SUPPORT VECTOR LEARNING BASED MODELING OF A SOLAR POWER PLANT

C. Pereira¹,² and A. Dourado¹

¹CISUC – Centro de Informática e Sistemas da Universidade de Coimbra, Pinhal de Marrocos, 3030 Coimbra, Portugal
Phone: +351 39 790000, Fax: +351 39 701266
email: {cpereira,dourado}@dei.uc.pt

²ISEC – Instituto Superior de Engenharia de Coimbra, Quinta da Nora, 3030 Coimbra, Portugal
Phone: +351 39 790200, Fax: +351 39 790270
email: cpereira@isec.pt

Abstract: Modeling and control of a solar power plant using support vector learning is considered in this work. The model is based on a radial basis function network architecture and uses subtractive clustering and support vector learning to find the parameters and size of the network. To achieve a more interpretable structure the proposed method proceeds in two phases. Firstly, the input-output data is clustered according to the subtractive clustering method. Secondly, the support vector learning algorithm finds the number and location of centers and the weights of the network. This approach will improve the interpretability analysis and reduces the complexity of the problem. The proposed learning scheme is applied to the distributed collector field of a solar power plant. An internal model control scheme is also suggested, in which the proposed strategy proves to be effective in modeling the plant dynamics and the corresponding inverse.

Keywords: Nonlinear modeling, RBF networks, support vector learning, subtractive clustering, internal model control, solar power plant.

1. INTRODUCTION

Artificial neural networks have attracted the growing interest of researchers for modeling and control of nonlinear systems. Neural networks (NN) are known to be universal approximators and provide the capability of learning. Most of the NN structures cannot incorporate linguistic system descriptions, because of the difficulties to extract knowledge from them in a comprehensible way. Also, there is no systematic way to set up the topology of a neural network. However, RBF neural networks are under certain conditions functionally equivalent to the fuzzy model with constant consequent parts, thus becoming interpretable structures. In addition support vector learning (SVL) is a valid way to set up its topology. In fact, an interesting option for designing radial basis function (RBF) network architecture is support vector learning. This is a technique proposed by Vapnik and co-workers (Vapnik, 1995), (Cortes and Vapnik, 1995) which in particular allows the construction of RBF networks by choosing a suitable kernel function. Instead of minimizing the empirical training error, like traditional methods, the support vector machines (SVM) aims at minimizing an upper bound of the generalization error, finding a separating hyperplane and maximizing the margin between the hyperplane and the training data. The particular attribute of a SVM is providing a good generalization performance despite the fact that it does not incorporate problem domain knowledge. The support vector algorithm provides a direct way of choosing the number and location of centers and weights of the network. Also SVM points an alternative point of view, with the centers being those examples which are critical to solve the given task. In the traditional view the centers are regarded as clusters.

The performance of a SVM largely depends on the kernel, however there is a lack of theory concerning how to chose good kernel functions in a data-dependent way. An information-geometric method of
modifying the kernel to improve performance has been suggested (Amari and Wu, 1999). If Gaussian functions are chosen, the width is a parameter that must be defined a priori. In addition, when designing a SVM for regression tasks the selection of the insensitivity zone for the loss function and the regularization parameter, which controls the trade-off between model complexity and training error is difficult. In practice, several values are used and cross-validation is performed. Also SVM needs to solve a quadratic optimization problem, exhibiting long running times, and in most cases slower training time than conventional neural networks. Iterative techniques that are easy to implement have been suggested (Platt, 2000; Joachims, 2000). Also for computational reasons, pre-processing the input-output data is a useful procedure. The proposed framework addresses these design difficulties. First, subtractive clustering is applied to input-output data. Second, support vector learning is used to compute the network parameters.

The presented framework is tested on a solar power plant. The main control requirement is to maintain the outlet oil temperature of the collector field at a pre-specified value. It is difficult to obtain a satisfactory performance over the whole operating range with a fixed linear controller since the solar radiation cannot be manipulated by the control system and it changes substantially during plant operation. Different authors have suggested intelligent control techniques such as neural networks (Arahal et al, 1997) or fuzzy systems (Berenguel et al, 1997). Commissioning of a switching controller, using different models of the plant for different operating points, has also been applied successfully (Henriques et al, 1999).

The Internal model control (IMC) structure has shown to be a suitable approach for nonlinear system control (Hunt and Sbarbaro, 1991). It incorporates models of the plant dynamics and the corresponding inverse, which are represented in this work with a RBF network. The particular generalization ability of support vector learning is applied in this paper to improve the modeling capacity of the intelligent structure.

The remainder of this paper is organized as follows. The next Section presents the network structure, describes the subtractive clustering method for input-output space clustering and introduces support vector machines. Description of the solar power plant and modeling results are given in Section 3. In Section 4 simulation results using an IMC controller are presented. In Section 5, conclusions are given.

2. NETWORK LEARNING METHOD

The proposed learning method proceeds on two distinct phases. First, subtractive clustering is applied, computing the number and position of the clusters representing the training data set. Second, local feature spaces are computed using support vector learning. The Gaussian functions are used as multidimensional membership functions.

2.1 Network Structure

A local basis function neural network has been adopted. The model belongs to a general class of function approximators, called the basis function expansion, taking the form:

\[ y = \sum_{i=1}^{m} \phi_i (x) \theta_i \]  

where \( \phi_i \) are the basis functions and \( \theta_i \) the consequent parameters. The well-known RBF neural networks belong to this class of models. These networks are under certain conditions functionally equivalent to the fuzzy model with constant consequent parts (Jang and Sun, 1993):

\[ R_i : \text{If} \ x \text{ is } A_i \text{ then } y = \theta_i, \ i = 1,2,\ldots,m \]  

where \( x \) is the antecedent linguistic variable, which represents the input to the fuzzy system, and \( y \) is the consequent variable representing the output. \( A_i \) is the linguistic term defined by the Gaussian multivariable membership function, and \( m \) denotes the number of rules.

Figure 1. Local basis function network structure.

The functional equivalence implies that the learning algorithms of RBF networks can be used to train models of this type. In most cases the RBF training proceeds in two phases. The basis functions are constructed by clustering the input training vectors in the input space and the consequent parameters, \( \theta_i \), are estimated from the data using least-squares methods. In a similar way, Takagi-Sugeno fuzzy models are constructed by product-space fuzzy clustering (Babuska, 1998). The way the input space is partitioned determines the number of fuzzy rules.

Usually, a model structure is created from numerical data in the form of If-Then rules. However these automated modeling techniques may introduce unnecessary redundancy into the rule base. It is of great interest to reduce the number of fuzzy rules or hidden layer units.

Support vector learning has been proposed to construct different types of learning machines. Applied to RBF networks the learning process using a given set of training data automatically determines the required number of hidden units (support vectors), its positions and weights. In this work, by using an effective partition of the input space, a more parsimonious structure is achieved.
2.2 Subtractive Clustering

In the proposed method, the input-output data is clustered according to the subtractive clustering method. The purpose of clustering is to classify a set of \( N \) data points \( X = \{x_1, x_2, \ldots, x_n\} \) into homogeneous groups of data \( P = \{p_1, p_2, \ldots, p_c\} \) with \( 1 \leq c \leq N \) (If the number of clusters \( c=1 \) all data belongs to the same class and if \( c=N \), each data sample defines a class). The fuzzy c-means clustering algorithm (Bezdek, 1981) is an extremely powerful classification method, which minimizes the Euclidean distance between each data point and its cluster center.

The number of clusters should reflect the level of the knowledge of the system under consideration or the level of generality in the user’s description of the system. The quality of the fuzzy c-means depends strongly on the choice of the number of centers and the initial cluster positions. The advantage of using the subtractive clustering algorithm is that the number of clusters does not need to be a priori specified, instead the method can be used to determine the number of clusters and their values.

The method is a modified form of the Mountain Method (Yager and Filev, 1994) for cluster estimation. Assuming \( N \) normalized points in an \( M \)-dimensional space, each data point is considered as a potential cluster center (Chiu, 1994) and defines a measure of the potential of data point \( x_i \) as:

\[
\text{Pot}_i = \sum_{j=1}^{N} e^{-\alpha ||x_i-x_j||^2} \tag{3}
\]

where \( \alpha = \frac{4}{r_a^2} \), and \( r_a \) is a positive constant (radius defining a neighborhood). The measure of potential for a given point is a function of its distances to all other data points. A point with many neighboring points will have a high potential value. After the potential of every data point has been computed the point with the highest potential \( x_i \) is selected as the first cluster center. Then the potential of each point \( x_i \) is updated by the formula:

\[
\text{Pot}_i \leftarrow \text{Pot}_i - \text{Pot}_i e^{-\beta ||x_i-x'||^2} \tag{4}
\]

where \( \text{Pot}_i \) is the potential of the selected cluster and \( \beta = \frac{4}{r_b^2} \), where \( r_b \) is a positive constant, somewhat greater than \( r_a \). When the potential of all data points have been updated according to equation 4, the data point with the highest remaining potential is selected as the next cluster to select, and the process repeats until a given threshold for the potential has been reached.

2.3 Support Vector Machines

The support vector learning, proposed by Vapnik, is a constructive learning procedure based on statistical learning theory. By choosing different kinds of kernels, this technique can be applied to a variety of representations such as multilayer perceptron neural networks, radial basis function networks, splines or polynomial estimators. It can be used either in classification or regression tasks. The SVM works by mapping the input space into a high-dimensional feature space using a set of nonlinear basis functions. The framework was originally designed for pattern recognition, but the basic properties carry over to regression by choosing a suitable cost function, like for example the e-insensitive or Vapnik’s loss function:

\[
L_e(d,y) = \begin{cases} y - f(x) - e, & \text{for } |y - f(x)| > e, \\ 0, & \text{otherwise} \end{cases} \tag{5}
\]

This is a linear function with an insensitive zone. The parameter \( e > 0 \) controls the width of the insensitive zone (errors below \( e \) are not penalized) and is usually chosen a priori.

In the support vector machines a function linear in the parameters is used to approximate the regression in the feature space:

\[
f(x) = \sum_{j=1}^{n} w_j g_j(x) \tag{6}
\]

where \( x \in \mathbb{R}^d \) represent the input, \( w_j \) the linear parameters and \( g_j \) denotes a nonlinear transformation. The goal is to find a function \( f \) with a small test error, based on training data \( (x_i, y_i), i=1, \ldots, n \), by minimizing the empirical risk:

\[
R_{\text{emp}}[f] = \frac{1}{n} \sum_{i=1}^{n} L_e(y, f(x)) \tag{7}
\]

subject to the constraint \( \|w\|^2 \leq c \), where \( c \) is a constant. The constrained optimization problem is reformulated by introducing the slack variables \( \xi_i \geq 0, \xi_i \geq 0, i=1, \ldots, n \):

\[
y_i - \sum_{j=1}^{n} w_j g_j(x) \leq e + \xi_i, \quad \sum_{j=1}^{n} w_j g_j(x) - y_i \leq e + \xi_i \tag{8}
\]

Assuming this constraint, the problem is posed in terms of quadratic optimization by introducing the following functional:

\[
Q_{\xi, \xi'}(w) = \frac{C}{n} \left( \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \xi_i' \right) + \frac{1}{2} w^T w \tag{9}
\]

The pre-determined coefficient \( C \) should be sufficiently large and affects the trade-off between complexity and the training error. Then this optimization problem is transformed into the dual problem by constructing a Lagrangian and applying the Kuhn-Tucker theorem (Vapnik, 1995). The regression function is then given by:

\[
f(x) = \sum_{i=1}^{n} (\alpha_i - \beta_i) K(x, x) \tag{10}
\]
where $K(x, x)$ represents the inner product kernel defined in accordance with Mercer’s Theorem. It is interesting to note that the explicit knowledge of $g_j$ is not really necessary, instead equation 1 is rewritten in terms of dot products. In the dual problem, the parameters $\alpha_i$ and $\beta_j$ are calculated by maximizing the functional:

$$Q(\alpha, \beta) = \sum_{i=1}^{n} \alpha_i (y_i - \beta_j) - \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j K(x_i, x_j)$$

subject to the following constraints

$$\sum_{i=1}^{n} (\alpha_i - \beta_j) = 0$$

$$0 \leq \alpha_i \leq \frac{C}{n}, \quad 0 \leq \beta_j \leq \frac{C}{n}, \quad i = 1, \ldots, n$$

(11)

(12)

In the SVM the nonlinear feature space is directly incorporated in the parameter optimization. By solving the quadratic optimization problem one obtains the number of hidden units, their values and the weights of the output layer. Choosing Gaussian functions as kernels:

$$K(x, x) = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right)$$

(13)

the support vector learning offers an alternative method to the design of RBF networks. The parameters to chose are the capacity control, the insensitive error and the spread of the Gaussian functions.

3. SOLAR POWER PLANT MODEL

The support based learning model is applied in this section to the distributed collector field of a solar power plant. The main characteristic of the solar plant is that the primary energy source, the solar radiation cannot be manipulated. The solar radiation changes substantially during plant operation, due to daily solar cycle, atmospheric conditions, such as clouds cover, humidity and turbidity leading to significant variations in the dynamic characteristics of the field, corresponding to different operation conditions. Therefore it is difficult to obtain a satisfactory model over the total operating range. The approach presented in this paper, using clustering to reduce the problem dimension and the generalization ability of the support vector machines can overcome this problem, thus achieving reasonable model characterizing the different operating regimes.

The Acurex distributed solar collector field of the solar power plant is well described in the literature (Camacho et al., 1998), and is located at the ‘Plataforma Solar de Almeria’, Spain. The field consists of 480 distributed solar collector arranged in 20 rows, which form 10 parallel loops. Each loop is 172 m long and the total aperture surface is 2672 m². A schematic diagram is shown in Figure 2. Each collector uses parabolic mirrors to concentrate solar radiation in a receiver tube. Synthetic oil is pumped through the receiver tube and picks up the heat transferred through the tube walls. The inlet oil, at temperature $T_{in}$, is pumped from the bottom of storage tank flowing through the collector field where its temperature is raised. After that, the fluid is introduced into the storage tank, from the top, to be used for electrical energy generation or feeding a heat exchanger in the desalination plant. The oil flow rate, $Q_{in}$, is the manipulated variable in the solar plant, and the main goal is to regulate the outlet field oil temperature, $T_{out}$, at a desired level, $T_{ref}$. The main disturbances are solar radiation $I_{rr}$ and the inlet oil temperature.

To deal with the several operating points some control strategies have been proposed. One of them applies adaptive control schemes, using local linear models of the plant (Camacho et al., 1998). Also a hierarchical control strategy consisting on a supervisory switching of PID controllers based on recurrent neural networks models was tested (Henriques et al., 1999).

![Fig. 2. Schematic diagram of the Acurex field.](image)

The considered model corresponds to the following regression NARX model:

$$T_{out}(k) = f(T_{out}(k-1), \ldots, T_{out}(k-n_o), I_{rr}(k-1), \ldots, I_{rr}(k-n_{rr}), Q_{in}(k-l), \ldots, Q_{in}(k-n_{Q_{in}}))$$

(14)

The figure 3 represents the model prediction for a given learning data set (VAF=95.9%). In these experiments, $n_o = n_{rr} = n_{Q_{in}} = 1$. The Figure 4 represents the model prediction for a validation data set (VAF=93.6%). Also in these experiments, the data set was normalized. These experiments confirm the excellent generalization ability when using support vector learning.

In order to compare the modeling ability, a traditional RBF network has been also applied, trained with k-means clustering and recursive least squares. The proposed model exhibits a good tradeoff between modeling error and number of units, achieving a parsimonious structure (see table 1). Also the data pre-processing brings computational advantages. In the table, the VAF criterion represents the percentile “Variance accounted for” measure between measured data and the model output:
\[ VAF = 100\% \left( 1 - \frac{\text{var}(y_1 - y_2)}{\text{var}(y_1)} \right) \]  

(15)

*VAF* equals 100% between two equal signals \( y_1 \) and \( y_2 \).

---

4. SOLAR POWER PLANT CONTROL

The internal model control is applied to control the outlet temperature of the collector field of the solar power plant. The scheme, represented in figure 5, incorporates the model of the plant dynamics and the corresponding inverse (Hunt and Sbarbaro, 1991). In the present work SVM is used for the construction of the plant model and its inverse.

These preliminary control results correspond to simulations using the non-linear distributed parameter model of the Acurex field (Berenguel *et al*., 1993).

Figure 6(a) represents the solar radiation and inlet oil temperature. As can be seen in figure 6(b) the controller behavior is quite acceptable. The disturbance rejection capabilities of the controller are also acceptable when intermittent clouds occurred.
5. CONCLUSIONS

This work has presented a framework for the construction of RBF network based on support vector learning and using subtractive clustering. The results prove the good approximation capabilities of the method, keeping a parsimonious network structure. The clustering of the input-output space is an effective way to decompose a large problem, which is difficult to solve using a support vector algorithm, in a smaller problem, easily solved.

Comparing to standard neural networks, the generalization ability is improved by the use of the support vector learning algorithm. In addition, it simplifies the design of the RBF networks, providing a direct way of choosing the number and location of centers and weights of the network. Future work will test the control strategy in the real plant and robustness analysis will be addressed in this context.

ACKNOWLEDGEMENTS

This work was partially supported by the Portuguese Ministry of Science and Technology, under programs PRAXIS/P/EEI/14155/1998 and ALCINE. The experiments described in this paper were carried out within the project Improving Human Potential program (EC-DGXII) supported by the European Union Program Training and Mobility of Researchers. The authors would like to express their gratitude to the personnel of the Plataforma Solar de Almeria.

REFERENCES

Amari S., S. Wu (1999), Improving support vector machine classifiers by modifying kernel functions, Neural Networks, 12, 783-789.


