A Hybrid Forecasting Methodology using Feature Selection and Support Vector Regression

José Guajardo, Jaime Miranda, and Richard Weber, Department of Industrial Engineering, University of Chile

Abstract

Various techniques have been proposed to forecast a given time series. Models from the ARIMA family have been successfully used as well as regression approaches based on e.g. linear, non-linear regression, neural networks, and Support Vector Machines. What makes the difference in many real-world applications, however, is not the technique but an appropriated forecasting methodology. Here we present such a methodology for the regression-based forecasting approach. A hybrid system is presented that iteratively selects the most relevant features and constructs the best regression model given certain criteria. We present a particular technique for feature selection as well as for model construction. The methodology, however, is a generic one providing the opportunity to employ alternative approaches within our framework.

Support Vector Regression; Time Series Prediction; Feature Selection.

1 Introduction

Forecasting problems can be solved basically using two different approaches: time series models, such as the ARIMA family of models or exponential smoothing on one hand, or on the other hand regression approaches, such as linear or non-linear regression, neural networks, regression trees, or Support Vector Regression. Hybrid models combining both approaches have been applied e.g. in [1].

We present a hybrid forecasting methodology combining feature selection with regression models where we propose Support Vector Regression for model construction. Our methodology, however, is independent of the particular regression model, i.e. any other regression approach can be used within the proposed framework.

Section II. provides a literature review related to feature selection and support vector regression. In section III. we develop the proposed hybrid forecasting methodology. Section IV. presents the results derived by our methodology as well as using alternative approaches. Section V. concludes this work and points at future developments.

2 Review of the related literature

The aim of this section is to provide a review of the major approaches for feature selection, as well as to describe the well known support vector regression algorithm. Both issues will be important to understand the methodology we are proposing.

2.1 Feature Selection

Given the complexity of certain data mining applications, such as e.g. regression, it is desirable to select the most important features to construct the respective model.

This problem motivates efforts in developing feature selection methods. Some of the benefits that can be obtained by performing feature selection are [13][21][29]: improvements in the accuracy of the forecasts delivered by the model, reducing computational times involved in model construction, facilitating data visualization and model understanding, and reducing the probability of overfitting.

Several approaches have been developed to carry out feature selection, being possible to group the existing methods in three major categories [13]: filters, wrappers and embedded methods.

Filter methods perform feature selection as a preprocessing step, independently of the learning algorithm used for model construction. An example of such a mechanism is variable ranking, using e.g. correlation coefficients between each feature and the dependent variable. Another filter approach selects features based on a linear model (this corresponds to the filter preprocessing step), and then constructs a non-linear model using the selected features (see e.g. [2]). As can be seen, filter mechanisms do not depend on the regression algorithm applied, and consequently the selected features are not related to it. This is one of their main weaknesses, but they are commonly applied for simplicity reasons. Also, filter methods in general do not take into account the relationship between variables.
On the other hand, \textit{wrapper methods} are characterized as being a subset selection approach. The main idea of wrapper methods is to assess subsets of variables according to their usefulness to a given learning algorithm [13]. Wrapper methods became popular by works such as [14][15]. Here, the learning algorithm is treated as a black box, and the best subset of features is determined according to the performance of the particular algorithm applied to build a regression model (e.g. linear or non-linear regression, neural networks, support vector machines, among others). Wrapper methods need a criterion to compare the performance of different feature subsets (e.g., minimization of the training error), as well as a search strategy to guide the process. \textit{Forward selection} and \textit{backward elimination} are two of the commonly used search strategies. A forward selection strategy starts with an empty set of features and incorporates in each iteration features considering their conjoint predictive power. On the other hand, a backward elimination procedure starts with all available features belonging to the set of selected features. In each iteration the feature which produces the minimum decrease in predictive performance is eliminated. Both strategies need a stopping criterion in order to determine the "best" feature subset.

Wrappers are often criticized because they seem to be a "brut force" method requiring exhaustive computation [13], but at the same time have the advantage of taking into account the performance of the predictive model to be used, as well as evaluating how a given subset of features performs as a whole.

Finally, \textit{embedded methods} incorporate feature selection as part of the training process, i.e., feature selection is done when building the predictive model. Such mechanisms usually involve changes in the objective function of the applied learning algorithm and therefore are commonly associated to a specific predictor. Examples of embedded methods are decision trees [3] [20], and the mechanisms developed in [17][21][30].

The methodology proposed in this paper uses wrapper methods and applies forward selection to guide the search strategy.

2.2 Support Vector Regression

Here we describe the standard Support Vector Regression (SVR) algorithm, which uses the so-called \(\epsilon\)-insensitive loss function, proposed by Vapnik [26]. This function allows a tolerance degree to errors not greater than \(\epsilon\). The description is based on the terminology used in [23][19].

Let \((\vec{x}_1, y_1), \ldots, (\vec{x}_l, y_l)\), where \(\vec{x}_i \in \Re^n\) and \(y_i \in \Re\) \(\forall i\), be the training data points available to build a regression model. The SVR algorithm applies a transformation function \(\Phi\) to the original data points\(^1\) from the initial Input Space, to a generally higher-dimensional Feature Space \((F)\). In this new space, we construct a linear model, which corresponds to a non-linear model in the original space:

\[
\Phi : \Re^n \longrightarrow F, w \in F
\]

\[
f(x) = \langle \vec{w}, \Phi(x) \rangle + b
\]

(1)

The goal when using the \(\epsilon\)-insensitive loss function is to find a function that fits current training data with a deviation less or equal to \(\epsilon\), and at the same time is as flat as possible. This means that one seeks for a small weight vector \(\vec{w}\), e.g. by minimizing the norm \(||\vec{w}|||^2\) [23]. The following optimization problem is stated for such purpose:

\[
\begin{align*}
\text{Min} & \quad \frac{1}{2}||\vec{w}||^2 \\
\text{s.t.} & \quad y_i - \langle \vec{w}, \Phi(x_i) \rangle - b \leq \epsilon \\
& \quad \langle \vec{w}, \Phi(x_i) \rangle + b - y_i \leq \epsilon
\end{align*}
\]

(2)

(3)

(4)

This problem could be infeasible. Therefore, slack variables \(\xi_i, \xi_i^*\) are introduced to allow error levels greater than \(\epsilon\), arriving to the formulation proposed in [26]:

\[
\begin{align*}
\text{Min} & \quad \frac{1}{2}||\vec{w}||^2 + C \cdot \sum_{i=1}^l (\xi_i + \xi_i^*) \\
\text{s.t.} & \quad y_i - \langle \vec{w}, \Phi(x_i) \rangle - b \leq \epsilon - \xi_i \\
& \quad \langle \vec{w}, \Phi(x_i) \rangle + b - y_i \leq \epsilon - \xi_i^* \\
& \quad \xi_i, \xi_i^* \geq 0 \quad i = 1, \ldots, l
\end{align*}
\]

(5)

(6)

(7)

(8)

This is known as the primal problem of the SVR algorithm. The objective function takes into account generalization ability and accuracy in the training set, and embodies the structural risk minimization principle [27]. Parameter \(C\) measures the trade-off between generalization ability and accuracy in the training data, and parameter \(\epsilon\) defines the degree of tolerance to errors.

To solve the problem stated in equation 5, it is more convenient to represent the problem in its dual form. For this purpose, a Lagrange function is constructed, and once applying saddle point conditions, the following dual problem is obtained:

\(^1\)When the identity function is used, i.e. \(\Phi(x) \longrightarrow X\), no transformation is carried out and linear SVR models are obtained.
The solution of this quadratic optimization problem will be function of the dual variables $\alpha_i$ and $\alpha_i^*$. To return to the original problem, it can be shown that the following relationships hold [27]:

$$\begin{align*}
\text{Max} & - \frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle \Phi(x_i), \Phi(x_j) \rangle \\
& - \varepsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i (\alpha_i - \alpha_i^*) \\
\text{s.t. } & \alpha_i, \alpha_i^* \in [0, C] \\
& \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \quad (9)
\end{align*}$$

The following figure provides a general view of our framework for SVR-based forecasting.

The proposed methodology can be summarized as follows: first, calculate base parameters ($\epsilon$, $C$, and kernel parameter) that work well under some general conditions. Next, and using base parameters $\epsilon^*$, $C^*$, $K^*$, perform feature selection using a wrapper method with forward selection strategy to obtain the best set of predictor variables $X^*$. Finally, using predictors $X^*$, return to the problem of model construction performing Grid Search around base parameters to get the optimal parameters $\epsilon^*$, $C^*$, $K^*$. Thus, at the end, the predictive model is determined by parameters $\epsilon^*$, $C^*$, kernel function $K^*$, and predictors $X^*$.

The final element in this framework is model updating.

### 3.2 Model Construction using Support Vector Machines within the Proposed Methodology

A critical issue in SVR is the selection of the model parameters $\epsilon$ and $C$, as well as the kernel function used to encode data to a higher dimensional space.

As discussed in section (3.1), our approach to deal with model construction calculates initial values for SVR parameters, and then performs grid search around them. This structure is shown in figure (2):

To calculate initial values for parameters $\epsilon$ and $C$, we use the empirical rules proposed by Cherkassky et al [6] (figure (2)). Experiments using various time series performed by the authors of this paper show that these rules are more effective than others like e.g. those proposed in [16].

Regarding kernel function, we use the RBF kernel transformation to the original data, since this function has performed well under some general conditions [23], and is the most commonly used in practice [5][6][9][16][18][28]. To set an appropriate value for the parameter of this kernel function ($\sigma$), we carry out some exploratory experiments.
trying different settings, and choosing the best one in terms of the results obtained in this phase.

Once defined the initial values for the three parameters, we select the best set of features (see next subsection). Using the selected features the well-known Grid Search mechanism \cite{4}\cite{9}\cite{18}\cite{24} around base parameters $\epsilon'$, $C'$, $K'$ is performed. The best point of the grid will define the final parameters of the SVR model.

### 3.3 Feature Selection within the Proposed Methodology

As discussed in section (2.1), there are several feature selection approaches in the context of time series prediction. We propose a wrapper method inspired by ideas presented in \cite{15}, which selects feature guided by a forward selection strategy. Figure (3) displays such a strategy:

![Figure 3. Feature selection heuristic](image)

3.4 Model Updating within the Proposed Methodology

The task of developing forecast models for a time series problem could be affected by changes in the phenomena we are studying over time. Therefore a static model that works well in some period could provide poor predictions in some future period. To deal with this issue, we designed a way to address model updating \cite{11} when building a predictive model, in which the basic idea is to define a two part training set: the first part contains historical data, and the second part contains the most recent data. When a predefined number of new samples arrives, these observations are incorporated into the training set. This way patterns in new data are taken into account in model construction. Finally, the proportion of data belonging to the training and validation sets is kept stable over time by shifting data points from the historical part of the training set to the validation set, when we perform model updating.

### 4 EXPERIMENTS AND RESULTS

We applied the proposed methodology to a real-world sales prediction problem, in which a company wants to predict the number of units that will be sold the following week, for 5 different products.

Consequently, we analyzed 5 time series with weekly sales data during the period from January 2001 to September 2004 each. Figure (4) shows one of these 5 series, which presents a strong monthly seasonality, increasing sales numbers during each month, reaching its peak in the last week of each month.

---

Given the set of initial predictors $(x_1, ..., x_n)$, we define a maximum number of predictors ($m$) to be included in the predictive model, taking into consideration the nature of the problem and the number of training data points. Also, the number of tuples\textsuperscript{2} to be achieved in each iteration ($k$) has to be defined. The greater the value of $k$, the greater the size of the search space induced by the strategy. Once defined both values, the strategy starts building models with each individual variable as the single predictor. In the second iteration, we mix the $k$ best individual predictors $(1 - tuple)$ with the remaining variables, and keep the $k$ best $2 - tuple$ of predictors for the next iteration, and so on until we have the $k$ best $m - tuples$ of predictors. Finally, we choose the best tuple of variables among the $m \cdot k$ best $i - tuples$ computed in the iterative process ($i : 1, \ldots, m$), to be the predictors $X^*$ that will be used in the final SVR model.

\textsuperscript{2}In this context, an $i$-tuple means a combination of $i$ different predictors used together in a predictive model.
Next, the proposed methodology will be applied and the obtained results are provided. Finally, we compare our results with the well-known ARMAX approach and neural networks.

We applied the proposed methodology to the 5 time series using a set of 23 initial features (independent variables). As a result we obtained for each series a different set of parameters describing the respective model. Following the above described wrapper strategy we also determined for each series a set of selected features. This way we defined the overall relevance of each one of the original features by simply counting how often each feature has been selected by the applied methodology.

The following listing shows the features ordered according to their average relevance from the most important one to the least important one for all 5 series.

- Normalized number of week within a month (taking values between 0 and 1)
- Sales one week before
- Sales two weeks before
- Binary variable indicating if the month under consideration has 4 or 5 weeks
- Sales eight weeks before
- Categorical variable indicating if the week under consideration contains holidays of certain categories
- Sales 13 weeks before
- Sales 14 weeks before
- Ordinal variable indicating the year under consideration (2001, ..., 2004)
- Sales seven weeks before
- Sales twelve weeks before
- Number of the week under consideration (taking values between 1 and 195 which is the number of weeks within the analyzed period)
- Number of month within a quarter (taking values 1,2,3)
- Sales three weeks before
- Sales six weeks before
- Sales ten weeks before
- Sales four weeks before
- Sales five weeks before
- Number of week within a year (taking values 1,..., 52)
- Number of month within a year (taking values 1,..., 12)
- Sales nine weeks before
- Number of quarter within a year (taking values 1,..., 4)
- Sales eleven weeks before

We applied the final model provided by the proposed methodology SVM-UP to the test set from April 2004 to September 2004.

Alternatively we developed forecasting models for each series using the well-known ARMAX approach as well as a multilayer perceptron (MLP) within the proposed methodology, i.e. a MLP-type neural network as regression technique instead of SVR. Since a wrapper strategy is employed the set of selected features depends on the regression techniques used. In the special application described here different sets have been identified in the cases of MLP and SVR.

Applying the three models to the same evaluation set of all 5 time series we obtained the results provided in the following table, which shows the average test set error levels using as error function MAPE (mean average percentage error).

Compared with ARMAX the proposed methodology using SVR (SVM-UP) gives slightly better results in 3 out of 5 cases and also in average over all 5 series.

Using a neural network within the proposed methodology (NN-UP) outperforms ARMAX also in 3 out of 5 cases but predicts worse on average.
The advantage of the proposed methodology, however, is not only limited to the forecasting performance, it provides also the capability of selecting the most relevant features and updating the respective model.

### 5 CONCLUSIONS AND FUTURE WORKS

We presented the hybrid methodology SVM-UP for time series forecasting, which combines a wrapper method with forward selection strategy to perform feature selection and regression model construction using Support Vector Machines. Model selection is based on calculating initial values for the SVR parameters and then performing grid search around them. The final component of our methodology is model updating: we define a training set formed by past and most recent information. When a predefined number of new observations arrives, this most recent information is incorporated into the training set, and the proportion of data belonging to the training and validation sets is kept stable over time by shifting data points from the historical part of the training set to the validation set.

We have applied the proposed methodology to a sales forecasting problem and compared its performance to a standard ARMAX approach and neural networks. Comparing the respective results shows that our methodology performs slightly better and additionally provides a selection of the most important features. This last point increases the comprehension of the phenomenon we are studying, and could be useful to understand certain variables involved in the final regression model and their influence on the dependent variable, e.g. the number of sold units.

Major advantages of the proposed methodology are expected when dealing with dynamic phenomena, where the performance of a forecasting model could be significantly improved by performing model updating. First results show that our methodology provides very good results for seasonal time series (see [11],[12]).

Future work has to be done for predicting non-seasonal time series and for selecting the most appropriate parameters of the kernel function based on theoretical approaches.

### Acknowledgment

This work has been supported partially by the Millennium Nucleus "Complex Engineering Systems" (www.sistemasdeingenieria.cl) and the Fondecyt project 1040926.

The first author would like to thank Penta Analytics S.A., a Chilean consulting company where he carried out part of this work during 2004.

### References


<table>
<thead>
<tr>
<th>Test Set Error</th>
<th>ARMAX Models (%)</th>
<th>NN-UP Models (%)</th>
<th>SVM-UP Models (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>10.98</td>
<td>14.31</td>
<td>11.86</td>
</tr>
<tr>
<td>P2</td>
<td>13.64</td>
<td>14.66</td>
<td>11.44</td>
</tr>
<tr>
<td>P3</td>
<td>6.87</td>
<td>6.66</td>
<td>6.51</td>
</tr>
<tr>
<td>P4</td>
<td>10.53</td>
<td>10.56</td>
<td>11.01</td>
</tr>
<tr>
<td>P5</td>
<td>10.90</td>
<td>9.85</td>
<td>9.98</td>
</tr>
<tr>
<td>Average (5 Prod.)</td>
<td>10.58</td>
<td>11.21</td>
<td>10.16</td>
</tr>
</tbody>
</table>

Table 1. Average Test Set Error Levels.


