Applied Soft Computing 10 (2010) 276–283

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A model updating strategy for predicting time series with seasonal patterns

Jose A. Guajardo [a], Richard Weber [b,*], Jaime Miranda [c]

[a] OPIM Department, The Wharton School, University of Pennsylvania, 3730 Walnut st., Philadelphia, USA
[b] Department of Industrial Engineering, University of Chile, Republica 701, Santiago, RM, Chile
[c] Department of Management Control and Information Systems, Universidad de Chile, Diagonal Paraguay 257, Santiago, Chile

ARTICLE INFO

Article history:
Received 7 February 2007
Received in revised form 30 June 2009
Accepted 26 July 2009
Available online 4 August 2009

Keywords:
Model updating
Dynamic models
Support vector regression
Time series prediction

ABSTRACT

Traditional methodologies for time series prediction take the series to be predicted and split it into training, validation, and test sets. The first one serves to construct forecasting models, the second set for model selection, and the third one is used to evaluate the final model. Different time series approaches such as ARIMA and exponential smoothing, as well as regression techniques such as neural networks and support vector regression, have been successfully used to develop forecasting models. A problem that has not yet received proper attention, however, is how to update such forecasting models when new data arrives, i.e. when a new event of the considered time series occurs.

This paper presents a strategy to update support vector regression based forecasting models for time series with seasonal patterns. The basic idea of this updating strategy is to add the most recent data to the training set every time a predefined number of observations takes place. This way, information in new data is taken into account in model construction. The proposed strategy outperforms the respective static version in almost all time series studied in this work, considering three different error measures.

1. Introduction

Time series involve dynamic phenomena, being one of the challenges in their analysis to discover patterns that govern the series behavior. Since these patterns may vary over time, it is reasonable to explicitly take into account possible changes that could affect the studied phenomena.

For instance, in a sales prediction problem, factors such as market changes, the actions of competitors and the development of new products, among others, could influence sales pattern, which therefore may suffer changes over time.

To obtain systematically good predictions in such a case, it may be necessary to update the respective models from time to time. Similar situations occur when dealing with predictive models to estimate, e.g. stock prices, the number of customers in a given market, as well as in other applications from business forecasting; see e.g. [15,25].

There exist several approaches to develop forecasting models. Two of the most common ones are on the one hand time series methods, with techniques like ARIMA and exponential smoothing and on the other hand regression methods, with techniques such as linear regression, neural networks (NN), and support vector regression (SVR). Especially SVR received recently a lot of attention in time series forecasting [17].

A practical approach to build dynamic models for time series prediction with seasonal patterns using regression is proposed. In particular, SVR will be used, but the model updating strategy presented in this paper can be applied using any regression technique.

The main contribution of the proposed approach is to introduce a systematic methodology for model updating this way avoiding the time-consuming task of model construction.

This paper is organized as follows. Section 2 provides a review of the literature related to model updating, as well as a description of the support vector regression (SVR) algorithm. In Section 3, the proposed model updating strategy is explained in detail. Section 4 presents experiments and results obtained when applying this strategy to different time series. In each application a comparison with the static case is provided. Finally, in Section 5 the main conclusions and possible extensions of this work are discussed.

2. Review of related literature

2.1. Model updating

Model updating has received increasing interest in the data mining community where a diverse terminology such as, e.g.

* Corresponding author. Fax: +56 2 978 4011.
E-mail addresses: jguajard@gmail.com (J.A. Guajardo), rweber@dii.uchile.cl (R. Weber), jmirandap@fen.uchile.cl (J. Miranda).

1568-4946/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.07.005

concept drift, incremental learning, stream data mining, or dynamic data mining is used.

First a brief explication of each one of these concepts is provided followed by an introduction to what shall be understood as model updating in this paper.

Concept drift [37] is known as the phenomenon of changes in the target concept of a learning task where a difficult problem is to distinguish between true drift and noise. Furthermore, it might be desirable to detect recurring concepts that may be due to cyclic phenomena, as is the case of the present paper. Methods to handle concept drift using example selection and example weighting are presented in [19], where support vector machines are employed for classification. Techniques to update a classifier given a drifting concept are presented, e.g. in [5] where changes to a decision tree are proposed in order to maintain its performance.

The most appropriate approach in the presence of concept drift is incremental learning [32] where the respective models are updated online as new instances become available. Batch learning systems, instead, examine a collection of instances adapting the respective model "*from time to time*".

Incremental concept formation using machine learning has been studied, e.g. in [23], where a new learning approach based on the theory of formal concept analysis (FCA) has been suggested.

Using neural networks for incremental learning can lead to the well-known phenomenon of "*catastrophic forgetting*". In [31] solutions for this problem based on the pseudo-rehearsal mechanism are analyzed and new developments are presented.

The approach presented in this paper learns incrementally but with an appropriate time window instead of an adaptation each time a new observation becomes available.

Recently, mining stream data has attracted interest in the respective research community, see e.g. [8,3]. Typical applications are the analysis of transaction data, e.g. for credit card fraud detection and time series analysis. One problem in this field is to find a trade-off between memory usage, i.e. window size and mining accuracy.

Dynamic data mining [36] has a broader view since it includes all data mining aspects where a changing environment is considered explicitly. Examples are analyses of object-describing trajectories instead of pictures taken at a certain point of time; see e.g. [18]. Dynamic class structures are considered, e.g. in [10] where a methodology for updating clustering models using fuzzy logic has been presented.

This paper proposes how to update regression models that will be used for time series prediction. In order to avoid confusion, a clear distinction between *model updating* and *forecast updating* is necessary.

In *forecast updating*, predictions determined in the past are updated using the most recent information available but maintaining the respective model unchanged. In *model updating*, however, the predictive model is updated, i.e. its predictors and parameters.

Forecast updating has been employed in various real-world applications. It affects, e.g. pricing decisions in a supply chain when a supplier updates his/her forecasts of a distributor's demand; see [13]. The respective research has considered techniques like ARIMA [6,21], where the effect of improving forecasts of aggregated data (e.g. quarterly aggregates) using disaggregated data (e.g. monthly data) is analyzed. Other related issues are, e.g. the evaluation of the respective forecast accuracy provided at different instants of time and the determination of the most convenient time lag to provide forecasts.

The added value of forecast updating has been studied in various experiments; see e.g. [22]. For trended time series it could be shown that incorporating the most recent observation improved forecast accuracy [30]. For relatively stable time series, however, there appeared to be no value in updating time series forecasts.

The problem of model updating, however, has not yet received proper attention in literature so far and is the main focus of the present paper.

2.2. Support vector regression

The standard SVR algorithm is described, which uses the so-called ε-*insensitive* loss function proposed by Vapnik [34]. This function allows a tolerance degree to errors not greater than ε. The description is based on the structure and terminology used in [28,33].

Let $(x_1, y_1), \ldots, (x_\ell, y_\ell)$, where $x_i \in \Re^n$ and $y_i \in \Re \, \forall i = 1, \ldots, l$, be the training data points available to build a regression model. The SVR algorithm applies a transformation function Φ to the original data points from the initial *Input Space* ($\Re^n$) to a generally higher-dimensional *Feature Space* (F). In this new space, a linear model f is constructed, which represents a non-linear model in the original space:

$$\begin{aligned} &\Phi : \Re^n \to F, w \in F \\ &f(x) = \langle w, \Phi(x) \rangle + b \end{aligned} \tag{1}$$

When the identity function is used, i.e. $\Phi(x) \to x$, no transformation is carried out and linear SVR models are obtained.

The goal when using the ε-*insensitive* loss function is to find a function f that fits given training data with a deviation less or equal to ε, and at the same time is as flat as possible in order to reduce model complexity. This means that one seeks a small weight vector w. One way to ensure this is by minimizing the norm $\|w\|^2$ [26,33] leading to the following optimization problem.

$$\begin{aligned} &\underset{w,b}{\text{Min}} && \frac{1}{2}\|w\|^2 \\ &\text{s.t} && \\ &y_i - \langle w, \Phi(x_i) \rangle - b \le \varepsilon && \forall i = 1, \ldots, \ell. \\ &\langle w, \Phi(x_i) \rangle + b - y_i \le \varepsilon && \forall i = 1, \ldots, \ell. \end{aligned} \tag{2}$$

This problem could be infeasible. Therefore, slack variables ξ_i, ξ_i^* are introduced to allow error levels greater than ε, arriving at the following formulation:

$$\begin{aligned} &\underset{w,b}{\text{Min}} && \frac{1}{2}\|w\|^2 + C \cdot \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) \\ &\text{s.t} && \\ &y_i - \langle w, \Phi(x_i) \rangle - b \le \varepsilon - \xi_i && \forall i = 1, \ldots, \ell. \\ &\langle w, \Phi(x_i) \rangle + b - y_i \le \varepsilon - \xi_i^* && \forall i = 1, \ldots, \ell. \\ &\xi_i, \xi_i^* \ge 0 && \forall i = 1, \ldots, \ell. \end{aligned} \tag{3}$$

This is known as the primal problem of the SVR algorithm; see e.g. [34]. The objective function takes into account generalization ability and accuracy in the training set, and embodies the structural risk minimization principle [35]. Parameter C indicates the trade-off between generalization ability and accuracy in the training data, and parameter ε defines the degree of tolerance to errors.

To solve the problem stated in Eq. (3), 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,

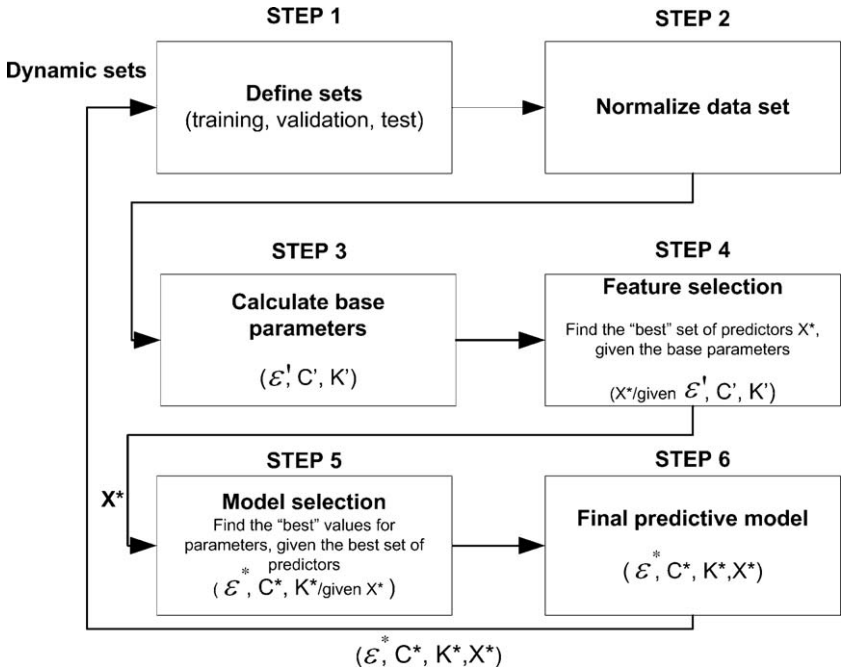

Fig. 1. The SVR-UP strategy.

the following dual problem is obtained:

$$\underset{\alpha,\alpha^*}{\text{Max}} -\frac{1}{2}\sum_{i,j=1}^{\ell}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle\Phi(x_i),\Phi(x_j)\rangle-\varepsilon\sum_{i=1}^{\ell}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{\ell}y_i(\alpha_i-\alpha_i^*) \tag{4}$$

s.t.

$$\alpha_i,\alpha_i^*\le C \qquad \forall i=1,\ldots,\ell \tag{5}$$

$$\alpha_i,\alpha_i^*\ge 0 \qquad \forall i=1,\ldots,\ell \tag{6}$$

$$\sum_{i=1}^{\ell}(\alpha_i-\alpha_i^*)=0 \tag{7}$$

This is the quadratic optimization problem which has to be solved to obtain the solution of the SVR model. This solution will be function of the dual variables α_i and α_i^*. Using saddle point conditions it can be shown that Eq. (8) holds [35]. Replacing this expression in Eq. (1), the final solution of the SVR algorithm is obtained (see Eq. (9)):

$$w=\sum_{i=1}^{\ell}(\alpha_i-\alpha_i^*)\Phi(x_i) \tag{8}$$

$$f(x)=\sum_{i=1}^{\ell}(\alpha_i-\alpha_i^*)K(x_i,x)+b \tag{9}$$

Here, the expression $K(x_i,x)$ is equal to $\langle\Phi(x_i),\Phi(x)\rangle$, which is known as the *Kernel Function* [34]. The existence of such a function allows us to obtain a solution for the original regression problem, without concerning about the transformation $\Phi(x)$ applied to the data.

3. SVR-UP: a model updating strategy for time series prediction using support vector regression

In this section, the strategy to develop dynamic forecasting models using support vector regression is described. First, a general perspective of this strategy called SVR-UP is presented briefly, to show how it performs feature selection, model selection, and model updating. Then, its details are provided.

3.1. General view of the proposed model updating strategy

The proposed model updating strategy combines feature selection using a wrapper method with model selection using grid search. It has been developed using SVR but any other regression technique can be used as well.

Fig. 1 illustrates the proposed strategy. The first step consists of dividing the data into training, validation, and test subsets. Training data is used for model construction, validation data for model and feature selection, and test data is a completely independent subset which provides an estimation of the error level the model would have in future applications. As will be seen later, these are dynamic subsets since model updating is performed.

Once all the variables have been normalized in order to have values from the same scale (step 2), some base parameters are fixed and then feature and model selection are performed. The basic idea of this strategy can be summarized as follows. First, base parameters (ε, C, and kernel parameter) are calculated that work well under some general conditions (step 3). Using these base parameters ε', C', K', feature selection using a wrapper method with forward selection strategy is carried out to obtain the best set of features X^* (step 4). Finally, using X^*, the final model parameters ε^*, C^*, K^* are determined (step 5) applying *Grid Search* around base parameters. This way, we arrive at a predictive SVR model which is determined by parameters ε^*, C^*, kernel function K^*, and features X^* (step 6).

Base parameters ε' and C' are calculated using the empirical rules proposed by Cherkassky and Ma [9], which gave good results in various time series applications. Finally, a RBF kernel transformation is applied to the original data, fixing its parameter (σ) based on exploratory analyses.

Feature selection (step 4) is applied using a wrapper method with forward selection, inspired by Kohavi and John (see [20]) as described next where the terms predictors, features and variables are used indifferently. A maximum number of possible predictors *maxp-tuple* is fixed regarding the nature of the problem and the number of training data points.

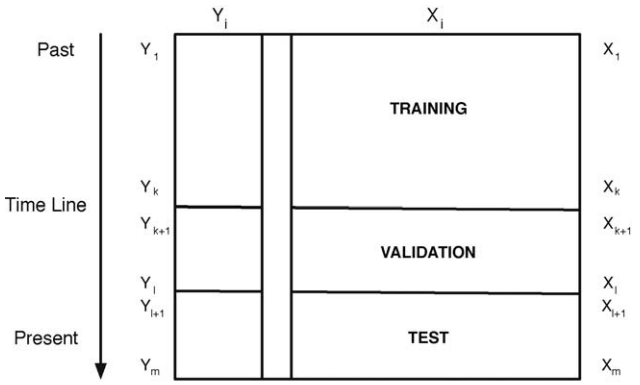

Fig. 2. Static configuration.

The feature selection method starts building models with each individual variable as single predictor. In the second iteration, the best individual predictors are combined with the other variables. The best 2-*tuple* of predictors are kept for the next iteration, and so on until the best *maxp-tuple* of predictors is obtained. Finally, the best tuple of variables among all the best i-*tuple* determined in the iterative process ($i = 1, \ldots, max\,p$) are chosen, to be the features X^* that will be used in the final SVR model.

Model selection (step 5) consists of performing the well-known *Grid Search* mechanism [7,12,27], around base parameters ε', C', K'. The best point of the grid is the set of model parameters ϵ^*, C^*, K^* that will be used for prediction. Fast grid search (see [4]) is an interesting alternative for parameter selection in time-critical applications since it determines parameters much faster than conventional grid search. Since our applications are not time-critical and fast grid search did not provide a superior forecasting performance we applied conventional grid search in our experiments.

The final element of our framework – and the central issue of the present paper – is model updating, which will be explained in detail in the following subsection.

3.2. Development of the proposed model updating strategy

Let $(x_1, y_1), \ldots\ldots, (x_m, y_m)$, where $x_i \in \Re^n$ and $y_i \in \Re\, \forall i = 1, \ldots, m$, be the data points available to build a regression model for predicting the values of the time series defined by y_i. We suppose the series to be characterized by seasonal patterns of length c, i.e. a season is formed by cycles of c data points.

3.2.1. The static case

The static approach sequentially splits the data into training, validation, and test sets.

The static configuration of these 3 subsets is represented by a matrix as shown in Fig. 2 where index 1 represents the position of the initial training data point and index k the position of the final training data point. The validation set has indices $(k+1, l)$ and test indices are $(l+1, m)$.

The indices k and l are defined to obtain the desired proportion of data belonging to each subset.

3.2.2. The dynamic case

The model updating strategy has been designed to deal with time series with seasonal patterns. A complete seasonal pattern will be called a cycle; examples are, e.g. monthly data with yearly seasonality, where a cycle is defined by a year (see Fig. 3) or weekly data with monthly seasonality, where a cycle is defined by a month.

First, the length of the cycles of the series has to be identified. For this purpose, graphical inspection, autocorrelation analysis or spectral analysis can be carried out. For example, the monthly series shown in Fig. 3 is characterized by yearly cycles.

Once identified the length of the series' cycles, a test set containing at least two cycles is defined (step 1 in Fig. 1). This test set is divided into subsets, each one containing a single cycle, as shown in Fig. 4.

Let there be p cycles of length c, i.e. $(m\text{-}l+1) = p \cdot c$ observations belonging to the test set. In the static case, just one model to predict all observations contained in the test set would be constructed, as well as for future observations, maintaining this model throughout the entire procedure unchanged. The main idea of the proposed updating strategy, however, consists of developing different predictive models for each cycle of the test set, as well as for any future cycle by considering the most recent information for model construction, i.e. adding it into the training set.

Next, the training and validation sets have to be configured for predicting the first cycle of the test set. Fig. 5 displays such configuration.

As can be seen, training data contains two parts where the first part is a set of past (or historical) data and the second part contains the most recent information.

An entire cycle prior to the first cycle of the test set is part of the second training set. This way, we ensure that most recent information influences model construction.

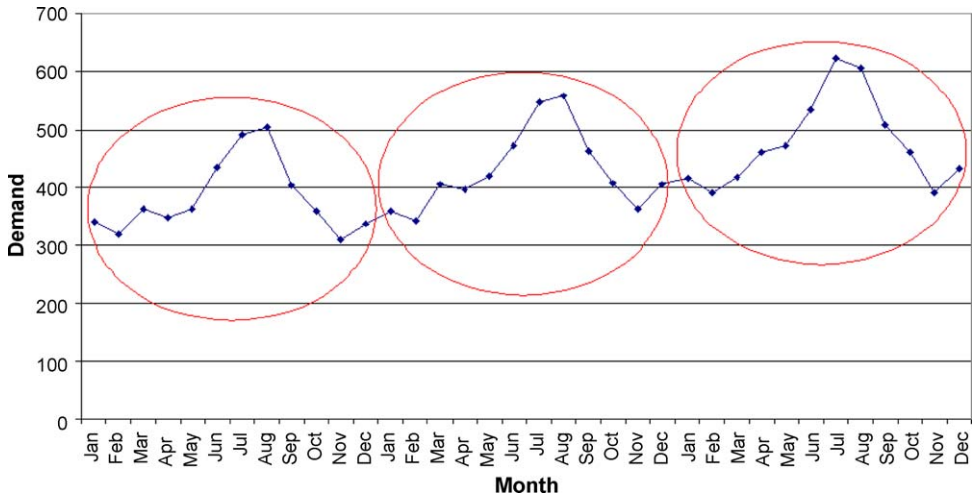

Fig. 3. Yearly cycles in monthly data.

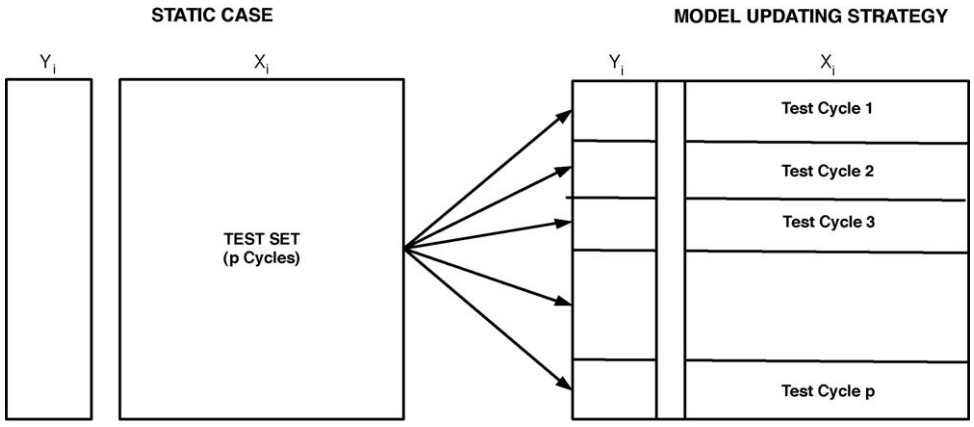

Fig. 4. Splitting the test set.

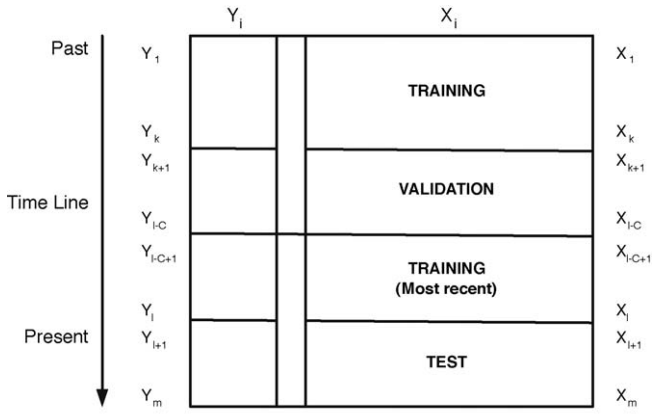

Fig. 5. Predicting the first cycle (C_1).

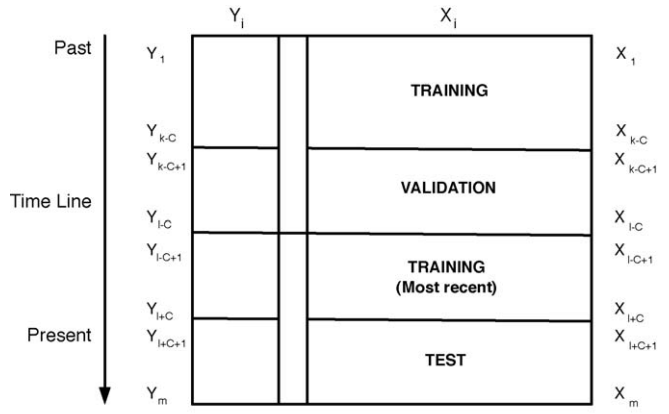

Fig. 6. Predicting the second cycle (C_2).

During the following cycles the desired proportions of data belonging to the training set (*PropTr*) and validation set (*PropVa*) have to be maintained. These are determined as in the static case and defined as:

$$PropTr = \frac{k + c}{l} \tag{10}$$

$$PropVa = \frac{l\text{-}c\text{-}k}{l} \tag{11}$$

These proportions do not consider data points belonging to the test set. To predict the second cycle of the test set, the first cycle of the test set (this data is now part of the past) is added to the most recent part of the training set, and a different predictive model having the configuration shown in Fig. 6 is built.

The proportions of data belonging to the training and validation sets have to be kept stable over time. For this purpose, data from the first part of the training set is moved to the validation set (see Fig. 6) as long as no data is moved from the most recent training set to the validation set.

The number of data points to be moved is determined according to the original proportion of data in each subset; see *PropTr* and *PropVa* defined above.

This strategy will be applied to build predictive models for the rest of the cycles of the test set, as well as for any future observations. When building a predictive model for a generic cycle, the first cycle of the previous test set is moved to the most recent training set. Then we move the first cycle from the most recent training set to the validation set and – if necessary – the first cycle from the validation set to the first training set in order to maintain the established proportions.

4. Experiments and results

The proposed SVR-UP strategy has been applied to two different problem domains. The following two subsections describe the respective data sets, the problems to be solved, the application of the strategy SVR-UP, and the respective results for each domain.

Static versions are obtained by applying the proposed methodology without the updating step described in Fig. 1, while dynamic models use this step for each cycle of the test set. Static models are called SVR-Stat, whereas dynamic models are called SVR-UP.

4.1. Time series from the M1 competition

The M1 competition consists of a set of 1001 time series from the business domain [24]; the data set is available, e.g. in http://forecasters.org/.

Four different time series with seasonal patterns from the M1 data set have been considered: Series 21 (QRM1), Series 27 (QNI10), Series 51 (MNM15), and Series 98 (MNG37). The first two series (21 and 27) consist of quarterly data, and the last two series (51 and 98) consist of monthly data. Those four series have been selected because they match the requirements to apply the proposed model updating strategy.

For quarterly series, the test set is formed by the last 2 years of information (8 observations). One static model is constructed for predicting the test set, and 2 different models are constructed

Table 1
Test set errors for four series of the M1 competition

Series	SVR-Stat	SVR-UP
Test set MAPE (%)		
Series 21	20.99	17.44
Series 27	7.09	3.98
Series 51	15.42	8.74
Series 98	2.92	2.88
Test set MAE		
Series 21	56.6	46.0
Series 27	8.1	4.5
Series 51	2625.0	1347.2
Series 98	3.3	3.3
Test set RMSE		
Series 21	63.6	50.8
Series 27	9.8	5.4
Series 51	3112.0	1629.2
Series 98	4.6	5.0

when applying our model updating strategy (one for each year of the test set since each year defines a different cycle).

For monthly series, the test set is formed by the last 18 observations (1.5 years). Every 6 months a new cycle of the series is defined, therefore when using the proposed dynamic strategy three different predictive models are constructed (one for each semester of the test set).

Lags of the series and temporal attributes are defined as possible predictors for the four series, and a maximum of 5 and 8 predictors to be included both in static and dynamic models are defined in the cases of quarterly and monthly series, respectively.

The employed error measures are: MAPE (mean average percentage error), MAE (mean average error), and RMSE (root mean square error).

Results obtained using SVR-Stat and SVR-UP models are shown in Table 1. In most of the cases improvements are achieved when applying the proposed model updating strategy; exceptions are series 98 using MAE (accuracy is maintained in this case) and RMSE.

4.2. Sales prediction problem

The proposed model updating strategy has also been applied to a real-world problem, where a company wants to predict the number of units that will be sold 1 week ahead. For confidentiality reasons the company's name and its products, as well as the specific industry can not be disclosed. Weekly sales data for 5 different products from January 2001 to September 2004 was provided to build and test the models.

These five time series are characterized by a strong monthly seasonality, as is shown exemplarily in Figs. 7 and 8 for the case of product P1.

Consequently, each month corresponds to a different cycle. Data from April 2004 to September 2004, containing 6 different cycles has been used as test set. Applying the model updating strategy, different predictive models will be constructed for these 6 cycles. Training and validation sets are defined in a proportion of around 6–1.

In the feature selection step lags of the series and temporal attributes (month, holidays, etc.) have been identified as possible variables of the regression model (see also [16]).

Table 2 shows the test set error obtained for each one of the 5 products, using static and dynamic SVR models.

As can be seen, improvements in all 5 series are obtained when using dynamic SVR models instead of static ones, being this validated by using 3 different error measures.

The above presented results show considerable improvements in the experiments conducted. Nevertheless, statistical relevance of these improvements has to be tested.

For this purpose, a paired samples t-test has been performed for each error measure, since forecasts generated by different models for a given time series are not statistically independent. The same procedure was applied, e.g. in [29]. In our case, we compare static and dynamic versions of SVR.

The paired samples t-test provided the following results for the three different error measures: MAPE ($t = 6.873$, df $= 181$, $p < 0.001$), MAE ($t = 5.312$, df $= 181$, $p < 0.001$), RMSE ($t = 3.556$, df $= 181$, $p < 0.001$).

These results show that improvements obtained by applying the proposed model updating strategy are statistically significant.

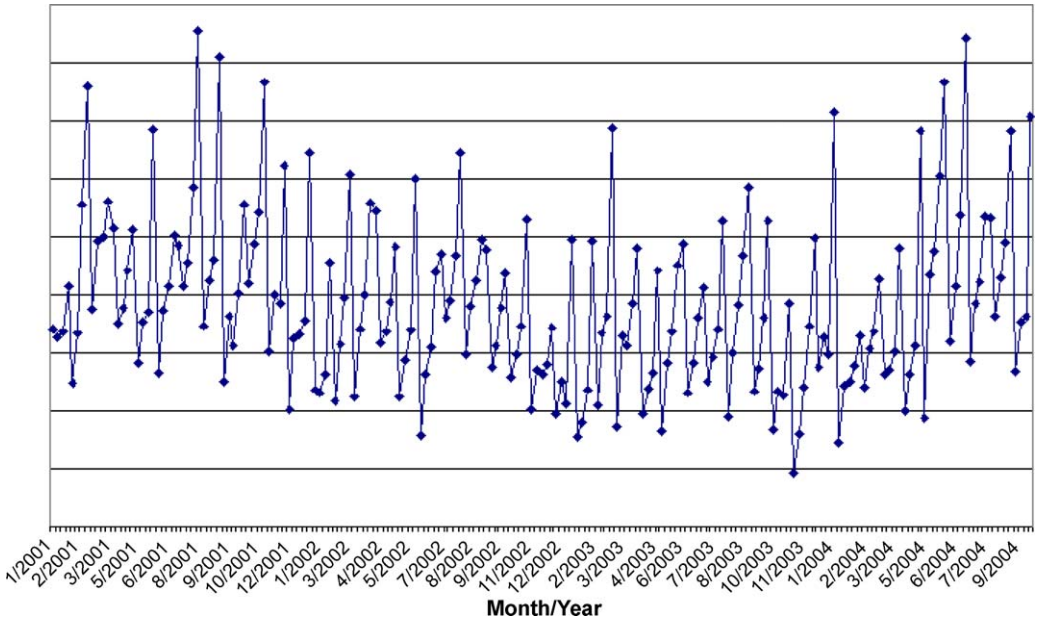

Fig. 7. Weekly sales series for product P1 (01/2001-09/2004).

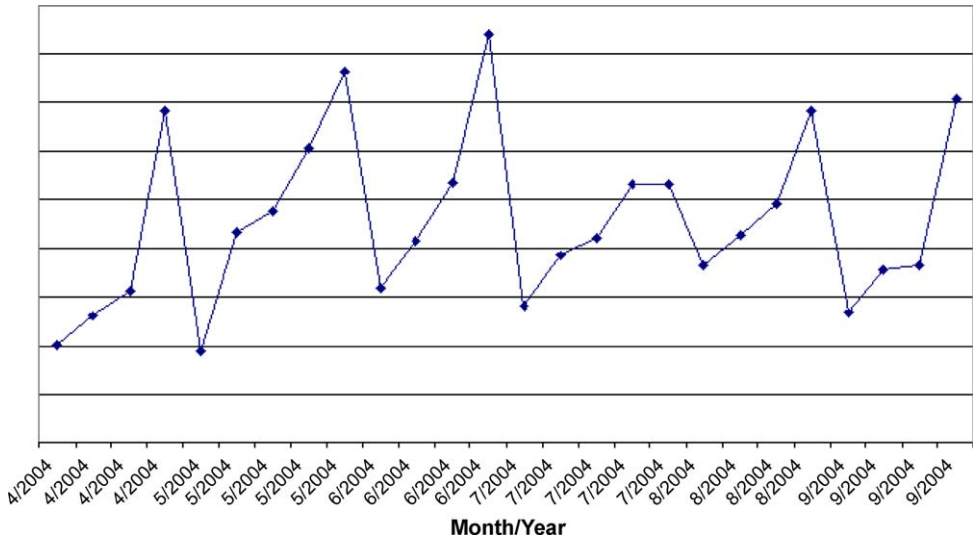

Fig. 8. Zoom of sales series for product P1 (04/2004-09/2004).

Table 2
Test set errors for five series of a sales prediction problem

Series	SVR-Stat	SVR-UP
Test set MAPE (%)		
Product P1	30.14	11.86
Product P2	14.89	11.44
Product P3	7.00	6.51
Product P4	15.24	11.01
Product P5	13.04	9.98
Test set MAE		
Product P1	721.8	342.3
Product P2	357.2	283.3
Product P3	104.7	96.4
Product P4	386.7	283.9
Product P5	407.7	264.1
Test set RMSE		
Product P1	835.4	534.7
Product P2	408.7	351.6
Product P3	147.4	138.2
Product P4	501.0	375.2
Product P5	550.6	369.1

5. Conclusions and future work

A model updating strategy for time series with seasonal patterns has been presented, that can be used when support vector regression is applied for time series prediction. The main characteristics of this updating strategy are:

- A test set containing at least two cycles is defined. Different predictive models are constructed for predicting different cycles of the test set.
- A training set consisting of two parts is defined: the first part contains historical data, and the second part contains the most recent data ("most recent" in the sense of the test set we are trying to predict).
- Predictive models are updated every time a complete new cycle is received; this information is incorporated into the most recent training set. This ensures that both, historical as well as most recent patterns are taken into account in model construction.
- The proportion of data belonging to the training and validation sets is kept stable over time by shifting data points from the training set to the validation set, each time when model updating is carried out.

This model updating strategy has been applied to predict in total nine time series from two different domains. The use of 3 different error measures (MAPE, MAE and RMSE) validates the results. Improvements were obtained in almost all the series studied in this work.

The findings obtained in this paper underline that future work has to be done in the field of model updating for time series prediction, since dynamic phenomena could be better predicted using dynamic models. Next, some ideas regarding future work are presented.

In the proposed approach historical information is never removed. Since short time spans were covered in the time series analyzed in this paper, there was no need to deal with this issue. For longer ranges of information, it could be undesirable to maintain all the data being part of the predictive model, since the underlying distribution of the data could change over time.

The following idea for removing old data of the formulation (Batch removing strategy) is proposed: define a number N of cycles for which all the data available to build models are maintained, without removing samples. During this period, the proposed model updating strategy is applied. The oldest N cycles of information will be removed when N cycles of new data had arrived. When this occurs, the respective indices are redefined to represent exactly the initial situation, but now with new data.

Another interesting issue regarding future work is model adaptation between two consecutive cycles. The proposed model updating strategy considers a retraining approach under certain well defined conditions and updating rules for the training and validation sets when new information is received. Alternatively, it could be interesting to develop an updating strategy which successfully updates a predictive model each cycle by just adapting the model's parameters instead of complete retraining.

As extensions of the presented work, it would be interesting to adapt our model updating strategy to alternative regression approaches (e.g. linear regression, non-linear regression, neural networks [11,14], and hybrid approaches [1,12]) as well as to time series techniques other than regression-based approaches, such as ARIMA or exponential smoothing.

Acknowledgment

Support by the Scientific Millennium Institute "Complex Engineering Systems" (www.sistemasdeingenieria.cl) and the Chilean Fondecyt project 1040926 is greatly acknowledged.

References

[1] L. Aburto, R. Weber, Improved supply chain management based on hybrid demand forecasts, Applied Soft Computing 7 (1) (2007) 136–144.
[2] L. Aburto, R. Weber, A sequential hybrid forecasting system for demand prediction, in: F. Petra Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition. LNAI 4571, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 518–532.
[3] C.C. Aggarwal, Data Streams—Models and Algorithms, Springer, Berlin Heidelberg, 2007.
[4] Y. Bao, Z. Liu, A fast grid search method in support vector regression forecasting time series, in: E. Corchado (Ed.), IDEAL, LNCS 4224, Springer-Verlag, 2006 , pp. 504–511.
[5] M. Black, R.J. Hickey, Maintaining the performance of a learned classifier under concept drift, Intelligent Data Analysis 3 (6) (1999) 453–474.
[6] G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, 1970.
[7] C. Chang, C. Lin, LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.
[8] J.H. Chang, W.S. Lee, Efficient mining method for retrieving sequential patterns over online data streams, Journal of Information Science 31 (5) (2005) 420–432.
[9] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neural Networks 17 (1) (2004) 113–126.
[10] F. Crespo, R. Weber, A methodology for dynamic data mining based on fuzzy clustering, Fuzzy Sets and Systems 150 (2) (2005) 267–284.
[11] S. Crone, Stepwise selection of artificial neural network models for time series prediction, Journal of Intelligent Systems 14 (2–3) (2005).
[12] A. Demiriz, K. Bennett, C. Breneman, M. Embrechts, Support vector machine regression in chemometrics, Computing Science and Statistics (2001).
[13] K.L. Donohue, Efficient supply contracts for fashion goods with forecast updating and two production modes, Management Science 46 (11) (2000) 1397–1411.
[14] J. Faraway, C. Chatfield, Time series forecasting with neural networks: a compartive study using the airline data, Journal of the Royal Statistical Society Series C-Applied Statistics 47 (1998) 231–250.
[15] R. Fildes, et al., Generalising about univariate forecasting methods: further empirical evidence, International Journal of Forecasting 14 (3) (1998) 339–358.
[16] J. Guajardo, J. Miranda, R. Weber, A hybrid forecasting methodology using feature selection and support vector regression., in: Proc. of Fifth International Conference on Hybrid Intelligent Systems HIS 2005, Rio de Janeiro, (2005), pp. 341–346.
[17] W. He, Z. Wang, H. Jiang, Model optimizing and feature selecting for support vector regression in time series forecasting, Neurocomputing 72 (2008) 600–611.
[18] A. Joentgen, L. Mikenina, R. Weber, H.-J. Zimmermann, Dynamic fuzzy data analysis based on similarity between functions, Fuzzy Sets and Systems 105 (1) (1999) 81–90.
[19] R. Klinkenberg, Learning drifting concepts: example selection vs. example weighting, Intelligent Data Analysis 8 (3) (2004) 281–300.
[20] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.
[21] S.G. Koreisha, Y. Fang, Updating ARMA predictions for temporal aggregates, Journal of Forecasting 23 (4) (2004) 275–296.
[22] M. Lawrence, M. O'Connor, Sales forecasting updates: how good are they in practice? International Journal of Forecasting 16 (2000) 369–382.
[23] M. Maddouri, Towards a machine learning approach based on incremental concept formation, Intelligent Data Analysis 8 (3) (2004) 267–280.
[24] S. Makridakis, A. Anderson, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, R. Winkler, The accuracy of extrapolation (time series) methods: results of a forecasting competition, Journal of Forecasting 1 (1982) 111–153.
[25] S. Makridakis, M. Hibon, The M3-competition: results, conclusions and implications, International Journal of Forecasting 16 (2000) 451–476.
[26] D. Mattera, S. Haykin, Support Vector Machines for Dynamic Reconstruction of a Chaotic System, in: B. Schölkopf, J. Burges, A. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999.
[27] M. Momma, K.P. Bennett, A pattern search method for model selection of support vector regression, in: Proc. of SIAM Conference on Data Mining, Arlington, Virginia, 2002.
[28] K. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik, Using Support Vector Machines for Time Series Prediction, in: B. Schölkopf, J. Burges, A. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999.
[29] M. Nelson, T. Hill, W. Remus, M. O'Connor, Time series forecasting using neural networks: should the data be deseasonalized first? Journal of Forecasting 19 (1999) 359–367.
[30] M. O'Connor, W. Remus, K. Griggs, Does updating judgmental forecasts improve forecasts accuracy? International Journal of Forecasting 16 (1999) 101–109.
[31] A. Robins, Sequential learning in neural networks: a review and a discussion of pseudorehearsal based methods, Intelligent Data Analysis 8 (3) (2004) 301–322.
[32] J.C. Schlimmer, R.H. Granger, Incremental learning from noisy data, Machine Learning 1 (3) (1986) 317–354.
[33] A.J. Smola, B. Schölkopf, A Tutorial on Support Vector Regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
[34] V. Vapnik, The nature of statistical learning theory, Springer-Verlag, New York, 1995.
[35] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.
[36] R. Weber, Dynamic Data Mining, in: J. Wang (Ed.), Encyclopedia of Data Warehousing and Mining, 2nd Edition, IGI Global, 2009, pp. 722–728.
[37] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Machine Learning 23 (1) (1996) 69–101.