Journal of the Operational Research Society (1999) 50, 85–94 0160-5682/99 $12.00

http://www.stockton-press.co.uk/jor

A hybrid model based on dynamic programming, neural networks, and surrogate value for inventory optimisation applications

CC Reyes-Aldasoro[1], AR Ganguly[2], G Lemus[2] and A Gupta[2]

[1]*Instituto Tecnológico Autónomo de México,* [2]*Massachusetts Institute of Technology*

This paper proposes a new approach to minimise inventory levels and their associated costs within large geographically dispersed organisations. For such organisations, attaining a high degree of agility is becoming increasingly important. Linear regression-based tools have traditionally been employed to assist human experts in inventory optimisation; endeavours; recently, Neural Network (NN) techniques have been proposed for this domain. The objective of this paper is to create a hybrid framework that can be utilised for analysis, modelling and forecasting purposes. This framework combines two existing approaches and introduces a new associated cost parameter that serves as a surrogate for customer satisfaction. The use of this hybrid framework is described using a running example related to a large geographically dispersed organisation.

Keywords: data mining; dynamic programming; inventory optimisation; neural networks

Introduction

The need for forecasting future conditions is important in a number of disciplines. For inventory optimisation applications, the need for forecasting becomes very important if the goods are non-durable, if the supply lead time is significantly greater than the demand lead time, if holding and ordering costs are large, and if a high degree of customer satisfaction is to be ensured[1–10].

The research described in this paper was originally motivated by data provided by a retail distribution company that sells goods to organisations and individuals via a network of warehouses and stores located across the United States. The organisation is called ‘Retailcorp’ in this paper. The nature of the items distributed by Retailcorp has a low ticket characteristic in general. This condition means that customers are expecting to find an item when they visit the store and are not willing to wait for it, which will result in lost sales for Retailcorp if the item cannot be found. The notion of partial lost sales as discussed by Nahmias and Smith[7] is not considered in this paper.

When this research was initiated, Retailcorp carried an aggregate inventory of one billion dollars on a running basis. Apart from the huge storage and allied inventory carrying costs, Retailcorp was incurring significant costs from discarding items which remained unsold on their respective expiration dates. The corporate philosophy stipulated a service level, or fill rate, of 95%, which means that a customer should find his/her product of interest on at least 95% of his/her visits to a Retailcorp store. The probability for a random item to be available at a random store, on any random day must be equal to or exceed 95%.

Previous efforts at addressing the above problem have concentrated on finding patterns in the data provided by the company’s ‘data warehouses’, and forecasting the sales levels of different items on the basis of historical data. This paper, however, adopts a slightly different approach. First, it focuses on a Dynamic Programming (DP) methodology to derive a state equation, and introduces a cost function that penalizes higher inventory subject to a service level constraint. Second, the process of training a neural network to help determine the optimal inventory levels is described.

This paper is organised as follows: First, the problem is discussed from two different aspects: a Dynamic Programming (DP) and a corresponding Neural Network (NN) approach. Second, the results obtained from an initial analysis of the data are highlighted: the sales data do not possess a normal distribution and the demand for individual items changes significantly from year to year. Third, a solution is obtained using a traditional model and another solution is obtained using a feed-forward Neural Network. Finally, the results obtained with a hybrid framework are presented.

Definition of problem

Dynamic programming definition

The nature of the inventory problem facilitates the use of Dynamic Programming techniques. The use of DP for inventory optimisation, especially when the demand is not constant over the planning horizon, has been analysed

Correspondence: Dr A Gupta, MIT Room E53-311, Cambridge, MA 02139, USA.
E-mail: agupta@mit.edu

by other researchers.[2–4] The situation varies from day to day, and depending on the conditions of that day, a certain decision has to be taken. The sequential decision-making process inherent to DP models offers several alternative solutions, from which an optimal solution can be chosen based on attaining the maxima or the minima of a certain variable. In our case, the goal was to seek the minima for the inventory. Traditional inventory and forecasting methods have been compared in other studies.[1,2,4]

The first step in solving the above problem is to construct a dynamic system (Figure 1) with the following state equation[11]:

$$x_{k+1} = x_k + u_k - w_k \tag{1}$$

where: x_k is the inventory available at the beginning of period k; u_k is the inventory ordered at period k; and w_k is the demand during the kth period.

The Retailcorp situation has two significant differences from the problem defined in Berstekas.[11] First, based on the commodities carried by Retailcorp, in a situation where it loses sales, no Retailcorp outlet can have a negative, or backlogged, demand. If a store runs out of a certain product, it is assumed that the unsatisfied customer will then go to another store (possibly a competitor), and may also return to the other store in the future. Therefore, unmet demand or lost sales not only imply lower profits, but could result in the permanent loss of customers. The 95% item fill rate level was established to avoid this situation. As such, the case of an isolated community with only one store where backlogged demand could easily occur is not considered in our running example.

The second difference is that there is a time lag between the placement of an order by the store and the delivery of the goods to that store, that is, the replenishment lead time is not zero. While customers can buy from the store on any day of the week at any time, the store is permitted to place orders only on a weekly basis, and to receive the ordered items a few days later. This supply lead time of more than a week is considered in our model.

To attain the desired level of customer satisfaction, Retailcorp utilised a policy of 'three weeks of supply' in which the average weekly demand for each item was computed using the last two years' data, and each store was expected to carry inventory equivalent to three weeks

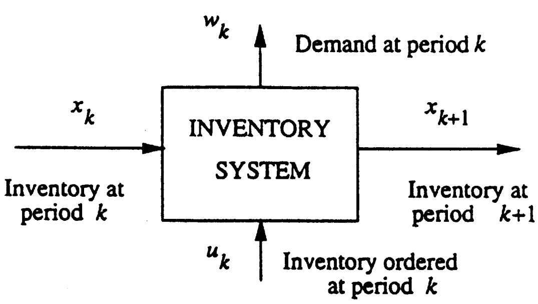

Figure 1 Inventory system.

of 'average demand' for that item. Several problems in this policy have been studied by Bansal *et al.*[12] First, the demand for slow and fast moving items are modeled and forecast in exactly the same fashion. Second, no special consideration is given to items characterised by abrupt or seasonal demand (as compared to ones with a flat rate of demand). Third, some kind of items require special handling, which may imply a higher cost and even a shorter life time in some cases. Based on these considerations, the formulation of a better policy was deemed necessary.

Equation (1) can be expanded to include an additional term to incorporate the constraint of 95% customer satisfaction. This parameter can be stated in terms of α (which in our case is 0.95) and is squared to avoid both overstocking and understocking. We then propose the minimisation of:

$$\min_{u_k} E\{x_{k+1}\} + c[P(x_{k+1} > 0) - \alpha]^2 \tag{2}$$

where c is a constant of proportionality. In classical Dynamic Programming,[11] a minimisation of $E\{cu_k + p\max(0, -x_{k+1}) + h\max(0, x_{k+1})\}$ would be required. However, in the case of Retailcorp, the costs associated for negative and positive inventory were not available, and emphasis was given to the total inventory reduction and the customer satisfaction. The minima is computed by the expected value because the demand over a certain period is supposed to be a random variable with a certain probability distribution, and is analysed later in this paper. The inventory equation can be expressed in terms of a cost function, $g(u_k)$:

$$g(u_k) = E\{x_k + u_k - w_k\} + c[P(x_{k+1} > 0) - \alpha]^2 \tag{3}$$

$$g(u_k) = x_k + u_k - E\{w_k\} + c[P(x_k + u_k > w_k) - \alpha]^2 \tag{4}$$

since x_k and u_k are not random variables. The computation of $P(x_{k+1} > 0)$ is done numerically using order statistics.[13] When $g(u_k)$ is minimised over u_k, x_k is independent of u_k, and u_k will be a monotonically increasing function. Therefore, the minimum will depend on the last two terms of (4). Experimental data from Retailcorp were used in order to evaluate $g(u_k)$ since the expected values of demand and probability were required. One can visualise three possible outcomes as follows:

$$u_k > c[P(x_{k+1} > 0) - \alpha]^2 \min g(u_k) \rightarrow \min u_k$$

$$u_k \simeq c[P(x_{k+1} > 0) - \alpha]^2 \min g(u_k) = f(c)$$

$$u_k < c[P(x_{k+1} > 0) - \alpha]^2 \min g(u_k) \rightarrow \min c[P(x_{k+1} > 0) - \alpha]^2$$

The three outcomes are shown graphically in Figure 2. One should note that without the 95% constraint, the last term of the equation will not exist; in such a scenario, the solution will be the one stated in the first condition where c is small, and then the minimum of $g(u_k)$ will tend to the minimum of u_k; in other words, the inventory supply should be the same

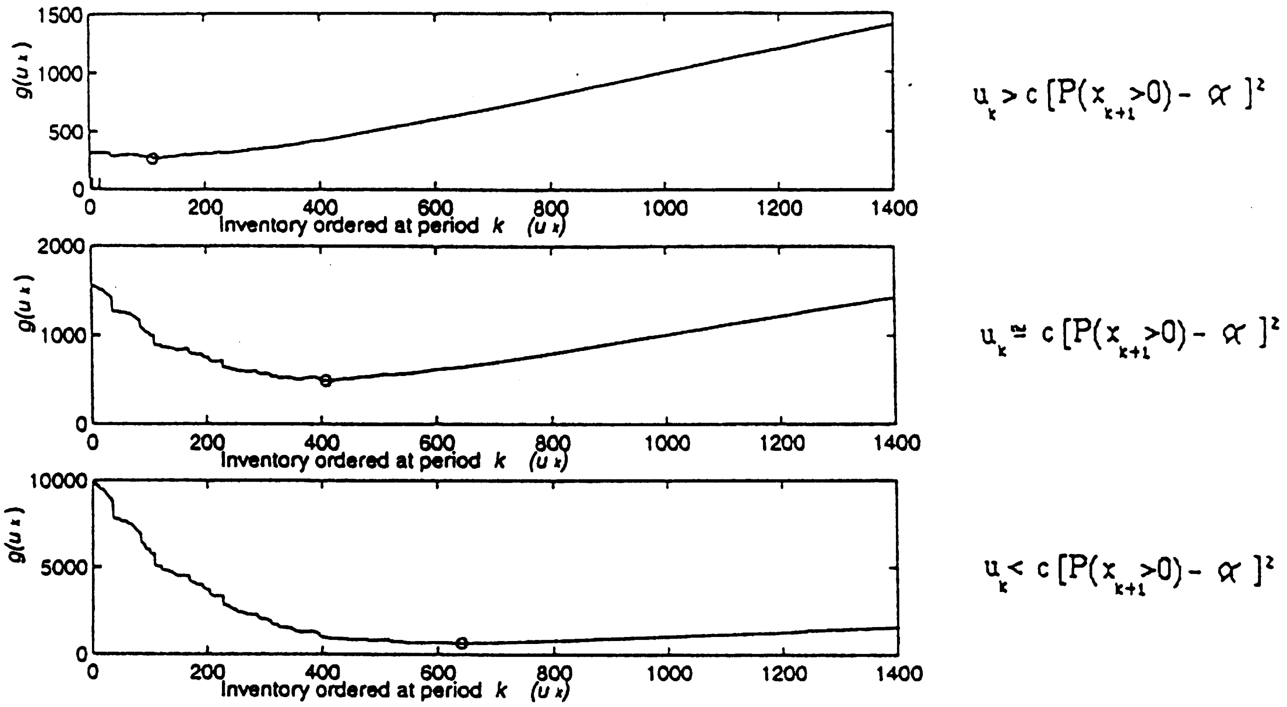

Figure 2 Minimising a cost function $g(u_k)$ to optimise inventory level for required customer satisfaction.

as the sales demand, and therefore, the accumulated inventory should be zero.

The process of predicting the exact demand from a time series has been discussed in a number of previous papers.[14–17] We opted to use a threshold approach, where the delineation of the threshold, rather than the analysis of the sales data, is the real objective. In this way, one avoids having to deal with daily variance, and can also adopt the weekly ordering concept utilised by Retailcorp. The influence of the 95% constraint depends on the constant *c*, which in turn defines the minimum value for the equation $g(u_k)$. The role of *c* is then to balance two important factors: the level of customer satisfaction (large *c*) and the need for holding minimum inventory levels (small *c*). The cost function defined here is simple, but serves to illustrate the value of these techniques as opposed to time series forecasting of raw sales data. More complicated cost functions[2–4] could be handled in a similar fashion.

Definition of neural networks

Neural Networks (NNs) are a class of input-output models capable of learning through a process of trial and error, and collectively constitute a particular class of nonlinear parametric models where learning corresponds to statistical estimation of model parameters. In the literature on NNs, a number of architectures, approaches and applications have been proposed.[18–22]

A common learning algorithm for feed-forward Neural Networks is error backpropagation.[22] The network is given a series of input-output pairs from which it tries to adapt its output to a certain target value by changing the weight and bias values inside the network. The process of calculating new weights and bias values is repeated until a certain error condition is satisfied; once this happens, the 'training' phase is terminated. The subsequent 'test' phase uses the values of weights and bias values that were generated during the training phase.

A common application of NNs relates to problems of forecasting and prediction of time series.[12,15–17,23] The general autoregressive time series forecasting case assumes that the next value in a time series will depend upon the previous data in the manner: $S_{n+1} = \mathscr{F}(S_n, S_{n-1}, \ldots)$, where $\mathscr{F}$ could be a nonlinear function. It should be noted that no causal factors such as sales or promotions were considered.[24] Classical techniques to forecast values of S_i assume that the function $\mathscr{F}$ is known, so that the problem becomes one of estimation of its parameters (for example, an assumption could be that the time series follows a linear growth, in which case linear regression technique suffices to provide the formula $\mathscr{F} = mS_i + b$).

Feed-forward Neural Networks and other kind of NNs have been proposed as possible solutions for this application area.[23,25,26] While ordinary feed-forward NNs are used to handle the nonlinear autoregressive component only, recurrent NNs are often used to model the nonlinear moving average component also.[15,17,26,27] Specifically, NNs are used to learn the correct form of $\mathscr{F}$ without having to assume any specific function. This approach offers two advantages: one does not need to know in advance the dynamics of the time series; and a NN will learn complex and highly nonlinear patterns (even chaotic functions[17]). The drawback is the amount of data required for the training of the NN.

The implementation of the above network was done in MATLAB[21] using a feed-forward network with back-propagation algorithm. Several configurations were evaluated. A typical network is presented in Figure 3 and is comprised of a network with an input layer of five neurons characterised by a pure linear function, a hidden layer with three neurons characterised by sigmoid functions, and a single output neuron characterised by a pure linear function.

Analysis of data

In a previous example,[12] the database from Retailcorp was used and analysed to predict future demand levels. One of the models proposed, the flat sales model, relies on two assumptions: that sales curves are always normal, and that sales data for one year match ones for the next year. We decided to test these two assumptions.

Figure 4 shows the histograms for the sales of one particular item at a number of stores. The graphs in the upper and lower subsets of the figure correspond to daily sales figures for two consecutive years: 1995 and 1996. The mean and the threshold (95% of the maximum) are shown. These figures show that the distribution is far from normal. Even if the data are analysed on a weekly basis using a moving average filter,[27] the hypothesis of normality is arguable, since a Poisson distribution, as proposed by Schwartz *et al*[9] or Negative Binomial distribution as proposed by Agrawal and Smith[5] would yield similar histograms. Not only do the mean sales values vary significantly from year to year, the histograms also show that the shape of the distribution is different for each year. In other words, the demand of a certain product varies not only in terms of quantity (different mean), but also in terms of frequency (histogram shape).

From this initial analysis of the data, we thought it necessary to develop a better framework which could

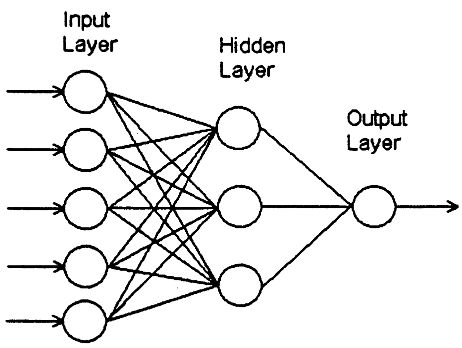

Figure 3 Neural network architecture (as implemented in MATLAB).

adequately address the random behaviour of the data. The use of a fixed distribution was discarded and a per-item analysis was performed using daily data from Retailcorp. This gave us some flexibility for modeling fast- and slow-moving items.

Traditional solution: sales threshold

In order to use (1) to minimise the inventory over time, a proper sequence for u_k needed to be found. The demand w_k is not known and the inventory x_k is the result of the previous time period. As such, the amount of inventory ordered for the period is the only variable which can lead to a solution to the equation. In Bertsekas,[11] a mapping function $u_k(x_k)$ that gives a proper u_k is introduced, and the optimal ordering rule for the inventory system is expressed as:

$$u_k(x_k) = \begin{cases} S_k - x_k, & x_k < S_k \\ 0, & x_k \geqslant S_k \end{cases} \tag{5}$$

where S_k is a proper threshold level. S_k can be either fixed, like the 'three weeks of supply' policy, or adaptable, that is:

$$S_{k+1} = \begin{cases} S_k \\ f_{(x_k)} \end{cases} \tag{6}$$

The first step in minimising the inventory levels for Retailcorp is to find an optimum threshold level *per item,* instead of using a flat 'three weeks' concept for all items. Accordingly, for each item, the sales data for 1995 were utilised to obtain the proper threshold values using order statistics.

Because of the time variant nature of the data for 1996, the use of a static threshold level for the whole year fails to comply with the restriction of 95% customer satisfaction. As such, the threshold level was used as a starting value for the first day of 1996, and an adaptation rule was used for later dates in that year.

Figure 5 shows the behaviour of the inventory level over time using a variable threshold. At Retailcorp, while the inventory level can decrease on a daily basis, it can increase only once every seven days. The circles in the upper part of the figure represent the level of the threshold for the day of supply. The asterix over the x axis represents each day that the inventory is exhausted, that is a day when the 95% fill rate was not satisfied. These days are termed as 'undershoots'.

Based on the DP system proposed, an empirical method to obtain the threshold can be derived. The threshold can be obtained through a periodic calculation that can adapt to the present conditions. The calculation is done by taking into account the past values of sales demand, as well as the values for the week that just ended. The question now is, how many past days should be considered while calculating the new threshold? This problem has certain similarities to

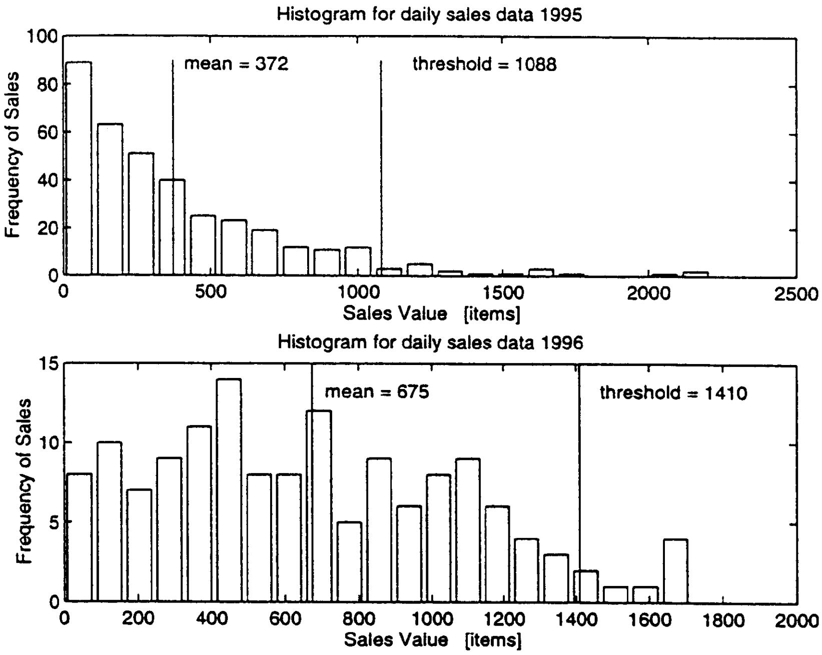

Figure 4 Histograms for sales in consecutive years.

the typical flow control problem, and flow control analysis techniques can therefore be used. A sliding window procedure[28] can be used, taking a fixed number of days to compute the future threshold. With the small size of the window, the calculated threshold is usually not very accurate, since a slight variation of the demand provokes a variation of the threshold and causes the behaviour of this threshold to be highly variable. As the window grows bigger, the threshold becomes increasingly stable. Above the 95% value for the customer satisfaction, the threshold is very stable. This implies an almost fixed value for the threshold with variations in the current demand having only minor impact on the threshold value.

An analysis of Figure 5 suggests several ideas for minimising the total amount of inventory: weights can be assigned to the previous days in order to give more

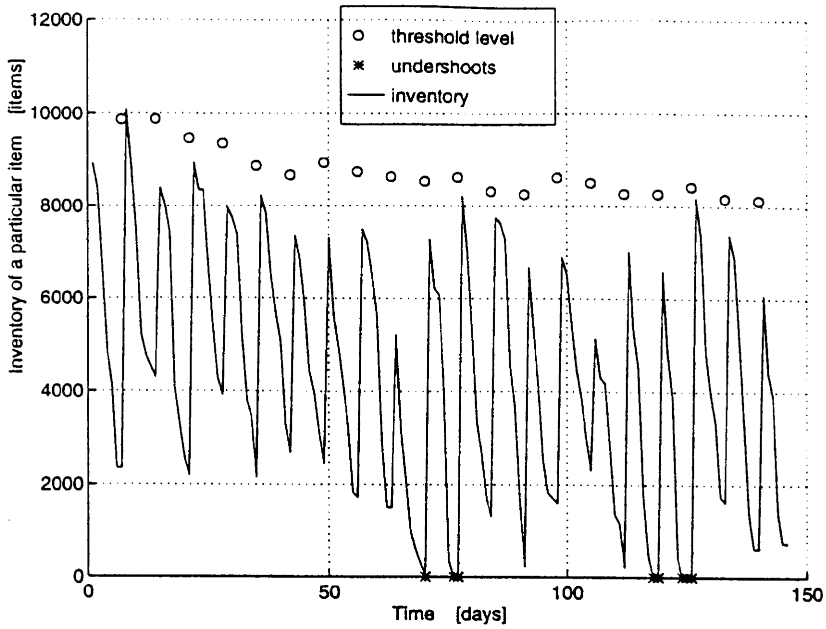

Figure 5 Behaviour of inventory over time.

importance to recent events and less importance to earlier ones; previous threshold calculations can be averaged with the new calculated value to avoid oscillations caused by sudden changes in the sales; and a penalty can be introduced to the new threshold if there is either a positive inventory level or an excessive number of undershoots occurred during the last week. Several of these ideas were tested and the resulting threshold values and inventory levels were analysed. In most cases, the inventory levels were reduced and the customer satisfaction level was still maintained above 95%. However, there was no rule for setting the new parameters for different sales patterns; furthermore, no generalised solution could be identified.

A NN approach, which provided better results for all kinds of input data, is described in the next section.

Neural network solution

Using the data from Retailcorp, a NN was trained to follow the threshold levels with the objective of forecasting the new value. The architecture of network used was a feed-forward Neural Network with nine input neurons, five hidden neurons and one output neuron. The standard back-propagation training algorithm turned out to be too slow, so two modified optimisation methods for backpropagation were used.[21] First, a *momentum* term was added to the weight update rule. This momentum term decreases the sensitivity of the algorithm to small details, so that if a local minima is located next to a global minima, the algorithm can reach the proper lowest value. Second, an *adaptive learning rate* was applied. This adaptive learning process tries to keep the learning pace as fast as possible while retaining stability. With the optimal backpropagation algorithm, several test data were used for training the network, testing its capability to approximate a certain pattern, and predicting new values.

The thresholds for every week serve as input data for the network, for both training and testing purposes. These data are received at the input layer of the Neural Network Artchitecture shown in Figure 3. The output layer presents the threshold predicted value. A simulation model with a fixed threshold, similar to the ones described in the previous section, was used. For each time-period of supply, the lack or surplus of inventory was calculated and the previous threshold was adjusted accordingly. This process was continued until the required threshold data set was created.

Two different prediction scenarios were evaluated: *multiple step prediction* and *single step prediction.* The multiple step prediction (msp) technique takes into account the data from the entire training period (namely, sales data from 1995) in order to predict the threshold levels for the test period (namely, 1996). The single step prediction (ssp) technique focuses on the prediction of the next value in the series only. The predicted value was then compared with the corresponding target value to see the difference between the two values and this information was then used to enhance the accuracy of the previous predicted value. The next value was then predicted using the corrected point in the series. This resulted in accurate results with the test data.

To precisely evaluate the performance of the network, a quantitative parameter was calculated. A common measure is the factor ε/σ where ε is the root mean square error of the test set, and σ is the standard deviation of the test set. This factor is calculated using the following equations[29]:

$$\epsilon = \sqrt{\sum_{\alpha=1}^{k} (x_\alpha - \hat{x}_\alpha)^2} \tag{7}$$

$$\sigma = \sqrt{\sum_{\alpha=1}^{k} (x_\alpha - \bar{x}_\alpha)^2} \tag{8}$$

where: x_α is the α^{th} target value; k is the total number of target values; $\hat{x}_\alpha$ is the approximation or predictions for the α^{th} value; and $\bar{x}_\alpha$ is the average of the series x_α.

After the network's ability to accurately predict test functions had been validated as above, data from Retailcorp were used to train and test the network. Figure 6(a) shows the raw data from the sales of one particular item of Retailcorp. The lines have three different patterns for training data, cross-validation data, and test data. (The difference between the latter two sets is explained in the next section.) Figure 6(b) shows the behaviour of the inventory levels during the training phase, and the thresholds required to keep the inventory to a minimum. These threshold levels are used as the target values during the training phase. Figure 6(c) shows the networks performance; for a training of 2000 epochs, ε/σ was 0.502.

It should be noted that as ε/σ tends to zero, the predicted values tend towards the test values, and if ε/σ tends to one, the prediction is no better than the simple mean. The top graph of Figure 7 shows the output patterns of the network, both multiple step and single step against the required threshold. The error values ε/σ were close to one, which implies a need for future refinement; this process is described in the next section of this paper. The same network was tested with a sinusoidal function shown in the bottom graph of Figure 7. The graph shows the predictions made by the network, both for the multiple step prediction (msp) and the single step prediction (ssp), and highlights the ability of the network to learn the regular pattern of the low frequency sinusoidal function.

When the predicted values were used as threshold values and subsequently analysed as a time series using NNs, the total inventory size was found to be significantly reduced. As the NN algorithm is concerned only with minimising the squared errors (and does not ensure that the predictions stay above the target values), the customer satisfaction levels dropped by about 10–15%. To ensure 95% customer satisfaction levels, a cross-validation technique with a surrogate parameter was used. This method (discussed in

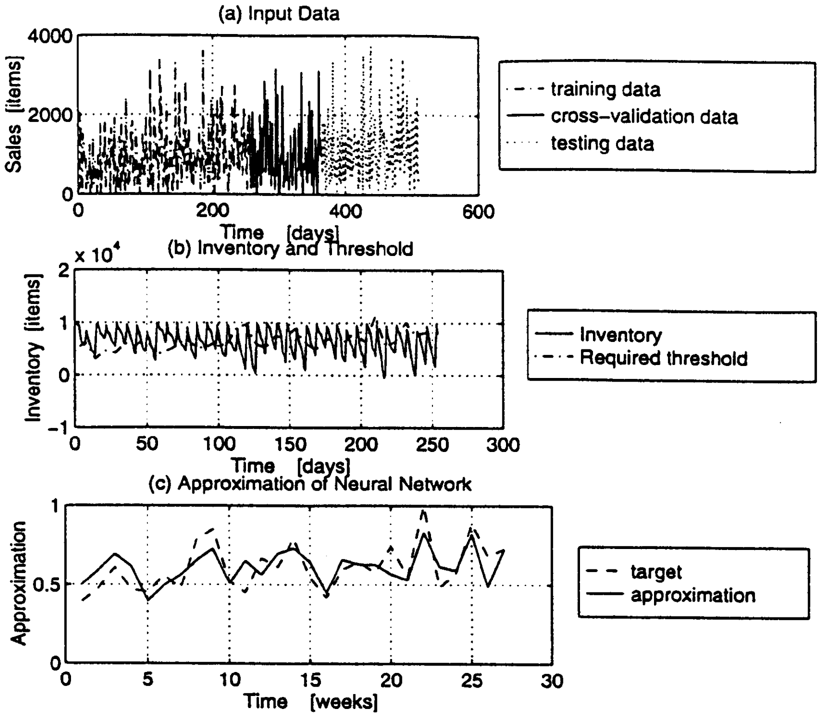

Figure 6 Approximation of neural network parameters through cross-validation.

the next section) ensured the lowest possible inventory levels while ensuring adherence to the 95% customer satisfaction constraint.

The results are shown in Table 1. 'Test 1', 'Test 2' and 'Test 3' denote the (hypothetical) test data sets that are based on a mixture of two sinusoidals of different frequencies; one sinusoidal series with noise; and a mixture of two sinusoidals with an increasing trend respectively. 'Item 1', 'Item 2', 'Item 3' and 'Item 4' refer to four different items that Retailcorp carries in its inventory. In six out of the seven cases, the addition of the surrogate parameter increases the customer satisfaction level, as well as the

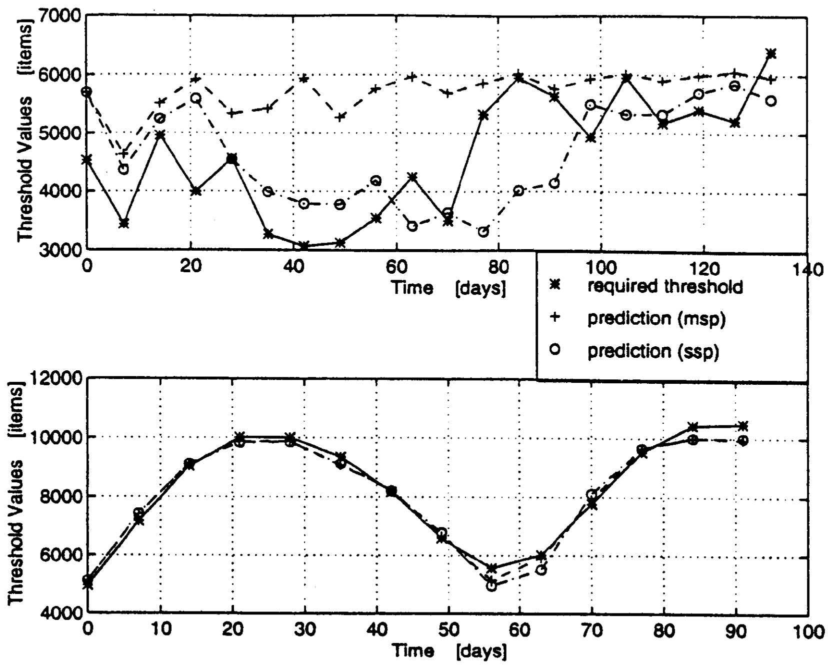

Figure 7 NN outputs for multiple and single step prediction for Retailcorp data and test sets.

Table 1 Effect of surrogate parameter in hybrid NN models

	Satisfaction level without surrogate parameter	*Satisfaction level with surrogate parameter*	*Total inventory without surrogate parameter [items]*	*Total inventory with surrogate parameter [items]*
Test 1	94.1%	98%	2.53×10^5	2.73×10^5
Test 2	97%	97%	3.064×10^5	3.31×10^5
Test 3	96%	98%	4.059×10^5	4.10×10^5
Item 1	82.1%	97.1%	9.8×10^4	2.13×10^5
Item 2	86.4%	95%	4.48×10^5	6.6×10^5
Item 3	92.8%	95.7%	1.143×10^5	1.38×10^6
Item 4	85.7%	96.4%	9.66×10^5	1.34×10^6

total inventory figures. For the real data from Retailcorp, the increase in the satisfaction level was more significant than for the hypothetical test data sets, and this is due to the randomness of the time series generated from the real data.

Hybrid model with surrogate parameters

As mentioned above, a number of NN configurations yielded significant reduction of inventory levels, but carried an associated reduction in customer satisfaction. This problem arises because the NN algorithm uses the summed *squared* error factor as the cost function. This error does not take into account the sign of the difference between the approximated value and the actual value. Further more, the existing learning algorithms described above only deal with the magnitude of the derivative, and not its sign. To mitigate this problem, an artificial compensation factor was included by us through cross-validation as described in the following paragraphs.

The data were again grouped into two categories: training data and testing data. The training phase was subdivided into two subsections: 'pure training' and 'cross-validation'. The pure training phase concentrated on determining the most probable NN parameters (weights and bias) and the cross-validation phase focused on reducing the undershoots. Assuming that the probability distribution of the errors conforms to a Gaussian distribution, the final thresholds were scaled up by a parameter which is a function of ε, which in turn is the standard deviation of the error and is determined from the training data. Assuming a linear form for this function, the corresponding parameter was determined during the cross-validation phase. By experimentation, a simple factor of one-third was determined to be adequate to ensure the 95% constraint, thus confirming that the results of the NN model were very close to the optimum point. Incidentally, a different technique based on Bayesian considerations has been proposed by some researchers.[25,30–32]

The final results are shown in Table 2. Again, 'Test' data sets and 'Item' data sets are evaluated, and for each case, the hybrid model performs significantly better than the models described in previous sections of this paper. (Please note that in both Tables 1 and 2, 'Test' refers to non-data items and 'Item' refers to real data.)

Substantial reduction in inventory levels were achieved in three of the four cases of the Retailcorp data, and marginal reduction was achieved in the fourth case of the Retailcorp data. These reductions occurred while ensuring that the customer satisfaction level was at least 95% in all cases. Further, the hybrid model performed better than the traditional model and the NN model in all four cases. The best performance was observed for Item 1, while Item 4 showed only marginal improvement. This disparity may be attributable to the performance of the NN model ($\varepsilon/\sigma = 0.65$ for Item 1 $\varepsilon/\sigma = 0.78$ for Item 4). It should be noted that while for Item 4 the improvement in inventory was lower than the other cases, the customer satisfaction was nevertheless increased. The above results demonstrate the

Table 2 Traditional model versus hybrid NN plus surrogate parameter

	Satisfaction level with traditional model	*Satisfaction level with NN model*	*Inventory level with traditional model*	*Inventory level with NN model*	*Total reduction in inventory size*
Test 1	98%	98%	4.09×10^5	2.73×10^5	33.2%
Test 2	99%	97%	5.22×10^5	3.31×10^5	36.5%
Test 3	95%	98%	4.66×10^5	4.10×10^5	12.0%
Item 1	98.5%	97.1%	3.12×10^5	2.13×10^5	31.7%
Item 2	97.85%	95%	8.6×10^5	6.6×10^5	23.2%
Item 3	99.2%	95.7%	1.732×10^6	1.38×10^6	20.3%
Item 4	95%	96.4%	1.377×10^6	1.34×10^6	2.6%

capability of the hybrid model to outperform both the traditional method and a NN model while handling two competing criteria (of minimising inventory levels and achieving high availability). We believe similar results will occur with other types of data sets as well.

Conclusions

We examined two approaches (traditional and Neural Network based) for minimising inventory levels while ensuring high customer satisfaction. Both of these approaches performed better than the simple ‘three weeks of supply’ policy currently pursued by a major retail organisation (Retailcorp). Next, we proposed a hybrid model that combines both these methodologies and introduces a new surrogate parameter. The hybrid model performed significantly better than existing models both for the set of hypothetical data and the set of real data.

The traditional model used a Dynamic Programming system to identify a running inventory level (or threshold) required to maintain the desired customer satisfaction level. This level was derived as a function of the preceding sales values. The threshold value, rather than the actual sales values, served as the centre of focus for our model. One advantage of this approach is that the lowest inventory level corresponding to given customer satisfaction probability could be analysed without having to model the transactional sales data that are characterised by the presence of zeros and high noise levels.

In the second model, we trained neural networks to learn the underlying trend in the threshold values, and created forecasts for the future. This model performed well in terms of minimising the inventory level, but not in terms of ensuring conformity to customer satisfaction level; this deficiency arose from the structure of the NN algorithm itself.

We introduced the surrogate parameter in the hybrid model to mitigate the problem associated with the NN learning rule. Assuming the error distribution at each point to be zero-mean Gaussian, the forecasted values were enhanced through cross-validation. The hybrid model with this surrogate parameter performed significantly better than either of the other two models (traditional and Neural Network based).

We believe that the combination of NNs with traditional data analysis tools presents a major opportunity to create sophisticated data mining capabilities and applications.

Acknowledgements—The authors wish to thank AnnFrances Yen for her valuable assistance with this paper.

References

1 Sani B and Kingsman BG (1997). Selecting the best periodic inventory control and demand forecasting methods for low demand items. *J Opl Res Soc* **48**: 700–713.
2 Wagner HM and Whitin TM (1958). Dynamic version of the economic lot size model. *Mgmt Sci* **5**: 89–96.
3 Hadley G and Whitin TM (1963). *Analysis of Inventory Systems*. Prentice-Hall: Englewood Cliffs.
4 Tersine RJ (1988). *Principles of Inventory and Materials Management*. North Holland: New York.
5 Agrawal N and Smith SA (1996). Estimating negative binomial demand for retail inventory management with unobservable lost sales. *Naval Res Logist* **43**: 839–861.
6 Nahmias S (1994). Demand estimation in lost sales inventory systems. *Naval Res Logist* **41**: 739–757.
7 Nahmias S and Smith SA (1994). Optimizing inventory levels in a two echelon retailer system with partial lost sales. *Mgmt Sci* **40**: 582–596.
8 Wecker WE (1978). Predicting demand from sales data in the presence of stockouts. *Mgmt Sci* **24**: 1043–1054.
9 Schwarz LB, Deuermayer BL and Badinelli RD (1985). Fill-rate optimization in a one-warehouse *N*-identical retailer distribution system. *Mgmt Sci* **31**: 488–498.
10 Saipe A (1997). Perfect Service—Every Time. *Logist Q* **3**: 4–5.
11 Bertsekas DP (1987). *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall: Englewood Cliffs.
12 Bansal K, Vadhavkar S and Gupta A (1998). Neural Networks Based Forecasting Techniques for Inventory Control Applications. *Data Mining and Knowledge Discovery* **2**: 97–102.
13 Kleijnen JPC (1987). *Statistical Tools for Simulation Practitioners*. Marcel Dekker Inc: New York.
14 Wolski R (1996). Dynamically Forecasting Network Performance Using the Network Weather Service. UCSD Technical Report Number TR-CS96-494. University of California at San Diego: San Diego, CA.
15 Vemuri VR and Rogers RD (1994). *Artificial Neural Networks, Forecasting Time Series*. IEEE Computer Society Press: Los Alamitos, CA.
16 Swingler K (1996). Financial prediction: Some pointers, pitfalls and common errors. *Neural Comput Applic* **4**: 192–197.
17 Weigend AS and Gershenfeld NA (eds) (1992). Time Series Prediction. *Forecasting the Future and Understanding the Past*. Addison Wesley: Reading.
18 Aleksander I (ed) (1989). *Neural Computing Architectures*. MIT Press: Cambridge.
19 Hertz J, Krogh A and Palmer RG (1991). *Introduction to the Theory of Neural Computation*. Addison Wesley: Redwood City.
20 Bishop CM (1996). *Neural Networks for Pattern Recognition*. Oxford University Press: Oxford.
21 Demuth H and Beale M (1993). *Neural Network Toolbox*. The MathWorks Inc.: Natick.
22 Rumelhart DE, Durbin R, Golden R and Chauvin Y (1995). Backpropagation: the basic theory. In: Chauvin Y and Rumelhart D (eds). *Backpropagation: Theory, Architecture, and Applications*. Stanford University: Palo Alto, USA, pp 1–34.
23 Bergman A, Grassberger P and TP Meyer (1992). Forecasting probabilities with neural networks. In: M Casdagli and S Eubank (eds). *Nonlinear Modeling and Forecasting*. Addison-Wesley: Redwood City, USA, pp 305–315.
24 Makridakis S and Wheelwright SC (1987). *The Handbook of Forecasting: A Manager's Guide*. John Wiley, Chichester.
25 Mackay DJC (1994). Bayesian non-linear modelling for the prediction competition. *ASHRAE Trans V. 100, Pt. 2*. 1053–1062.
26 Faraway J and Chatfield C (1995). Time series forecasting with neural networks: A case study. Research report 95-06 of the statistics group, University of Bath.
27 Chatfield C (1989). *The Analysis of Time Series, An Introduction*. Chapman and Hall: London.

28 Bertsekas DP and Gallager R (1992). *Data Networks*. Prentice-Hall: Upper Saddle River.
29 Miyano T and Girosi F (1994). Forecasting global temperature variations by neural networks. A.I. memo no. 1447 c.b.c.l memo no. 101, August, M.I.T. Artificial Intelligence Laboratory and Center for Biological Computational Learning.
30 Jordan MI and Bishop CM (1996). Neural networks. In: Tucker A (ed), *CRC Handbook of Computer Science*. CRC Press: Boca Raton, USA, pp 536–556.
31 Mackay DJC (1991). *Bayesian Methods for Adaptive Methods*. PhD thesis, California Institute of Technology, USA.
32 Mackay DJC (1992). A practical Bayesian framework for back-propagation networks. *Neural Comput* **4**: 448–472.

Received August 1997;
accepted August 1998 after two revisions