lambda)\alpha_i \quad i < t \tag{17}$$

$$\hat{\alpha}_i := -\eta_t l'(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t) \quad i = t \tag{18}$$

where usually $f_1 = 0$.

In practice, the set of update rules used to apply NORMA optimization is precisely (16), (17) and (18). As shown in [14], if it is necessary to take into account the possibility of existence of an offset b for the function f, this can be added through an extra update rule:

$$b_{t+1} = b_t - \eta_t l'(f_t(\boldsymbol{x}_t), \boldsymbol{y}_t) \tag{19}$$

2.4 Uncertainty intervals for SVR

A method to compute error intervals for SVR predictions is proposed in [15]. The idea is to estimate the real distribution of prediction errors Ψ, performing maximum likelihood estimation (MLE) over a set of out-of-sample residuals $\{\psi_i, i = 1, 2, \ldots, l\}$ obtained by applying k-fold cross-validation over the training data.

In this work, the authors assume that conditional distribution of y given x depends on x only through $\hat{f}(x)$. In theory, the distribution of prediction errors Ψ may depend on input x and therefore the length of the uncertainty interval with a pre-specified coverage probability may vary from one example to another. The authors in [15] admit this could be a critical drawback for some particular regression tasks but the problem remains unaddressed.

Only two noise distributions assumptions are considered in [15], zero-mean Laplace and Gaussian, which are fitted using MLE. Specifically, we can estimate the distributions parameters by maximizing the logarithm of the likelihood l of sample residuals. Assuming that $\{\psi_i, i = 1, 2, ..., N\}$ is independent, this is equivalent to:

$$\max_{\theta} \sum_{i=1}^{N} \ln g(\psi_i|\theta) \tag{20}$$

where g denotes the density function of prediction errors; θ is the distribution parameter to estimate.

Finally, given a pre-specified probability $1 - 2s$ with $s \in (0, 0.5)$, the goal is to obtain the corresponding error interval (a, b), which in this method is constant for each point. For a zero-mean symmetric variable this is obtained setting $a = -p_s$ and $b = p_s$, where p_s is the upper s^{th} percentile.

2.5 Clustering methods

In [16], the k-means algorithm is proposed. Its aim is to find k clusters that minimize the within-cluster sum of

squares, or squared Euclidean distance shown in (21) with $S = \{s_1, s_2, \ldots, s_k\}$ clusters and their centroids $\boldsymbol{D}_i$.

$$\min_{S} \sum_{i=1}^{k} \sum_{x \in s_i} ||x - \boldsymbol{D}_i||^2 \tag{21}$$

The solution found by k-means depends on how the centroids are initialized. Forgy method, i.e., to randomly choose k points from the dataset and use them as initial centroids, is recommended for standard k-means.

Given an initial set of cluster centroids $\{\boldsymbol{D}_i^0, i = 1, 2, \ldots, k\}$, the k-means algorithm proceeds by iterating two steps:

1) Assignment step: assign each observation x_i to the cluster s_w with the minimum euclidean distance between its centroid $\boldsymbol{D}_w$ and the observation.
2) Update step: compute the mean of all points in each cluster and set it to be the new cluster centroid.

These two steps are iterated until the convergence criteria $\boldsymbol{D}_i^t = \boldsymbol{D}_i^{t-1}$ is reached, where t is a particular iteration of the algorithm and $t - 1$ the previous one.

k-means algorithm can only be applied to numerical values. k-prototypes [17] is an algorithm which extends k-means algorithm to datasets of mixed numeric and categorical values by changing the squared Euclidean distance to:

$$d(\boldsymbol{w}, \boldsymbol{z}) = \sum_{j=1}^{p} (w_j - z_j)^2 + \gamma \sum_{j=p+1}^{m} G(w_j, z_j) \tag{22}$$

where $\boldsymbol{w}$ and $\boldsymbol{z}$ are two mixed vectors; $w_1, \ldots, w_p, z_1, \ldots, z_p$ are numerical variables; $w_{p+1}, \ldots, w_m, z_{p+1}, \ldots, z_m$ are categorical variables; γ is a weight factor to balance each type of attributes; G is defined by:

$$G(w_j, z_j) = \begin{cases} 0 & w_j = z_j \\ 1 & w_j \neq z_j \end{cases} \tag{23}$$

3 Proposed framework

3.1 General noise SVR models via NORMA optimization

The use of the ϵ-insensitive loss function in the classical SVR formulation explained in Section 2.1 implies the assumption of a particular error distribution, related to the Gaussian family, in the data [4]. However, it has been observed that the noise in some real-world applications may satisfy other distributions. For example, it has been proved that for wind power forecast it is preferable to assume a Beta distribution [5]. We think that examining whether some distributions other than the Gaussian better fit also the problem of solar energy prediction may be worthwhile.

Taking this into account, we look to build a general noise formulation for SVR where a particular distribution p for the noise is assumed, the optimal loss function for that distribution is computed, and then this function is plugged into the model to obtain a SVR formulation for that distribution assumption.

As described in Section 2.2, a general noise formulation for SVR has been described in the past, providing an expression of the dual problem that allows to insert different loss functions into it. The difficulty with this formulation is that it aims to solve the dual problem, which for some choices of noise distributions results in a very complex optimization problem, one that cannot be tackled using standard optimization techniques such as SMO [18]. Therefore, we need to find a different optimization method for our proposed model.

On the other hand, NORMA optimization can be used in a straightforward manner not only in classification problems, but also in novelty detection and regression tasks, the latter being the focus of this paper. Furthermore, its extension from linear models to non-linear ones is also largely direct via the use of the kernel trick. Finally, its formulation and implementation is fairly simple and its generalization to any loss function does not suppose great difficulties. Our goal is to find a rather simple formulation of the model that is the most general possible, one that allows to insert the optimal loss function corresponding to any choice of noise distribution without this decision increasing significantly the difficulty of the optimization problem to solve.

NORMA is perfectly suited for this task, avoiding the extra complexity derived of inserting general noise functions to the dual problem in (10). For all these reasons, NORMA is the optimization method used in our research.

We study now the optimization problem resulting of using NORMA with the distributions considered for this work. These distributions have been chosen for either being standard alternatives, as Laplace and Gaussian distributions [15], being related to radiation forecasting, as is the case for the Beta and Weibull distributions [19], or being relevant to other particular kind of regression tasks such as healthcare problems, as the Marshall–Olkin distribution [20]. First, we have to compute their optimal loss functions. Following [19], the optimal loss function in a maximum likelihood sense for a particular choice of error distribution $P(\psi_i)$ can be formulated as:

$$l(\psi_i) = -\ln P(\psi_i) \tag{24}$$

Therefore, using (24) and removing all factors constant with respect to ψ_i, we can obtain the optimal loss functions

associated to a given choice of noise distribution. Optimal loss functions for the distributions considered in this paper and their derivatives can be expressed as:

1) Laplace

$$l(\psi_i) = \frac{|\psi_i - \mu|}{\sigma} \tag{25}$$

$$l'(\psi_i) = \begin{cases} \frac{1}{\sigma} & \psi_i - \mu > 0 \\ 0 & \psi_i - \mu = 0 \\ -\frac{1}{\sigma} & \psi_i - \mu < 0 \end{cases} \tag{26}$$

where μ and σ are the mode and standard deviation of ψ_i, respectively.

2) Gaussian

$$l(\psi_i) = \frac{(\psi_i - \mu)^2}{2\sigma^2} \tag{27}$$

$$l'(\psi_i) = \frac{\psi_i - \mu}{\sigma^2} \tag{28}$$

where σ^2 is the variance of ψ_i.

3) Beta

$$l(\psi_i) = (1 - A)\ln\psi_i + (1 - B)\ln(1 - \psi_i) \tag{29}$$

$$l'(\psi_i) = \frac{1 - A}{\psi_i} - \frac{1 - B}{1 - \psi_i} \tag{30}$$

where A and B are the shape parameters of the Beta distribution.

4) Weibull

$$l'(\psi_i) = \begin{cases} (1 - \kappa)\ln\psi_i + \left(\frac{\psi_i}{L}\right)^{\kappa} & \psi_i > 0 \\ 0 & \psi_i \le 0 \end{cases} \tag{31}$$

$$l'(\psi_i) = \begin{cases} \frac{1 - \kappa}{\psi_i} + \frac{\kappa}{L}\left(\frac{\psi_i}{L}\right)^{\kappa - 1} & \psi_i > 0 \\ 0 & \psi_i \le 0 \end{cases} \tag{32}$$

where L and κ are the scale and shape of the Weibull distribution, respectively.

5) MOGE

$$l(\psi_i) = \begin{cases} 2\ln(T + (1 - T)(1 - \mathrm{e}^{-L_2\psi_i)^{A_2}}) + \\ \quad L_2\psi_i + (1 - A_2)\ln(1 - \mathrm{e}^{-L_2\psi_i}) & \psi_i > 0 \\ 0 & \psi_i \le 0 \end{cases} \tag{33}$$

$$l'(\psi_i) = \begin{cases} L_2\left\{1 + \mathrm{e}^{-L_2\psi_i}\left[2A_2(1 - T)(1 - \mathrm{e}^{-L_2\psi_i})^{A_2 - 1}\right.\right. \\ \quad \left.\left. + \frac{1 - A_2}{1 - \mathrm{e}^{-L_2\psi_i}}\right]\right\} & \psi_i > 0 \\ 0 & \psi_i \le 0 \end{cases} \tag{34}$$

where A_2, L_2, T are the parameters of the MOGE distribution.

Full computations to obtain these optimal loss functions are given in our previous work [21]. Note that, technically, at $\psi_i = 0$ the derivatives corresponding to Laplace distributions are non-differentiable. However, this case corresponds to predictions with no error, so we take as a proxy for the derivative at this point the value $l' = 0$. Plugging the derivatives $l'(\psi_i)$ into the NORMA update rules as shown in (18) and (19), we get a NORMA formulation adapted to a particular choice of distribution. For instance, for the Gaussian distribution we get:

$$\begin{cases} \hat{\alpha}_t := -\eta_t \frac{\psi_i - \mu}{\sigma^2} \\ b_{t+1} = b_t - \eta_t \frac{\psi_i - \mu}{\sigma^2} \end{cases} \tag{35}$$

Equations (16) and (17) do not depend directly on the choice of loss function so they remain the same regardless of the noise distribution assumption.

NORMA is based on stochastic gradient descent. Asymptotic convergence to a stationary point for these methods is proved in [22] in the non-convex case, but this point is not guaranteed to be a global minima as opposed to the convex situation. Therefore, this problem must be addressed and in Section 5.4 we describe how we have deal with it.

The extension of the approach presented here to other choice of distribution assumption is straightforward, with only simple computations of MLE to get the optimal loss functions and calculation of the derivatives of these functions required.

As far as we know, at the time of writing this paper there has not been described a methodology to give explicit, feasible to solve and easy to extend formulations of general noise SVR models that allow to use noise distribution assumptions such as the Weibull distribution. In particular, we have not found in the literature any approach that tried to use NORMA to solve the optimization problem of general noise SVR. We think this is one of the main contributions of our work and one that may prove to be useful in this line of research.

3.2 Nonconstant uncertainty intervals using clustering methods

For the computation of uncertainty intervals for our model predictions we propose to follow an approach, based in the method described in Section 2, consisting of three stages: clustering, parameter estimation via maximum likelihood, and probability interval computation.

3.2.1 Clustering

As stated before, in [15] the conditional distribution of y given x is assumed to depend on x only through the prediction value $\hat{f}(x)$ and therefore the width of the uncertainty intervals is the same for each instance in the test set. To solve this drawback we propose the following method:

1) Use clustering methods to split train, and validation data if used, into several groups s_i, $i = 1, 2, \ldots, k$. k-means or k-prototypes are suggested as clustering algorithms. Forgy method is preferable as initialization method for standard k-means.
2) Fit a model M_i for each cluster s_i.
3) For each s_i use cross-validation or validation errors of model M_i to build uncertainty intervals following steps described below in Sections 3.2.2 and 3.2.3.
4) For each test instance $x_{i,test}$, assign it to the cluster s_i with the nearest centroid using a given distance metric and apply to $x_{i,test}$ the error interval corresponding to this selected cluster. We suggest the Euclidean distance for k-means and (22) for k-prototypes as distance functions.

To choose the value of k we propose to use a grid search over a region of possible values and pick the k that results in the most accurate uncertainty intervals with respect to a given metric. Our choice of accuracy metric for error intervals p_{err} is described in Section 5.1.

3.2.2 Parameter estimation

In [21] we gave the computational steps required for parameter estimation via MLE for all the distributions considered in this work. Newton–Raphson method is used for the Beta, Weibull, and MOGE computations.

3.2.3 Probability intervals

Given a pre-specified probability $1 - 2s$, we can obtain the prediction error interval (a, b) as follows:

1) Laplace and Gaussian: the percentile p_s is computed by solving:

$$1 - s = \int_{-\infty}^{p_s} p(z)\mathrm{d}z \tag{36}$$

As the distribution is centered at μ and not necessarily at zero, the prediction error interval is $(\mu - (p_s - \mu), \mu + (p_s - \mu))$.

2) Beta, Weibull, and MOGE: for Beta distribution it holds that $z \geq 0$, so p_s is obtained by solving:

$$1 - s = \int_{0}^{p_s} p(z)\mathrm{d}z \tag{37}$$

The prediction error interval is then $(0, p_s)$. For Weibull and MOGE distributions only the case $z \geq 0$ is relevant, so we determine the error interval the same way as for the Beta distribution.

4 Implementation

We used the R programming language for implementation of the proposed framework. In particular, we developed two R libraries:

1) NORMA: used to build general noise SVR models by applying NORMA optimization.
2) errint: employed to compute and analyze error intervals for a particular model predictions assuming different distributions for noise in the data.

Four other already implemented R libraries and one python library have also been used to carry out our experiments:

1) e1071: R version of the popular library LIBSVM [23]. We used it to build standard ϵ-SVR models.
2) stats: included in the basic packages for R. Contains functions for statistical calculations and random number generation. Employed to apply k-means.
3) clustMixType: functions to perform k-prototypes partitioning for mixed variable-type data according to [17].
4) ncdf4: provides a high-level R interface to files written using Unidata's network common data form version 4 (netCDF4), as is the case for the files used in the AMS contest and described in Section 5.3.
5) pvlib-python: provides a set of functions and classes for simulating the performance of photovoltaic energy systems [24]. Used to compute clear sky curves.

All these libraries can be publicly downloaded via CRAN, for the R libraries, or GitHub, for the case of the pvlib-python library.

5 Experiments

5.1 Metrics

The metric used to evaluate the quality of model predictions in the AMS competition is the pure mean absolute error (MAE):

$$MAE = \frac{\sum_{i=1}^{N} |\hat{f}(x_i) - y_i|}{N} \tag{38}$$

However, based on our experience in solar and wind energy tasks we consider the relative mean absolute error (RMAE) to be a better choice to evaluate performance of a model in this particular task. This metric is defined as follows:

$$RMAE = 100 \frac{\sum_{i=1}^{N} \frac{|\hat{f}(x_i) - y_i|}{|y_i|}}{N} \tag{39}$$

Regarding evaluation of the uncertainty interval accuracy, given a pre-specified probability $1 - 2s$ we compare the percentage of test prediction errors $\psi_{i,test}$ lying inside the corresponding uncertainty intervals $[a, b]$, with the expected number $1 - 2s$:

$$p_{err} = \left| \frac{|\{\psi_{i,test} : \psi_{i,test} \in [a, b]\}|}{N} - (1 - 2s) \right| \tag{40}$$

We choose here an absolute error as the accuracy measure over one with different weights for positive or negative deviations because our preference towards a positive or negative error, i.e. which one is considered more or less detrimental of the two, is extremely problem-dependent. In some tasks it is preferable to take a more conservative approach, penalizing more negative errors, but in others a more risky approach could be a better option, tending to punish positive errors more. Here we opt to use the most neutral possible option as our measure.

In the GEFCom2014 competition the goal is to find the best quantile predictions for solar power generation. Therefore, an evaluation metric suited to this purpose must be used. They opt to use the pinball loss function to evaluate the accuracy of these probabilistic forecasts. This metric is defined as follows:

$$PL_{\tau}(y, z) = \begin{cases} (y - z)\tau & y \geq z \\ (z - y)(1 - \tau) & y < z \end{cases} \tag{41}$$

where τ is the target quantile; z is the predicted quantile value; y is the exact numerical value of solar power.

5.2 Model parameters selection

We use the Gaussian kernel for all models considered, as it has been shown in the past and based on our own experience to be the best choice for SVR models for most regression tasks. The formulation of the Gaussian kernel is:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\gamma \parallel \boldsymbol{x} - \boldsymbol{x}' \parallel^2\right) \tag{42}$$

Before performing our tests we must select the best hyperparameters for each model. We select $\{C, \epsilon, \gamma\}$ for classical ϵ-insensitive SVR by a standard grid search over a fixed validation set.

For general noise SVR models using loss functions other than ILF, the density parameters are selected applying the MLE formulas shown in [21], which in some cases involve solving numerically the equations over a set of residuals. We use for this purpose the validation residuals of the previously computed optimum ϵ-insensitive SVR. Afterwards, these same equations are solved to obtain the density parameters used to build the corresponding error intervals for each model, but in this case the validation residuals of the corresponding optimum general noise SVR calculated previously are used. Finally, the kernel width γ is obtained for each general noise SVR model the same way as for the ϵ-insensitive SVR case, i.e by means of a grid search over the validation set.

5.3 Datasets

The first dataset analyzed corresponds to the Kaggle AMS 2013–2014 solar radiation prediction contest. The goal of this contest is to discover which statistical and machine learning models provide the best predictions of daily-aggregated solar radiation. In particular, models must predict the total daily incoming solar radiation at 98 Oklahoma mesonet sites, which will serve as 'solar farms' for the contest. Real values of total daily incoming solar radiation (J/m^2) at these 98 points are provided in the AMS dataset. Location coordinates and elevation for each station are also given.

Input numerical weather prediction data for the contest comes from the global ensemble forecast system (GEFS) reforecast version 2. The data are in netCDF4 files; each one contains the total data for one of the model variables and is stored in a multidimensional array. The first dimension is the date of the model run. The second dimension is the ensemble member that the forecast comes from. The GEFS has 11 ensemble members with perturbed initial conditions but we use only ensemble 1 in our experiments for simplicity. The third dimension is the forecast hour, which runs from 12 to 24 hours in 3 hours increment. All models run start at 00 coordinated universal

time (UTC), so they will always correspond to the same universal time although local solar time will vary over each year. The fourth and fifth dimensions are the latitude and longitude uniform spatial grid. The longitudes in the file are in positive degrees from the prime meridian, so subtracting 360 from them will translate them to a similar range of values as the ones given for the stations. The list of variables is given in [25]. Elevation of each GEFS point is provided in a separate netCDF4 file.

Data of the contest covers years from 1994 to 2007. For the purpose of our experiments, we split this dataset into train (1994–2005), validation (2006) and test (2007). The complete dataset is freely available at [25].

The second dataset employed in the experiments is the one used in the GEFCom2014 contest, where the probabilistic solar power forecasting track aims to estimate the probabilistic distribution, in quantiles, of solar power generation for three adjacent solar farms on a rolling basis. The target variable is solar power and there are 12 independent numerical weather prediction (NWP) variables from the European centre for medium-range weather forecasts (ECMWF). The complete list of these 12 variables is given in Table 1.

Data is given in comma separated values with each row corresponding to one hour of a particular day. The dataset includes 15 different tracks, but we will focus only in track 15 for the purpose of this paper. Data available goes from 2012-04-01 to 2014-07-01. We will split the data using the following approach: 1) train from 2012-06-01 to 2013-05-31; 2) validation from 2013-06-01 to 2014-05-31; 3) test from 2014-06-01 to 2014-07-01. The complete dataset is accessible via [8].

Table 1 GEFCom2014 dataset variables and their corresponding units

Variable	Description	Unit
VAR78	Total column liquid water	kg/m^2
VAR79	Total column ice water	kg/m^2
VAR134	Surface pressure	Pa
VAR157	Relative humidity at 1000 mbar	%
VAR164	Total cloud cover	0–1
VAR165	10 metre eastward wind	m/s
VAR166	10 metre northward wind	m/s
VAR167	2 metre temperature	K
VAR169	Surface solar rad down	J/m^2
VAR175	Surface thermal rad down	J/m^2
VAR178	Top net solar rad	J/m^2
VAR228	Total precipitation	m

5.4 Experiment I—general noise SVR versus classical ϵ-SVR

The purpose of this experiment is to test the performance of classical ϵ-SVR versus our proposed general noise SVR models for the AMS and GEFCom2014 datasets described in Section 5.3. In particular, we build general noise SVR models following the approach proposed in Section 1 using the Laplace, Gaussian, Beta, Weibull, and MOGE distributions as noise assumptions. Hyperparameters are optimized as described in Section 5.2. We discard night hours, where solar radiation is zero or close to zero, for evaluation.

As stated before, the use of non-convex loss functions could lead to local minima when applying NORMA optimization. We use two mechanisms to deal with this problem:

1) Constrain the parameters of the chosen distribution to be outside the set of parameters which cause the loss function to be non-convex, e.g. in the beta distribution it will mean to use the constraints $\alpha \geq 1,\ \beta \geq 1$.
2) A more general and less restrictive alternative to deal with this obstacle is to compute several times the optimization algorithm using different choices of initial points and keep the best solution to the optimization problem as our final function.

In this experiment we try both approaches and keep the model that gives the best results. Moreover, we have also tested the use of a theoretical clear sky solar irradiance model and add its estimates as a new feature to the winning model in the case of the AMS contest, where stations geolocation is available, to test if performance is improved. For this purpose, we follow the simplified Solis method proposed by Ineichen in [26] and implemented in pvlib-python.

5.5 Experiment II—uncertainty intervals for general noise SVR

In this experiment we test the accuracy of uncertainty intervals built following the method proposed in Section 3.2 under different assumptions of noise distribution and distinct choices of clustering methods. As noise distributions we try the same options as in experiment I. The list of clustering methods tested is the following one:

1) M_{unique}: build a unique interval for all instances in the test set.
2) M_k: cluster data using standard and general methods as described in Section 3.2. In particular, we use k-means here as all features are numerical.

3) M_{expert}: analogous to M_k but using this time techniques based on expertise to cluster data. Keeping in mind that the experiment corresponds to a solar radiation regression task and based on results showed in [27], we propose to split data into 3 groups: group 1 corresponds to low radiation hours; group 2 corresponds to medium radiation hours; group 3 corresponds to high radiation hours.

The experiment is carried out two times, the first one computing intervals that should contain 80% of the test predictions and the second time with 90% intervals, i.e. choosing $s = 0.1$ and $s = 0.05$, respectively. The mean of both results is then computed to obtain the final error. Besides, we also compute the required quantiles in order to compare our proposed method with the public leaderboards available for the GEFCom2014 competition at CrowdANALYTIX, where the pinball function is used for evaluation as described in Section 5.1. As in experiment I, once again the datasets used are the ones corresponding to the AMS and GEFCom2014 contests.

6 Results analysis

6.1 Experiment I

The global results for experiment I are shown in Table 2. Three conclusions can be drawn for them. First, the choice of noise distribution assumption is highly relevant for model accuracy, as the worst result is 79% and 74% higher than the lowest RMAE obtained for the AMS and GEFCom2014 datasets, respectively. Second, providing that the distribution assumption is properly chosen, general noise SVR models achieve significantly higher precision than classical ϵ-SVR. Finally, the Weibull and primarily Beta distributions seem to capture better the underlying noise distribution for the task of solar energy prediction. Although we would need further testing of our models with different datasets to confirm these results, they seem to be in line with previous works, such as [9] or [28], that suggest the Beta distribution as a good choice to model solar irradiation.

In Table 3 we can see that, for each choice of noise distribution assumption, the total number among the 98 Oklahoma mesonet sites for the AMS competition or among the 3 available solar farms for the GEFCom2014 contest where the corresponding model achieves the best performance. It is again clear that the Beta distribution is consistently performing better than the other distributions, with Weibull in second place. For the AMS dataset, we have analyzed sites where Beta is not the winning model and have not found any clear patterns in geolocation or other of the available input features that could allow us to distinguish between these stations and the rest.

Table 2 RMAE for AMS and GEFCom2014 competitions

Dataset	ϵ-SVR	Laplace	Gaussian	Beta	Weibull	MOGE
AMS	12.35	13.38	12.47	9.76	10.81	17.48
GEFCom	18.05	19.88	17.95	15.83	16.32	27.48

Table 3 Number of sites where a particular model performs best than the rest for AMS and GEFCom2014 competitions

Dataset	ϵ-SVR	Laplace	Gaussian	Beta	Weibull	MOGE
AMS	5	1	6	71	15	0
GEFCom	0	0	0	3	0	0

Moreover, although as we stated before, we consider RMAE as a more suited metric for the problem at hand, we also tested our models through the Kaggle site where standard MAE is used for evaluation. Results can be found in Table 4. Our model using Beta noise assumption gets a score of 2207121.72, good enough for eight place among all the 160 participants visible on Kaggle private leaderboard. This is a quite positive result, specially taking into account that the goal of this work is not to find the best possible model in terms of accuracy, as we follow a quite simple and straightforward pipeline to tackle the problem with very little data processing and almost nothing of feature engineering or expertise integration, and we also use a relative small grid for the parameter search. Our aim is instead to compare the performance of the different noise distributions among themselves and to compare the proposed models with classical ϵ-SVR.

Finally, the results of adding clear sky information as a new feature to the best model for the AMS contest, i.e. the Beta-noise SVR, are shown in Table 5. It appears to be clear that addition of this clear sky feature has a positive impact on the model, improving even more the score previously obtained for all evaluation metrics considered: RMAE, MAE and number of sites where the model performs better than any other.

Table 4 MAE given by Kaggle after submission in AMS contest with corresponding leaderboard ranking

Model	MAE (J/m^2)	Leaderboard ranking
ϵ-SVR	2328401.83	36
Laplace	2403362.81	48
Gaussian	2328018.55	36
Beta	2207121.72	8
Weibull	2259056.34	18
MOGE	2559516.97	109

Table 5 Impact of adding clear sky, CS, information in the performance of the best model for AMS competition

Model	RMAE	Sites	MAE (J/m^2)	Leaderboard ranking
Beta	9.76	71	2207121.72	8
Beta + CS	9.41	75	2188527.25	6

Table 6 Uncertainty intervals p_{err} by noise assumption and clustering technique for AMS and GEFCom2014 competitions (mean p_{err} value for $s = 0.1$ and $s = 0.05$ is shown)

Model	M_{unique}		M_k		M_{expert}	
	AMS	GEFCom	AMS	GEFCom	AMS	GEFCom
Laplace	1.78	2.94	1.21	1.68	1.17	1.64
Gaussian	1.85	2.94	1.32	1.64	1.15	1.66
Beta	1.86	2.72	0.82	1.34	0.68	1.30
Weibull	1.97	2.86	0.87	1.34	0.78	1.33
MOGE	2.12	3.37	1.57	2.02	1.31	1.98

6.2 Experiment II

Table 6 contains the results obtained in experiment II. The negative impact of computing error intervals with constant width in method M_{unique} is clear when looking at these table, as the best p_{err} obtained when using this approach is more than twice the ones accomplished when applying some sort of clustering techniques, as is the case for M_k and M_{expert}. Besides, when following this method, the noise distribution assumption that gives best results for the AMS dataset is the Laplace, whereas the Beta is the most accurate one for the other two approaches. This is a result more in line with previous research and the outcome of experiment I, a fact that seems to indicate a bad functioning of M_{unique} that makes this method unable to properly capture the underlying noise distribution of the task at hand.

General clustering techniques such as k-means seem to solve, at least in large part, this drawback, with M_k method obtaining results that are competitive with ad hoc expertise clustering approaches that are problem-dependent, as the one employed in M_{expert}. Moreover, we also tested our method against the available CrowdANALYTIX public leaderboard for the GEFCom2014 contest. The results are detailed in Table 7, with our best model achieving fifth position.

Table 7 PL given by CrowdANALYTIX after submission in GEFCom2014 contest with corresponding leaderboard ranking

Model	PL	Leaderboard ranking
ϵ-SVR	0.01412	10
Laplace	0.01467	14
Gaussian	0.01403	10
Beta	0.01298	5
Weibull	0.01342	7
MOGE	0.01821	17

7 Conclusion

In this paper we have proposed a framework to build general noise SVRs with non-constant uncertainty intervals that involves two main phases. On one hand, a method to build general noise SVR models using NORMA as the optimization algorithm. On the other hand, an approach to compute error intervals for these regression models avoiding constant width by the use of standard clustering methods, instead of employing ad hoc partitioning approaches as proposed in our previous work [6].

Both techniques rely on a concrete choice of noise distribution assumption and in this work we have given the mathematical framework needed for their implementation under several distributions, namely Laplace, Gaussian, Beta, Weibull, and MOGE. It is important to remark that just some easy computations are needed to extend the method to other distribution choices. The algorithms necessary to apply these two techniques have been implemented using R as programming language and made publicly available via CRAN. Moreover, the datasets employed in the experiments correspond to public competitions and therefore are freely accessible. Therefore, we have carried out our work in accordance with the principles of reproducible research, which was one of our main goals.

Finally, experiments have been made to test our proposed framework in real-world tasks related to the problems of solar radiation and energy prediction. These tests show that the suggested general noise SVR models can achieve more accurate predictions than classical ϵ-SVR models if the noise distribution assumption is properly chosen. Furthermore, the proposed clustering methods seem to largely solve the critical drawback of constant width in the uncertainty estimates that could arise in our framework, and are shown to be competitive with problem-dependent clustering based on expertise such as the ones employed in our previous works. Lastly, the distributions that seem to capture best the underlying noise distribution in these solar tasks are the Weibull and, even more, Beta distributions.

Regarding possible lines of further work, one of them could be to add more distributions to the ones studied in this paper, such as Cauchy or Logistic, and then test the performance of our proposed framework for problems where these distributions may be of relevance.

Another reasonable extension of the research carried out here will be to compare the accuracy of the uncertainty intervals built following the approach suggested here versus error intervals computed using ensemble weather prediction as the one from GEFS, which provides 11 separate forecasts, or ensemble members, and therefore allows to build 11 different predictions and compute error intervals by counting how many of these predictions fall within a specific range.

Lastly, checking the use of general noise loss functions like the ones considered in this research in other regression methods where models are built by minimizing concrete loss functions, such as deep learning or model stacking frameworks, could also be an interesting idea worthy of further investigation.

Acknowledgements With partial support from Spain's grants TIN2013-42351-P, TIN2016-76406-P, TIN2015-70308-REDT, as well as S2013/ICE-2845 CASI-CAM-CM. This work was supported also by project FACIL–Ayudas Fundación BBVA a Equipos de Investigación Científica 2016 and the UAM–ADIC Chair for Data Science and Machine Learning. We gratefully acknowledge the use of the facilities of Centro de Computación Científica, CCC, at Universidad Autonoma de Madrid, UAM.

References

[1] Gala Y, Fernandez A, Diaz J et al (2013) Support vector forecasting of solar radiation values. In: Proceedings of hybrid artificial intelligent systems, Salamanca, Spain, 11–13 September 2013, pp 51–60

[2] Yang H, Huang K, King I et al (2009) Localized support vector regression for time series prediction. Neurocomputing 72(10):2659–2669

[3] Tomar D, Agarwal S (2011) Weighted support vector regression approach for remote healthcare monitoring. In: Proceedings of 2011 international conference on recent trends in information technology (ICRTIT), Chennai, India, 3–5 June 2011, pp 969–974

[4] Pontil M, Mukherjee S, Girosi F (2000) On the noise model of support vector machines regression. In: Proceedings of algorithmic learning theory, Sydney, Australia, 11–13 December 2000, pp 316–324

[5] Bludszuweit H, Domínguez-Navarro JA, Llombart A (2008) Statistical analysis of wind power forecast error. In: Proceedings of IEEE transactions on power systems, Quebec, Canada, 19–22 September 2008, pp 983–991

[6] Prada J, Dorronsoro JR (2015) SVRs and uncertainty estimates in wind energy prediction. In: Proceedings of international work-conference on artificial neural networks, Palma de Mallorca, Spain, 10–12 June 2015, pp 564–577

[7] Mcgovern A, Gagne DJ, Basara J et al (2015) Solar energy prediction: an international contest to initiate interdisciplinary research on compelling meteorological problems. Bull Am Meteorol Soc 96:1388–1395

[8] Hong T, Pinson P, Fan S et al (2016) Probabilistic energy forecasting: global energy forecasting competition 2014 and beyond. Int J Forecast 32(3):896–913

[9] Ettoumi FY, Mefti A, Adane A et al (2002) Statistical analysis of solar measurements in Algeria using Beta distributions. Renew Energy 26:47–67

[10] Shawe-Taylor J, Cristianini N (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

[11] Fletcher R (2013) Practical methods of optimization. Wiley, Chichester

[12] Minh HQ, Niyogi P, Yao Y (2006) Mercer's theorem, feature maps, and smoothing. In: Proceedings of international conference on computational learning theory, Pittsburgh, USA, 22–25 June 2006, pp 154–168

[13] Schölkopf B, Smola AJ (2002) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge

[14] Kivinen J, Smola AJ, Williamson RC (2004) Online learning with kernels. IEEE Trans Signal Process 52(8):2165–2176

[15] Lin C, Weng R (2004) Simple probabilistic predictions for support vector regression. National Taiwan University, Taipei

[16] Hartigan JA (1975) Clustering algorithms. Wiley, New York

[17] Huang Z (1998) Extensions to the *k*-means algorithm for clustering large data sets with categorical values. Data Min Knowl Discov 2(3):283–304

[18] Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. In: Scholkopf B, Burges CJC, Smola AJ (eds) Advances in kernel methods—support vector learning. MIT Press, Cambridge, pp 185–208

[19] Hu Q, Zhang S, Xie Z et al (2014) Noise model based *v*-support vector regression with its application to short-term wind speed forecasting. Neural Netw 57:1–11

[20] Klein JP, Keiding N, Kamby C (1989) Semiparametric Marshall–Olkin models applied to the occurrence of metastases at multiple sites after breast cancer. Biometrics 45(4):1073–1086

[21] Prada J, Dorronsoro JR (2017) General noise SVRs and uncertainty intervals. In: Proceedings of international work-conference on artificial neural networks, Cadiz, Spain, 14–16 June 2017, pp 734–746

[22] Kushner HJ, Clark DS (2012) Stochastic approximation methods for constrained and unconstrained systems. Springer, New York

[23] Chang C, Lin C (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):1–27

[24] Holmgren WF, Andrews RW, Lorenzo AT et al (2015) PVLIB python 2015. In: Proceedings of 42nd photovoltaic specialists conference, New Orleans, USA, 14–19 June 2015, pp 1–5

[25] Kaggle (2014) AMS 2013–2014 solar energy prediction contest. https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest/data. Accessed 10 October 2014

[26] Ineichen P (2008) A broadband simplified version of the Solis clear sky model. Solar Energy 82:758–762

[27] Fernandez A, Gala Y, Dorronsoro JR (2014) Machine learning prediction of large area photovoltaic energy production. In:

Proceedings of data analytics for renewable energy integration, Nancy, France, 19 September 2014, pp 38–53
[28] Assuncao HF, Escobedo JF, Oliveira AP (2003) Modelling frequency distributions of 5 minute-averaged solar radiation indexes using Beta probability functions. Theor Appl Climatol 75:213–224

J. PRADA received his double B.S. degrees in Informatics and Mathematics and double M.S. degrees of Applied Mathematics + Master of Investigation and Innovation in Information and Communications Technology, Machine Learning Specialty, from Universidad Autónoma de Madrid (UAM), Spain, in 2013 and 2015, respectively. He is currently finalizing his Ph.D. in the Machine Learning Field at UAM. He has more than five years of experience as a researcher in machine learning applied to real-world problems at Machine Learning Group at UAM. He has published four papers on this topic. He also has knowledge of the corporate world, having worked for more than 3 years as a data scientist for different companies ranging from healthcare startups to airline companies. His main research interests are machine learning, deep learning, SVMs, general cost functions, uncertainty intervals, renewable energy forecasting, and healthcare.

J. R. DORRONSORO received his B.S. degree in Mathematics at Universidad Complutense de Madrid, Spain, in 1977, and his Ph.D. degree in the same field at Washington University in St. Louis, USA, in 1982. He is a professor since 1997 at the Informatics Department of UAM, Spain. He has published more than 40 research papers, written 3 books, collaborated with chapters in 7 other books, and participated in 18 research projects throughout his career. His main research interests are machine learning, SVMs, neural networks, deep learning, spectral clustering and diffusion maps, sparse convex models, and data science.