ISSN 2281-4299

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

An application of learning machines to sales forecasting under promotions

Gianni Di Pillo
Vittorio Latorre
Stefano Lucidi
Enrico Procacci

Technical Report n. 4, 2013

An application of learning machines to sales forecasting under promotions *

G. Di Pillo†, V. Latorre†, S. Lucidi†, E. Procacci‡

Abstract

This paper deals with sales forecasting in retail stores of large distribution. For several years statistical methods such as ARIMA and Exponential Smoothing have been used to this aim. However the statistical methods could fail if high irregularity of sales are present, as happens in case of promotions, because they are not well suited to model the nonlinear behaviors of the sales process. In the last years new methods based on Learning Machines are being employed for forecasting problems. These methods realize universal approximators of non linear functions, thus resulting more able to model complex nonlinear phenomena. The paper proposes an assessment of the use of Learning Machines for sales forecasting under promotions, and a comparison with the statistical methods, making reference to two real world cases. The learning machines have been trained using several configuration of input attributes, to point out the importance of a suitable inputs selection.

Keywords. Learning Machines, Neural networks, Radial basis functions, Support vector machines, Sales forecasting, Promotion policies, Nonlinear optimization.

*This work was partially supported by ACT Solutions under contract 581/2009 on "Forecasting by Neural Networks and SVM".

†Dipartimento di Ingegneria Informatica Automatica e Gestionale, Università di Roma "La Sapienza", via Ariosto 25 - 00185 Roma, Italy. E-mails: dipillo@dis.uniroma1.it, latorre@dis.uniroma1.it, lucidi@dis.uniroma1.it

‡ACT Solutions SRL, via Nizza 45 - 00198 Roma, Italy. E-mail: enrico.procacci@act-operationsresearch.com

1 Introduction

This paper is concerned with sales forecasting in a retail store of large distribution. In past times managers of these stores normally used their experience to predict the daily sales and to decide the resupply quantities. In more recent years, with the development of computer aided decision making, especially in the bigger firms, the use of mathematical methods has became more and more widespread. In years 70s and 80s the principal methods used were statistical methods based on time series autoregressive models, like the ARIMA method, the Box-Jenkins' method and the Winter's exponential smoothing method (see e.g. [19]). The data used by these methods are taken from the same time series that one wants to forecast, and that can be therefore considered as an output series.
In the 90s the new mathematical model of Artificial Neural Network (ANN), based on the brain's neurons interconnection structure, was developed and employed also for forecasting applications. A neural network has a more flexible structure than the usual statistical models, and it bases its prediction not only on the output series, but also on several input series on which the output may depend. These input series are called attributes of the output.
The basic structure of an ANN is a multilayer network of neurons, with one input layer, one or more hidden layers and one output layer. Each neuron is characterized by an activation function that depends on some parameters. Neurons are connected by weighted arcs. Making an ANN able to perform a forecast corresponds to tuning its parameters and weights.
An alternative characterization of the artificial neuron of an ANN is obtained, rather than in terms of activation functions, in terms of radial basis functions (RBF). The structure of an ANN of RBF is given by the input layer, only one hidden layer and the output layer. In the following we will make use, for sales forecasting, both of multilayer and RBF neural networks.
By the end of 90s, a mathematical model different than ANN was also developed for classification and forecasting, named Support Vector Machine (SVM). The analytical roots of SVM are in the Statistical Learning Theory, the algorithmic roots for its training are in the duality theory of Mathematical Programming. Since its introduction, the SVM has been considered a valid competitor of the ANN in the same fields of application.
Multilayer ANN, RBF ANN and SVM belong to the class of Learning Machines, machines that adapt themselves by a training process using given sets of input and output data so as to forecast outputs corresponding to different sets of given input data, not used for training. In all cases the training process is performed by solving mathematical optimization problems. Once trained, the learning machine provides a surrogate model of a complex unknown phenomenon.
The literature on Learning Machines and their training is huge. We confine ourselves to cite only some introductory references, like [3], [11] and [26] as concerns ANN, [5] and [8] as concerns SVM, [4], [12], [21] and [24] as concerns, more in general, learning machines.
In this paper the complex phenomenon of concern is how the amount of sales of a given commodity depends on different suitable input attributes. The aim of the paper is to assess the relative effectiveness of the three kinds of learning machines considered before in sales forecasting, also in comparisons with time series based methods, using the real data of a retail store. A distinguishing features of the paper is that it focuses on the effects of an

abnormal input attribute, that is occurrence of promotions on sales.
There are several works in literature that deal with these issues. One of the first works dating to the 90s, [2] showed the superiority of ANN on the ARIMA method in sales forecasting. A state of the art on the use of ANN in forecasting as in 1997 is provided in [28]. In [1] several comparisons are made between learning machines and statistical methods, showing from empirical results that learning machines have an edge on statistical methods especially in periods of volatile economic conditions. Sales forecasts on a weekly basis using different inputs are obtained in [22] and [23], proving again the efficacy of ANN. As concerns SVM, their potential application in sales forecasting is dealt with in [16]. Other works focus on the flexibility of learning machines. For example in [15] fuzzy neural networks, and in [7] both fuzzy neural networks and clustering methods, are used, to improve neural networks results. In [14] and [27] particular optimization procedures are used, like genetic algorithms or swarm optimization, to improve the forecast and to obtain better results than the statistical methods. In a more general framework, see [9] and [25], the authors use learning methods in the economical context of marketing for predicting consumer's future choices.

The paper is organized as follows. In Section 2 we shortly describe the learning machines employed for forecasting and the optimization problems to be solved in their training. In Section 3 we consider the implementation issues to be taken into account in the practical applications of learning machines. In Section 4 we describe the experimental environment of our application, making use of real sales data from two retail stores of large distribution. In Section 5 we report and analyze the results obtained in sales forecasting under promotion policies using the different learning machines. Section 6 summarizes some concluding remarks.

This work has been developed in cooperation with a specialized company vendor of a multi-paradigm forecasting platform and willing to improve the sales forecast under difficult conditions (slow movers products, sales under promotions).

2 Learning Machines

In this section we will describe shortly the mathematical models of the learning machines that we use for forecasting, and the related optimization problems to be solved in their training.

2.1 Multilayer artificial neural networks

The structure of a multilayer ANN is inspired by the brain's structure of evolved organisms. Basically, like the brain, it is a network formed by simple units that are linked by connections. Every single unit of the network, called neuron, processes an input vector $x \in \Re^n$ weighted by a vector of weights $w \in \Re^n$ according to an activation function g that compares the weighted input vector $w^T x$, with a threshold value θ, giving an output $y(x) = g(w^T x - \theta)$. A multilayer ANN is composed of:

- a number of n input units, without elaboration capabilities, that are associated to the n attributes in input to the network,

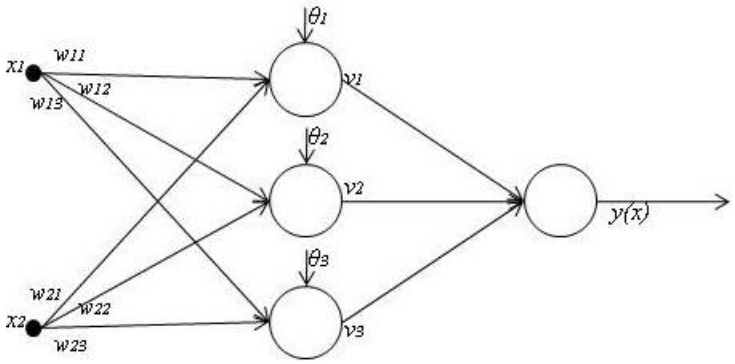

Figure 1: A two-layer artificial neural network.

- a set of N artificial neurons, characterized by activation functions, organized in $L \geq 2$ layers with $L-1$ hidden layers in which the output of every layer is the input of the successive layer,
- an output layer with $K \geq 1$ neurons that are associated to the outputs of the network,
- a set of oriented and weighted arcs that represent the connections between neurons. We suppose that there are no connections between neurons of the same layer, and that there are only forward connections without feedback ones.

As an example, we show in Fig. 1 a multilayer ANN with 2 input attributes, 1 output, 2 layers, 1 hidden layer with 3 artificial neurons. Note that a threshold value θ can be considered as the weight of a dummy input equal to -1.

A basic result in the theory of ANN states that given any continuous function $f(x)$, $f : \Re^n \to \Re$ defined on a compact set $\mathcal{C} \subset \Re^n$, it is possible to build a 2-layer network as the one in Fig. 1 with the property that, for any $\epsilon > 0$ it results

$$\max_{x \in \mathcal{C}} |f(x) - y(x)| < \epsilon,$$

provided that the activation function g is continuous and not polynomial. Therefore, a 2-layer with 1 hidden layer ANN can be considered as an *universal approximator* for continuous functions on compact sets.
On the basis of this result, we will adopt for our multilayer ANN the architecture described in Fig. 1. With this kind of architecture the output function of the network $y(x)$ is given by:

$$y(x) = \sum_{i=1}^{N} v_j g(w_j^T x - \theta_j),$$

where:

- w_j: $n-$vector of weights of the connection between each unit of the input layer and neuron j of the hidden layer,

- θ_j: threshold value for neuron j,
- v_j: weight of the connection between each neuron of the hidden layer and the output neuron.

As to the activation function g, in our application we will use the sigmoid function:

$$g(t) = \frac{1}{1 + e^{-\sigma t}},$$

with $\sigma = 0.5$.
Once specified the architecture and the activation function, the knowledge gained by training is stored in the connections between neurons, in particular it is stored in the weights associated with every connection, including the dummy ones that may represent the thresholds. The learning process of the ANN consists in adjusting $w_j, \theta_j, v_j, j = 1, \ldots, N$, in such a way that the output $y(x)$ of the ANN is able to predict the value $f(x)$ produced in a given environment by the input x.
The learning process makes use of a training set

$$\mathcal{T} = \{(x^p, y^p), x^p \in \Re^n, y^p \in \Re, \quad p = 1, ..., P\},$$

where P is the cardinality of the set, (x^p, y^p) is an input-output pair, a sample of the relation that we want to reproduce. Let us denote by w the $n \times N$ dimensional vector collecting as subvectors the weights $\{w_j, j = 1, \ldots, N\}$, by θ and v the $n-$vectors with components $\theta_j, v_j, j = 1, \ldots, N$, and by $y(x^p; w, \theta, v)$ the output of the network given the input x^p and the weights w, θ, v. Then the training is based on the solution of an unconstrained optimization problem of the kind:

$$\min_{w,\theta,v} E(w, \theta, v) = \frac{1}{2} \sum_{p=1}^{P} (y(x^p; w, \theta, v) - y^p)^2 + \gamma_1 \|w\|^2 + \gamma_2 \|\theta\|^2 + \gamma_3 \|v\|^2, \tag{1}$$

where $\gamma_1, \gamma_2, \gamma_3 > 0$ and $\|.\|$ denotes the Euclidean norm.
In the function $E(w, \theta, v)$ the first term measures the distance between the output of the network $y(x^p; w, \theta, v)$ and the real output y^p. As to the remaining three terms, they add a penalty on the norm of the weights w, θ, v that makes compact the level sets of the objective function $E(w, \theta, v)$, and regularizes the class of functions realized by the network; the first effect is beneficial for the convergence of the training algorithm, the second one is exploited in cross-validation of the network, as we will mention in the following.

2.2 Neural networks of radial basis functions

The neural networks of RBF have been introduced as a tool for interpolating multivariate functions. Given again a set

$$\mathcal{T} = \{(x^p, y^p), x^p \in \Re^n, y^p \in \Re, p = 1, ..., P\}, \tag{2}$$

where (x^p, y^p) are respectively arguments and values of a function $f(x), f : \Re^n \to \Re$, and given a continuous radial basis function $\phi(r), \phi : \Re^+ \to \Re$, a RBF interpolation of $f(x)$ is a function $y(x)$ obtained as a weighted sum of terms $\phi\|(x - x^p)\|$, with weights v_p:

$$y(x) = \sum_{p=1}^{P} v_p \phi(\|x - x^p\|), \tag{3}$$

and with the property that

$$y(x^p) = f(x^p) = y^p, \quad p = 1, \ldots, P. \tag{4}$$

The function ϕ is called radial basis function from the fact that its argument is the radial distance $r = \|x - x^p\|$. One of the most used RBF is the multi quadratic inverse RBF, given by:

$$\phi(\|x - x^p\|) = (\|x - x^p\|^2 + \sigma^2)^{-1/2},$$

that we adopt in our application, with $\sigma = 0.5$.
We note that condition (4) imposes to find a function $y(x)$ that matches perfectly the pairs of the set $\mathcal{T}$. This requires the solution of the $P \times P$ system of equations

$$y(x^q) = \sum_{p=1}^{P} v_p \phi(\|x^q - x^p\|), \quad q = 1, \ldots, P,$$

in the unknowns $v_p,\ p = 1, \ldots, P$. Since this system may be very large in practical applications, the so called *generalized* RBF have been introduced, where the interpolating function $y(x)$ is obtained as:

$$y(x) = \sum_{i=1}^{N} v_i \phi(\|x - c_i\|), \tag{5}$$

where $N \leq P$ and $c_i \in \Re^n, i = 1, \ldots, N$ are so-called *centers* of the RBF.
From (5) we see that the function $y(x)$ can be considered as the output of a 2-layer ANN, where N neurons in the hidden layer process the input x by means of the activation function $\phi(||x - c_i\|), i = 1, \ldots, N$, and the output neuron performs the weighted sum of the outputs of the N neurons with weights $v_i, i = 1, \ldots, N$.
The fact that the generalized RBF ANN can be viewed as a 2-layer ANN allows to demonstrate that a generalized RBF ANN enjoys the same property of being an universal approximator of continuous function on compact sets, in the same sense of the 2-layer ANN.
The training problem of a generalized RBF neural network can be formulated in a way similar to that of the 2-layer ANN: denote now by v and c the N-vectors collecting the weights $v_i, i = 1, \ldots, N$ and centers $c_i, i = 1, \ldots, N$, and by $y(x; v, c)$ the output of the generalized RBF ANN given by (5); then the training consists in solving the unconstrained minimization problem:

$$\min_{v,c} E(v, c) = \frac{1}{2} \sum_{p=1}^{P} (y(x^p; v, c) - y^p)^2 + \gamma_1 \|v\|^2 + \gamma_2 \|c\|^2, \tag{6}$$

where $\gamma_1, \gamma_2 > 0$. For the terms on the r.h.s. of (6) the same considerations made for the r.h.s. of (1) can be repeated.

2.3 Support vector machines

The SVM have been developed in the context of Statistical Theory of Learning, originally for solving classification problems. Later their use has been extended to regression problems. As before, let $\mathcal{T}$ given by (2) be a set of input-output samples (x^p, y^p). A *linear* SVM aims to realize a linear regression function

$$y(x) = w^T x + b$$

with the property that for each sample the regression error $y(x) - y^p$ is bounded by a value $\epsilon \geq 0$ so that:

$$|y^p - w^T x^p - b| \leq \epsilon,\ p = 1, \ldots, P,$$

and with the property of being as much flat as possible, where flatness is measured by the squared norm of w. Therefore we are lead to the problem:

$$\begin{array}{l} \min_{w,b} \ \ \frac{1}{2}\|w\|^2 \\ |y^p - w^T x^p - b| \leq \epsilon, p = 1, \ldots, P \end{array} \tag{7}$$

However, problem (7) could be infeasible. To tackle this possible failure, slack variables $\xi^p, \hat{\xi}^p, p = 1, \ldots, P$ are introduced, and Problem (7) is modified as follows:

$$\begin{array}{ll} \min_{w,b,\xi,\hat{\xi}} & \frac{1}{2}\|w\|^2 \ + \ C \sum_{p=1}^{P} (\xi^p + \hat{\xi}^p) \\ & w^T x^p + b - y^p \leq \epsilon + \xi^p \\ & y^p - w^T x^p - b \leq \epsilon + \hat{\xi}^p \qquad p = 1, \ldots, P, \\ & \xi^p,\ \hat{\xi}^p \geq 0. \end{array} \tag{8}$$

where the second term in the objective function provides a measure on how much the regression errors exceed the value ϵ.
Problem (8) is a quadratic convex problem in the variables $w, b, \xi, \hat{\xi}$, and therefore the solution can be found by solving its Wolfe dual problem, which is easier to be solved. Denoting by $\lambda^p, \hat{\lambda}^p, p = 1, \ldots, P$ the dual variables corresponding to the Lagrange multipliers associated with the constraints on the regression errors, and by $\lambda, \hat{\lambda}$ the vectors with components $\lambda^p, \hat{\lambda}^p,\ p = 1, \ldots, P$, the dual problem is obtained as:

$$\begin{array}{c} \min \Gamma(\lambda, \hat{\lambda}) = \frac{1}{2} \sum_{p=1}^{P} \sum_{q=1}^{P} (\hat{\lambda}^p - \lambda^p)(\hat{\lambda}^q - \lambda^q)(x^p)^T x^q \\ - \sum_{p=1}^{P} (\hat{\lambda}^p - \lambda^p) y^p + \epsilon \sum_{p=1}^{P} (\hat{\lambda}^p + \lambda^p) \\ \\ \sum_{p=1}^{P} (\hat{\lambda}^p - \lambda^p) = 0 \\ 0 \leq \lambda^p \leq C \qquad p = 1, \ldots, P \\ 0 \leq \hat{\lambda}^p \leq C \qquad p = 1, \ldots, P \end{array} \tag{9}$$

The structure of Problem (9) is of main interest, because it can be exploited for generalizing the linear SVM to the *nonlinear* SVM. To this aim it is sufficient to substitute the inner product $(x^p)^T x^q$ with the value $k(x^p, x^q)$ given by a suitable kernel function $k : \Re^n \times \Re^n \to \Re$.

Since we will make use of nonlinear SVM, the problem of concern becomes the following:

$$\begin{array}{c} \min \Gamma(\lambda, \hat{\lambda}) = \frac{1}{2} \sum_{p=1}^{P} \sum_{q=1}^{P} (\hat{\lambda}^p - \lambda^p)(\hat{\lambda}^q - \lambda^q) k(x^p, x^q) \\ - \sum_{p=1}^{P} (\hat{\lambda}^p - \lambda^p) y^p + \epsilon \sum_{p=1}^{P} (\hat{\lambda}^p + \lambda^p) \\ \\ \sum_{p=1}^{P} (\hat{\lambda}^p - \lambda^p) = 0 \\ 0 \leq \lambda^p \leq C \qquad p = 1, \ldots, P \\ 0 \leq \hat{\lambda}^p \leq C \qquad p = 1, \ldots, P, \end{array} \tag{10}$$

where we adopt, as kernel function, the commonly used Gaussian kernel:

$$k(x^p, x^q) = \exp(-\sigma \|x^p - x^q\|^2),$$

with $\sigma = 1/n$.
Problem (10) is a quadratic convex problem in the unknowns $\lambda, \hat{\lambda}$. Once solved, with solution $\lambda^*, \hat{\lambda}^*$, the nonlinear regression function $y(x)$ for the set of input-output samples $\mathcal{T}$ is given by

$$y(x) = \sum_{p=1}^{P} ((\hat{\lambda}^p)^* - (\lambda^p)^*) k(x, x^p) + b^*, \tag{11}$$

where b^* can be determined making use of the complementarity condition.
From (11) it appears that the SVM can be seen as a neural network of RBF, with P radial basis functions given by the kernels $k(x, x^p), p = 1, \ldots, P$ weighted by the coefficients $((\hat{\lambda}^p)^* - (\lambda^p)^*)$ and one dummy input equal to 1 weighted by b^*. Therefore the property of being an universal approximator of continuous functions on compact sets extends also to the SVM.

3 Implementation issues

Once the kind of a learning machine for forecasting has been chosen, its development requires the availability of:

- a data set,
- an optimization procedure,
- a validation procedure.

In this section we will shortly describe the three items.

3.1 Data set

The data set is the set of available input-output samples $\{(x^p, y^p),\ x^p \in \Re^n, y^p \in \Re,\ p = 1, \ldots, R\}$, where R is usually very large. It must be divided into three subsets:

- the training set $\mathcal{T} = \{(x^p, y^p),\ p = 1, \ldots, P\}$ used by the optimization procedure in the training phase,

- a validation set $\mathcal{V} = \{(x^p, y^p), p = P+1, \ldots, Q\}$ used for validating the machine as a tool for generalizing its forecasting ability also with respect to input-output pairs that are not in the training set,
- a test set $\mathcal{S} = \{(x^p, y^p),\ p = Q+1, \ldots, R\}$ used for measuring the quality of forecast produced by the resulting learning machine within the data set.

Let x^p an input value, y^p the corresponding output and $y(x^p)$ the value predicted by the machine after the learning and validation procedures. Then the test set $\mathcal{S}$ is used to compute the mean absolute error $MAPE(\mathcal{S})$ value:

$$MAPE(\mathcal{S}) = \frac{1}{(R-Q)} \sum_{Q+1}^{R} \frac{|y(x^p) - y^p|}{\max\{1, |y^p|\}}, \tag{12}$$

which provides an overall measure of the quality of the forecast. The term $\max\{1, |y^p|\}$ in the (12) is used in order to avoid that the error increases to infinity in case of zero sales for a single day of the prediction .

3.2 Optimization procedure

As shown before, the training of a learning machine turns out to be an optimization problem, with two significant features: the first one is that the problem is usually of very large scale, the second one is that for every machine it has a particular structure, so that the second feature may in some way balance the first one. Indeed very specialized, and therefore efficient, algorithms have been proposed for the training of learning machines, that take explicitly into account the problem structure, see for instance [10], [11], [17], [18], [20]. In our application we have used the Truncated Newton Method of [13] for the unconstrained optimization problems (1), (6) arising in the training of multilayer and RBF ANN, and the algorithm available through [6] for the constrained optimization problem (10) of SVM training.

3.3 Validation

In the optimization problem to be solved in the training procedure, the structure of a learning machine is given, as well as the values of the parameters that appear in the optimization model. For instance, in training a RBF ANN, when solving problem (6), the number N of neurons and the values of coefficients γ_1 and γ_2 are given. The validation procedure aims to determine the complexity of the learning machine and the values of the parameters that appears in the optimization model so as to obtain the best performances in forecasting.
Indeed, in training a learning machine, there is a tradeoff between the capacity of the machine to interpolate the training samples and its capacity to predict values that do not belong to the training set. Making again reference to a RBF ANN, if the number N is small, the machine could not be able to realize the function that links the input and the output; on the other hand, if N is large the phenomenon of overfitting may occur, that is the machine interpolates very well the training samples, but becomes inefficient on the samples of the test set, since it looses its generalization proprieties with respect to samples not in the training set.

The validation procedure is performed by using the $MAPE(\mathcal{V})$ value, defined in a way similar to the $MAPE(\mathcal{S})$, with reference to the validation set rather than to the test set:

$$MAPE(\mathcal{V}) = \frac{1}{(Q-P)} \sum_{P+1}^{Q} \frac{|y(x^p) - y^p|}{\max\{1, |y^p|\}}.$$

As concerns the neural networks considered before, both two-layers and RBF, we have to choose the number of neurons N. A simple methodology consists in computing the $MAPE(\mathcal{V})$ value in correspondence to increasing values of N. Usually we observe that, by increasing N, the $MAPE(\mathcal{V})$ value first decreases, and then begins to raise. This is the symptom that the network is beginning to overfit the training data, so that we stop the increase of N. In summary, we train a sequence of networks with increasing values of N, starting from a small value, until the $MAPE(\mathcal{V})$ value begins to rise.
In the optimization model of two-layer and RBF ANN, we have also the possibility of tuning the γ parameters. The optimal values of the γ parameters can be determined by using again a validation technique. Increasing γ helps the generalization capacity of the network because it puts a limitation on the choice of weights' values, that corresponds to making more regular the output function realized by the network. From an algorithmic point of view, choosing the value of γ high enough simplifies the optimization problem by convexifing the objective function.
As concerns the SVM, in the optimization model (8) we have to give values to the parameter ϵ that bounds the regression errors and to the parameter C which weights the fact that the regression errors exceed the value ϵ. Given ϵ, increasing C has similar effects as increasing γ, of making more regular the output function realized by the machine. However, if exaggerated, it produces the trouble of overfitting. The validation procedure for SVM is similar to that used in the ANN: for different values of ϵ we train a sequence of SVM increasing the value of C until the $MAPE(\mathcal{V})$ error begins to rise. Often the value $\epsilon = 0.1$ is adopted [6], and the validation procedure reduces to determine only the value of C.

4 Experimental environment

In this section we describe how the learning machines have been used for sales forecasting. In our application, we used two input-output time series, taken from two different retail stores of the same chain of large distribution. As concerns the output y we are interested in the daily sales of a particular kind of pasta of a popular brand; as concerns the input vector x, we will describe below which attributes have been taken into account. In particular we are interested in capturing the effects of promotion policies on the sales. The input-output samples used for training, validation and testing cover three years: 2007, 2008 and 2009. In particular, the years 2007 and 2008 have been used only for training and validation, the year 2009 for forecasting and testing. The first time series is taken from retail store #1, which is characterized by a bad storage management, so that a stockout occurs often. This brings the difficulty of not knowing whether an output sample is zero because there was no demand or because there was stockout. This series can be considered unreliable because of the presence of an high number of stockouts. Nevertheless we analyze this kind of data

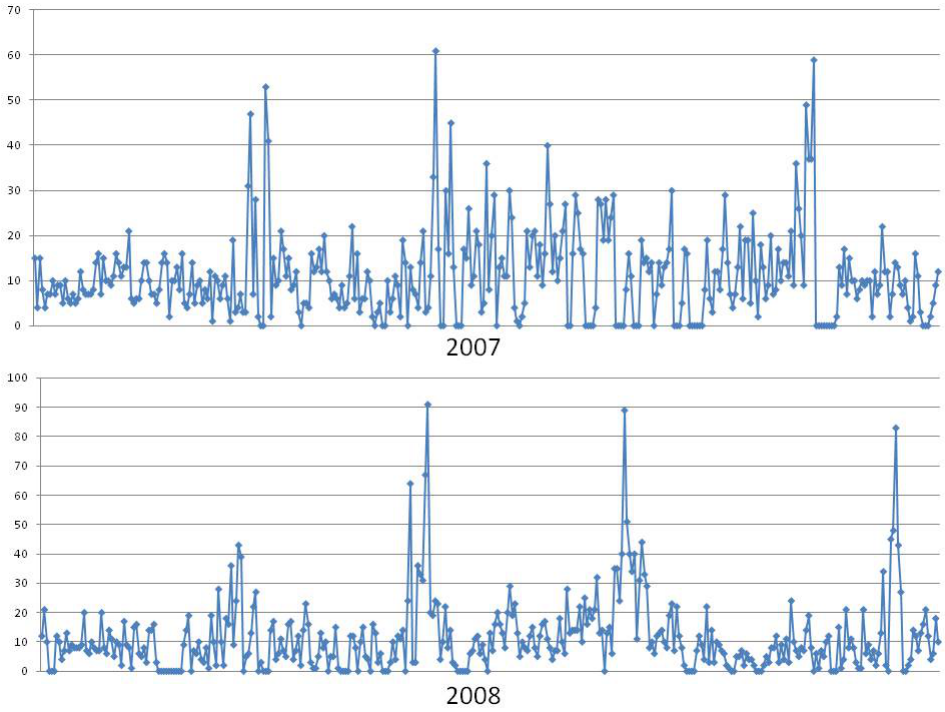

Figure 2: Sales of store #1 (2007-2008).

because we want to test the reliability of these forecasting tools on real datasets that are not in ideal conditions. The second series is taken from store #2, which has a good storage management, so that stockout happens rarely. The sales in the years 2007 and 2008 are drawn in Fig.2 and Fig. 3, respectively for store #1 and store #2. It appears clearly that sales increase during promotion periods, that have been 3 during 2007 and 4 during 2008.
In our forecasting we will use as input attributes subsets of the following set of 13 attributes:

- 9 calendar attributes, linked to the specific day in which the output is given: month, day of the month and day of the week. The day of the week is represented by 7 mutually exclusive boolean attributes. These attributes bring into consideration typical human behaviors and customs. For example in Saturday it is expected to sell more than in the other days of the week.

- 4 problem specific attributes: one boolean attribute whose value is one if there is promotion of the product in that day, zero otherwise, number of hours the store is open that day and the daily price of the product; moreover the overall number of receipts released that day in the store, which accounts for the overall volume of sales.

As concerns the last attribute listed before, that is the number of receipts released in the same day for which the forecast is done, we point out that of course its value is not known. Therefore we implemented a SVM for forecasting the number of receipts per day. This SVM used the 2007 series for training and the 2008 series, divided into two, for validation and testing. Then we used this SVM to produce a forecast of the number of receipts in the 2009.

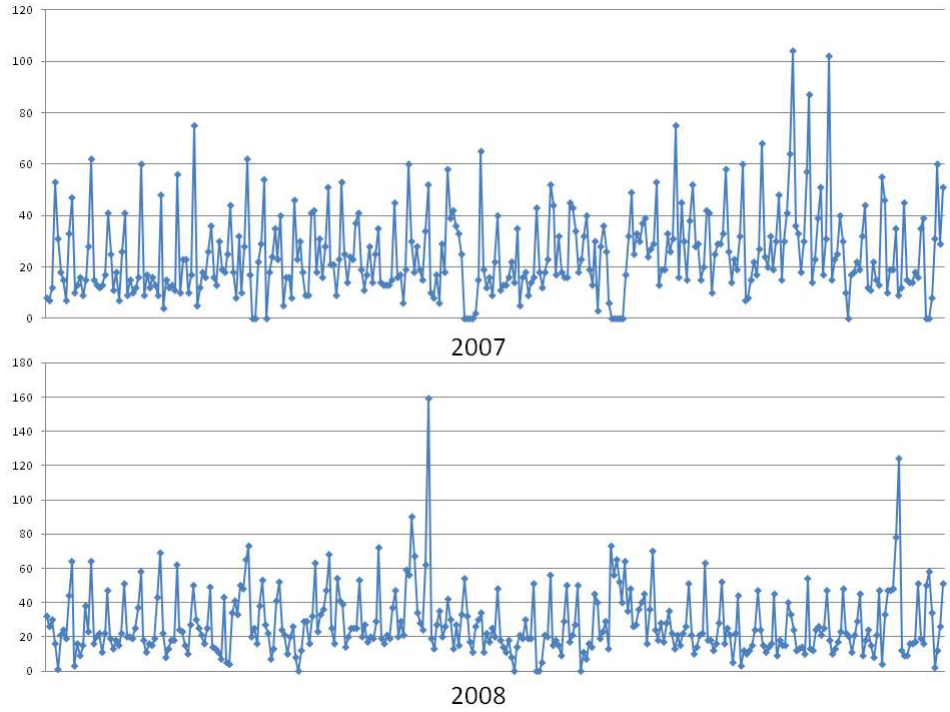

Figure 3: Sales of store #2 (2007-2008).

This SVM uses in input 11 attributes: the 9 calendar attributes also used in forecasting sales, the number of hours the store is open and a last attribute that indicates if in that day are expected high or low sales. This attribute is 0 in normal days, 1 in days before festivities, -1 when the store is open on Sundays and 2 on the day of Christmas eve and new year eve. A forecasted attribute can be considered a risky choice for the robustness of the final predictive model. However as we already said, we consider this attribute very important in the prediction and it also can be used in place of the calendar attributes in order to avoid the curse of dimensionality.
We realized several experiments changing the attributes in input:

- in the first experiment we use 4 inputs: promotion, number of opening hours, price of the product and number of daily receipts (forecast);
- in the second experiment we use 12 inputs: promotion, number of opening hours, price of the product and the nine calendar attributes;
- in the last experiment we use all 13 attributes listed before.

In the 4 inputs experiment we test the the goodness of final prediction with the forecasted attribute. In the 12 inputs experiment we test the goodness of the prediction with the calendar attributes, but without the forecasted number of receipts. With the final experiment with 13 attributes we test the goodness of the prediction with all the attributes together.

The forecasting is executed by adopting the *sliding window* method often used in this kind of applications. After eliminating the days in which the store was closed, we divided the year of test, the 2009, into 10 intervals of the same size. Store #1 has 10 intervals made of 36 days, because it was opened in the Sunday, while store #2 has 10 intervals made of 32 days because of the Sunday closure.
We make reference to the time series of store #1 in order to explain how we proceeded. The 10 sets of samples used for testing are denoted by $\mathcal{S}_i, i = 1, \ldots, 10$. The tenth interval of 36 input-output samples of the year 2008 is used first as validation set $\mathcal{V}$; the remaining samples of the year 2008 and the samples of the whole year 2007 are used first as training set $\mathcal{T}$.
For different learning machines, belonging to the three classes of Multilayer ANN, RBF ANN, SVM, we first perform the training procedure using the set $\mathcal{T}$ and the validation procedure using the set $\mathcal{V}$, so as to select the best performing learning machine in each class; then we use the selected machine to perform the forecast of the output samples in the test set $\mathcal{S}_1$, and we measure the quality of the forecast by the $MAPE(\mathcal{S}_1)$ value.
Then we add the set $\mathcal{V}$ to the training set and we take the set $\mathcal{S}_1$ as new validation set, in order to perform the forecast of the output samples in the set $\mathcal{S}_2$ and to measure the $MAPE(\mathcal{S}_2)$ value. The procedure is repeated, until we reach the last interval of the year 2009: in order to forecast the output samples in the set $\mathcal{S}_{10}$ we use as training set the samples of the whole years 2007 and 2008, as well as the samples in $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_8$, and we use as validation set the samples in $\mathcal{S}_9$.

The time series of store #2 has been treated in the same way, the only difference being in the number of samples in each interval, 32 instead of 36, due to a larger number of days in which the store was closed.

We use this method of prediction on the data series because we want to realize the prediction from the point of view of a practitioner who realizes a monthly prediction with the most updated data available.

5 Computational results

In this section we report the results obtained in forecasting the sales during 2009, making use of the different Learning Machines, and we make a comparison with the forecasts provided by traditional statistical methods. In particular, for each store we run 12 computations, 9 for the three different learning machines by using the three different configuration of input attributes, and 3 for statistical methods, the first method being ARIMA, the second one being the exponential smoothing (ES) and the third one being the Holt-Winter variation of exponential smoothing (HWES).

5.1 Forecast of daily receipts

Preliminarily we show the results obtained using a SVM for forecasting the number of daily receipts in 2009, used as input attribute. As already said, we used the samples of 2007 for training and the samples of 2008 for validation and testing, with the 11 input attributes listed in Section 4. In particular the validation was performed by taking $\epsilon = 0.1$ and adjusting the value C heuristically.

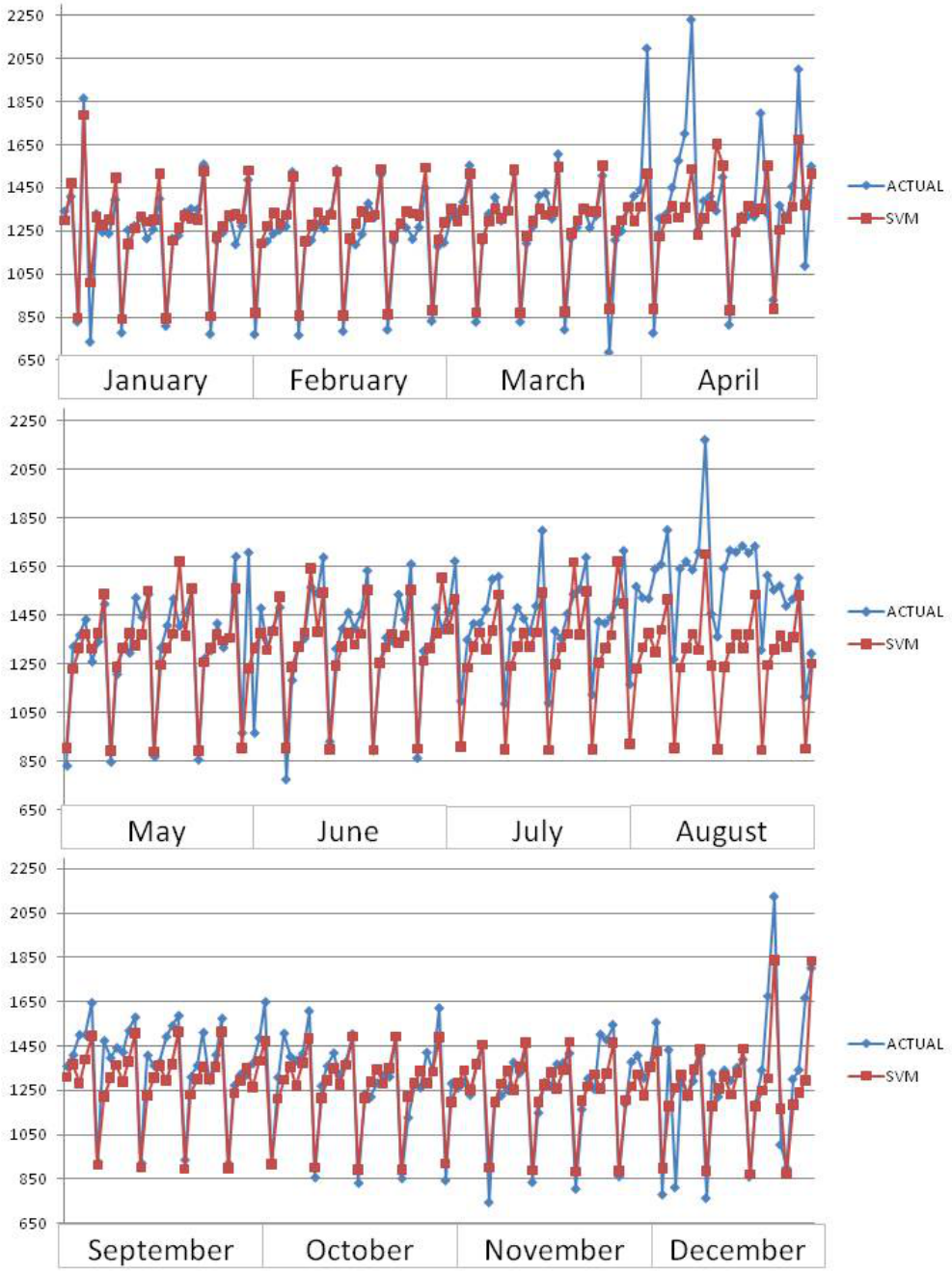

Figure 4: Daily receipts forecast for store #1 (2009).

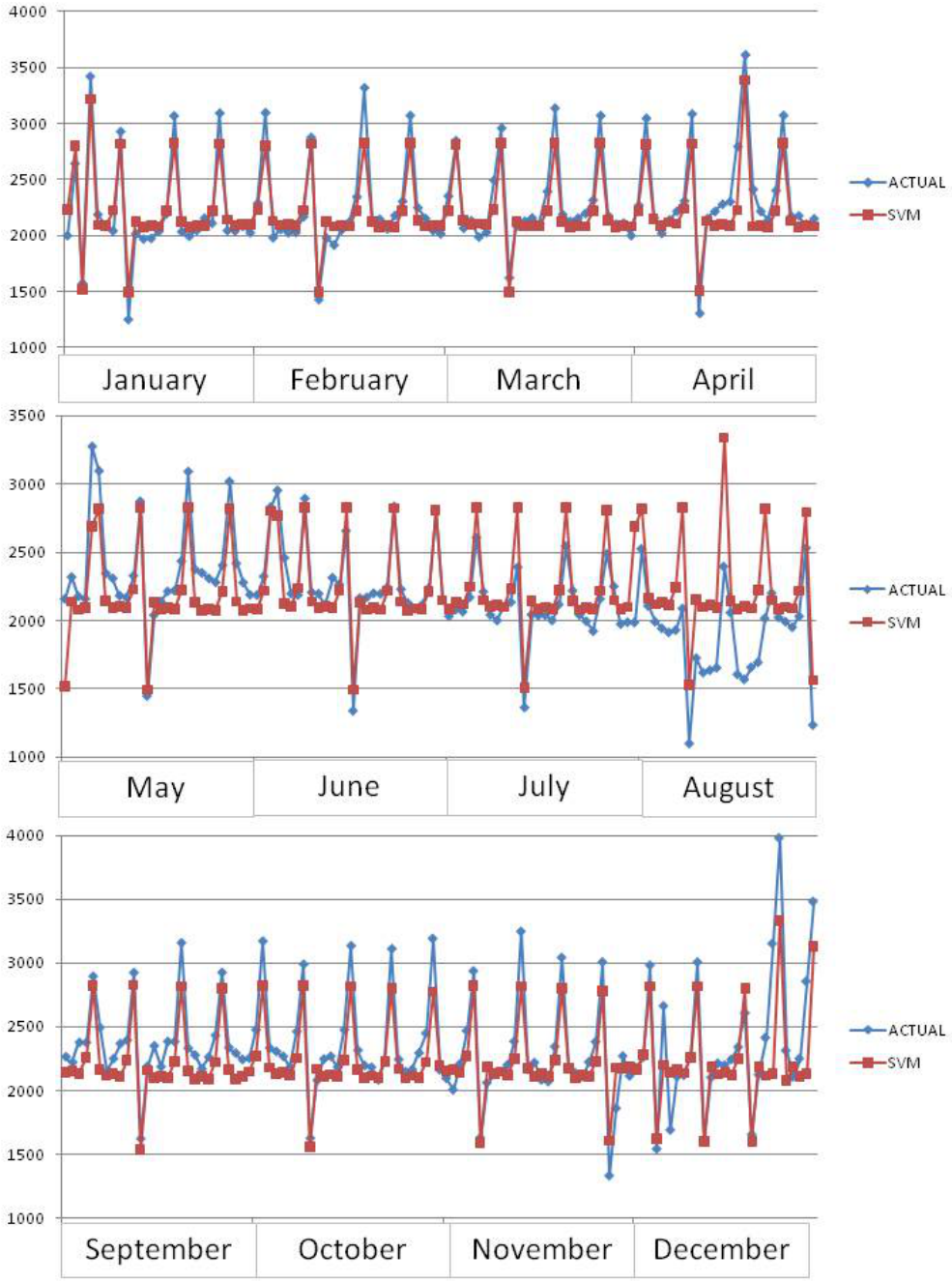

Figure 5: Daily receipts forecast for store #2.

We terminated the validation procedure for store #1 with $C = 3$ and for store #2 with $C = 1$. We draw in Fig. 4 and in Fig. 5 the actual and the forecast number of receipts in 2009, respectively for store #1 and store #2. The corresponding values of $MAPE(2009)$ are 0.47 for store #1 and 0.51 for store #2.
We can see from Figures 4 and 5 that the forecast of daily receipts produced by the SVM is quite satisfactory.

5.2 Sales forecast in store #1

The results obtained after training and validating the three kind of learning machines, each one with three different configuration of input attributes, denoted by $4i$, $12i$, $13i$ are given in terms of $MAPE(\mathcal{S}_i), i = 1, \ldots, 10$ in Table 1. In the same table are given the $MAPE(\mathcal{S}_i)$ values obtained using the three statistical methods.

Method	1	2	3	4	5	6	7	8	9	10	Mean
Mul.4i	1.86	0.52	0.80	0.88	1.31	0.54	0.71	0.78	0.54	0.78	0.87
Mul.12i	2.42	0.52	0.96	0.98	1.90	0.76	0.51	0.94	0.80	0.82	1.06
Mul.13i	2.80	0.54	0.58	1.22	1.68	0.68	0.54	0.76	0.71	0.84	1.03
RBF4i	1.71	0.52	0.80	0.86	1.40	0.52	0.62	0.67	0.53	0.48	0.81
RBF12i	1.90	0.62	0.85	0.82	1.83	0.59	0.54	0.69	0.62	0.58	0.90
RBF13i	3.56	0.60	0.95	0.80	1.94	0.60	0.47	0.74	0.55	0.37	1.06
SVM4i	2.14	0.56	0.88	0.90	1.46	0.51	0.66	0.55	0.49	0.54	0.87
SVM12i	2.10	0.50	0.87	0.71	1.63	0.50	0.53	0.62	0.51	0.51	0.85
SVM13i	2.39	0.52	0.82	0.75	1.58	0.51	0.53	0.58	0.51	0.47	0.87
ARIMA	2.26	0.62	0.91	1.19	2.32	0.51	0.50	0.83	0.84	0.71	1.07
ES	3.01	0.56	0.87	1.14	2.12	0.49	0.53	1.12	0.94	0.97	1.17
HWES	2.84	0.58	1.07	1.24	2.26	0.54	0.70	1.07	1.06	0.98	1.24

Table 1: $MAPE(\mathcal{S}_i)$ for store #1

From the table we get that, among the learning machines, the best mean performance is given by RBF4i, while among the statistical methods it is given by ARIMA. The values of N resulting by the validation phases of RBF4i are respectively, for the 10 intervals, $N = (19, 19, 35, 22, 10, 39, 12, 12, 43, 11)$; the values of γ_1, γ_2, after some preliminary tests, have been adjusted to the constant values $\gamma_1 = \gamma_2 = 0.01$.
We note that for every learning machine, good results are obtained with the 4 inputs configuration. We underline that the receipts attribute in the 4i computations brings with it information concerning the calendar attributes, and this is probably the reason for these good performances.
We recall that store #1 is characterized by a bad stock management, so that several stockouts occur, as can be seen from Fig. 2. This makes forecasting difficult, and explains the relative high values of the $MAPE(\mathcal{S}_i)$ entries in Table 1. However, in all cases the Learning Machines perform better than statistical methods, thus confirming that Learning Machines are more suited than statistical methods for forecasting series with irregular behavior.
In Fig. 6 we draw the actual sales in store #1 during 2009, and the forecasts produced by RBF4i and ARIMA. Promotion periods are displayed by vertical dashed lines. It appears that mainly in these periods RBF4i outperforms ARIMA.

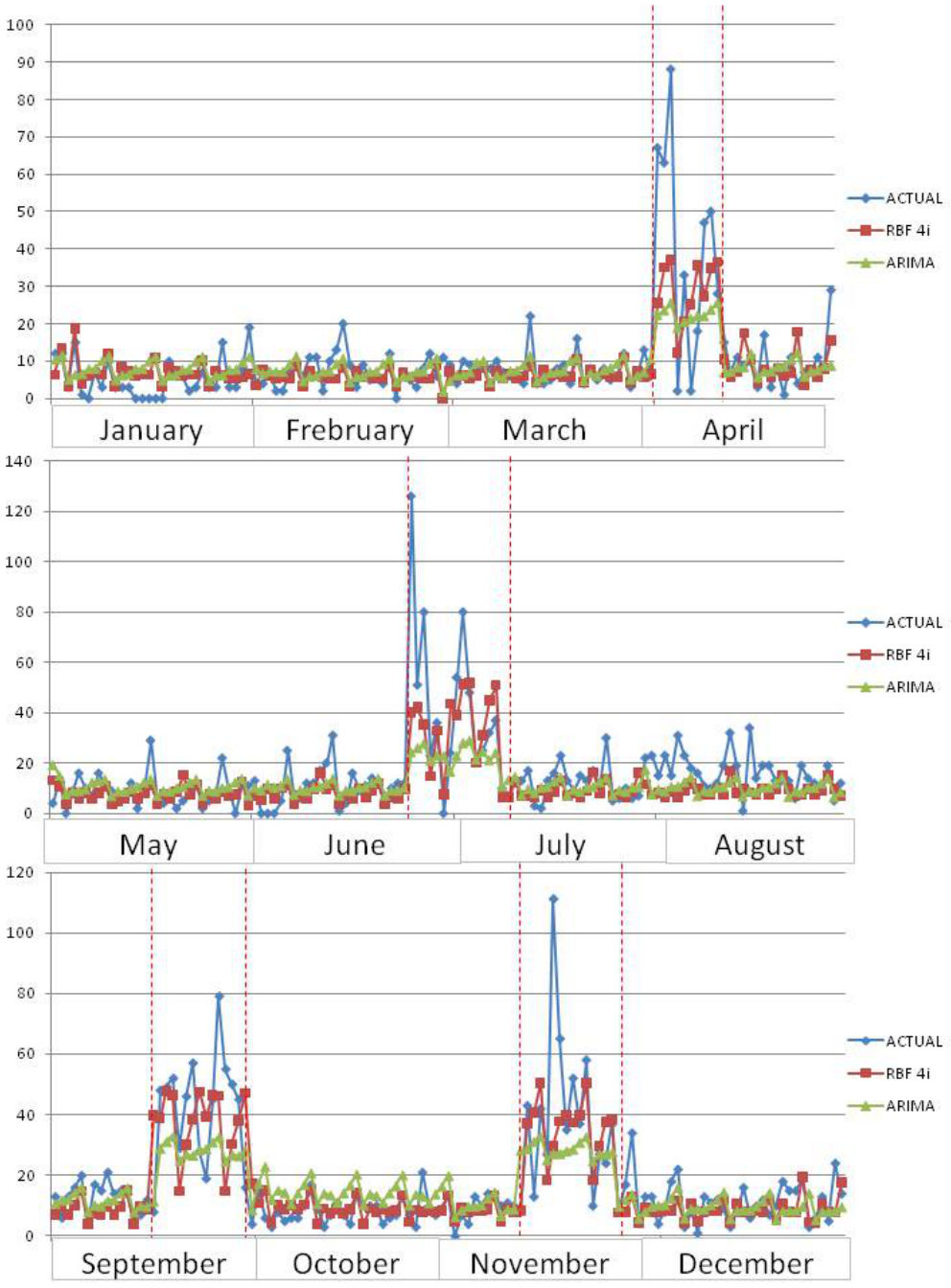

Figure 6: Actual and forecast sales in store #1 during 2009.

5.3 Sales forecast in store #2

In Table 2 we report the results obtained for store #2, organized in the same way as in Table 1. From the table it appears that, among the learning machines, the best results are given by SVM4i, and among the statistical methods by HWES. Assuming $\epsilon = 0.1$, the values of C resulting from the validation phases of SVM4i are respectively, for the 10 intervals, $C = (2, 2, 2, 1, 1, 1, 1, 38, 14, 1)$. In this case for every learning machine the best results are obtained with the 4 inputs configuration, thus confirming the effectiveness of the daily receipts attribute.

Method	1	2	3	4	5	6	7	8	9	10	Mean
Mul.4i	0.38	0.32	0.23	0.24	2.08	0.36	0.32	1.05	0.56	0.58	0.61
Mul.12i	1.45	0.38	0.51	0.33	2.06	0.35	0.39	1.54	0.72	0.59	0.83
Mul.13i	0.51	0.36	0.34	0.29	2.32	0.55	0.36	1.07	0.74	0.50	0.70
RBF4i	0.36	0.31	0.24	0.24	2.41	0.44	0.31	1.04	0.56	0.53	0.64
RBF12i	0.54	0.40	0.31	0.29	2.44	0.42	0.38	1.16	0.77	0.46	0.72
RBF13i	0.43	0.35	0.31	0.27	2.23	0.47	0.37	0.93	0.59	0.57	0.65
SVM4i	0.36	0.31	0.23	0.26	1.96	0.41	0.32	1.01	0.57	0.63	0.61
SVM12i	0.66	0.31	0.26	0.24	2.03	0.42	0.34	0.99	0.70	0.61	0.66
SVM13i	0.53	0.29	0.26	0.23	1.97	0.44	0.33	1.06	0.64	0.60	0.64
ARIMA	0.45	0.37	0.25	0.25	2.01	0.48	0.36	1.18	0.62	0.61	0.67
ES	0.44	0.33	0.26	0.27	1.99	0.46	0.36	1.24	0.66	0.66	0.67
HWES	0.46	0.33	0.27	0.27	1.97	0.43	0.33	1.21	0.69	0.65	0.66

Table 2: $MAPE(\mathcal{S}_i)$ for store #2

We note that the mean values in the last column of Table 2 are significantly smaller than the mean values in Table 1. This is probably due to the fact that stockouts in store #2 are much less frequent that in store #1, so that the series of sales results more regular. For the same reason in case of store #2 the statistical methods compare better with the learning machines than in the case of store #1.
In Fig. (7) we draw the actual sales in store #2 during 2009, and the forecasts produced by SVM4i and HWES. Promotion periods are again displayed by vertical dashed lines, and again it appears that in these periods SVM4i outperforms HWES.

We conclude this section by pointing out that, looking at the results of the 20 tests reported in Tables 1 and 2, we see that both SVM and RBF perform better 7 times on 20, the 35% of the total, that Multilayer ANN performs better 5 times on 20, the 25% of the total, while the statistical methods are better than the others only once on 20, the 5% of the total.

6 Conclusions

We have described how Learning Machines can be applied to sales forecasting, making also a comparison among them. The application has concerned the daily sales of a kind of pasta in two retail stores, in the presence of promotion. We have pointed out the importance of a suitable selection of input attributes for the machines. From the computational results we have shown that Learning Machines provide a valuable tool for sales forecasting, even if it does not appear that one kind of machine is definitively superior to the others. As a conclusion, we claim that any sales manager could take advantage by enlarging the class of

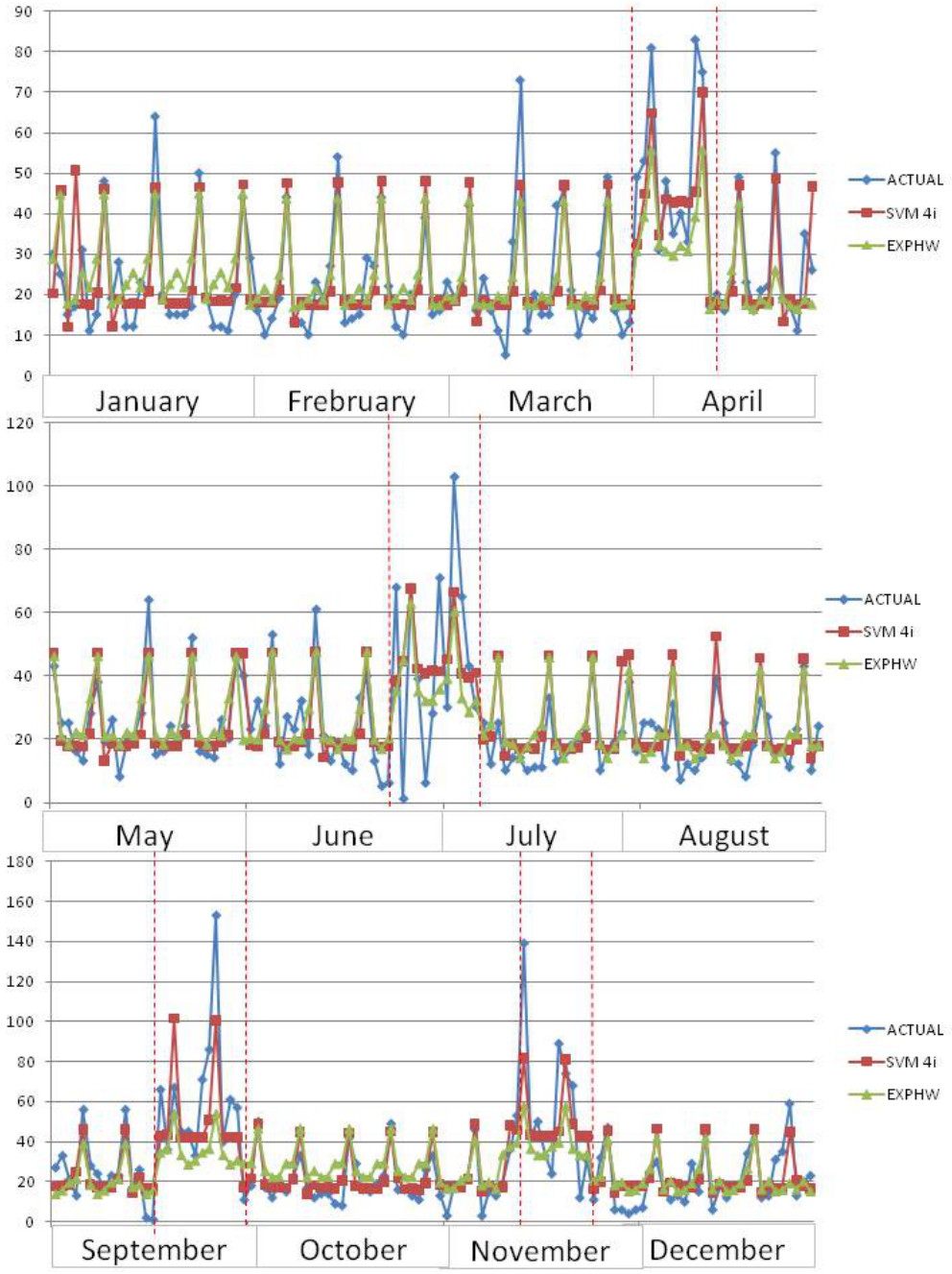

Figure 7: Actual and forecast sales in store #2 during 2009.

methods employed for sales forecasting, so as to include, with the more traditional statistical methods, also the Learning Machines described in this paper.

7 References

References

[1] I. Alon, M Qi, R. J. Sadowski, *Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods*, Retailing and Consumer Services, **8**, 147–156, 2001.

[2] A. P. Ansuj, M. E. Camargo, R. Radharamanan, D. G. Petry, *Sales forecasting using time series and neural networks*, Computers Industrial Engineering, **31**, 421–424, 1996.

[3] Anthony M, P.L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambrige University Press, 1999.

[4] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[5] C.J.C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Mining and Knowledge Discovery, **2**, 121–167, 1998.

[6] C.-C. Chang, C.-J. Lin, *LIBSVM: a library for support vector machines*, Software available at http://www.csie.ntu.edu.tw/˜ cjlin/libsvm, 2001.

[7] P. C. Chang, C. H. Liu, C. Y. Fan, *Data clustering and fuzzy neural network for sales forecasting: a case study in printed circuit board industry*, Knowledge-Based Systems, **22**, 344–355, 2009

[8] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*, Cambridge Univerity Press, 2000.

[9] D. Cui, D. Curry, *Prediction in marketing using the support vector machine*, Marketing Science, **24**, 595–615, 2005.

[10] R.-E. Fan, P.-H. Chen, C.-J. Lin, *Working set selection using second order information for training SVM*, Journal of Machine Learning Research 6, 1889-1918, 2005.

[11] S. Haykin, *Neural Networks, a Comprehensive Foundation*, Prentice-Hall, 1999.

[12] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference and Prediction*, Springer, 2001.

[13] G. Fasano, S Lucidi, *A nonmonotone trucated Newton-Krylov method exploiting negative curtature directions, for large scale uncostrained optimization*, Optimization Letters, **3**, 521–535, 2009.

[14] R.J. Kuo, T. L. Hu, Z. Y. Chen , *Application of radial basis function neural network for sales forecasting*, Inter. Asia Conf. on Informatics in Control, Automation and Robotics, 325–328, 2009.

[15] R.J. Kuo, K.C. Xue, *A decision support system for sales forecasting through fuzzy neural networks with assymetric fuzzy weights*, Decision Support Systems, **24**, 105–126, 1998.

[16] A. A. Levis, L. G. Papageorgiou, *Customer demand forecasting via support vector regression analysis*, Chemical Engineering and Design, **83**, 1009–1018, 2005.

[17] C.-J. Lin, *On the convergence of the decomposition method for Support Vector Machines*, IEEE Trans. on Neural Networks, **12**, 1288–1298, 2001.

[18] S. Lucidi, L. Palagi, A. Risi, M. Sciandrone, *A convergent hybrid decomposition algorithm model for SVM training.* IEEE Trans. on Neural Networks, **20**, 1055–1060, 2009.

[19] S. Makridakis, S.C. Wheelwright, R.J. Hyndman, *Forecasting: Methods and Applications*, John Wiley and Sons, 1998.

[20] T. Serafini, L. Zanni, *Parallel software for training large scale support vector machines on multiprocessor systems*, Journal of Machine Learning Research, **7**, 1467–1492, 2006.

[21] B. Schlkopf, A. Smola, *Learning with Kernels, Support Vector Machines, Regularization, Optimization and beyond*, The MIT Press, 2002.

[22] F.M. Thiesing, U. Middelberg, O. Vornberger, *Short term prediction of sales in supermarkets*, IEEE Inter. Conf. on Neural Network Proceedings, **2**, 1028–1031, 1995.

[23] F.M. Thiesing, O. Vornberger, *Sales forecasting using neural networks*, IEEE Inter. Conf. on Neural Network Proceedings, **4**, 2125–2128, 1997.

[24] V. N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, 1998.

[25] P. M. West, P. L. Brockett, L. L. Golden, *A comparative analysis of neural networks and statistical methods for predicting consumer choice*, Marketing Science, **16**, 370–391, 1997.

[26] H. White, *Artificial Neural networks*, Blackwell, 1992.

[27] Q. Wu, H. S. Yan, H. B. Yang , *A forecasting model based on support vector machine and particle swarm optimization*, Workshop on Power Electronics and Intelligent Transportation System, PEITS'08 Proceedings, 218–222, 2008.

[28] G. Zhang, B. E. Patuwo, M. Y. Hu, *Forecasting with artificial neural networks: the state of the art*, International Journal of Forecasting, **14** 35–62, 1997.