IFAC
Publications
www.elsevier.com/locate/ifac

MODELS FOR INCOMING CALLS FORECASTING IN A CUSTOMER ATTENTION CENTER

Manuel R. Arahal, Fernando Pavón P., Eduardo F. Camacho

Departamento de Ingeniería de Sistemas y Automática.
Escuela Superior de Ingenieros.
Camino de Los Descubrimientos s/n. 41092 Sevilla

Abstract: Telephone customers attention centers (CAC) are complex systems. In order to provide the best service to clients with minimum costs a careful scheduling of human resources (agents) is needed.
Call centers often receive thousands of incoming calls. A large data base of services is in many cases available for modelling. Such data has been used in different ways to improve the quality of service. In this particular case, the schedule of attention staff a week in advance.
In this paper the number of incoming calls in the hour is modelled using autoregressive models, both linear and nonlinear (neural networks). As it turns out, the most important part of the modelling procedure is the selection of appropriate input variables. *Copyright © 2003 IFAC*

Keywords: Call center, forecasting, linear models, nonlinear models, neural networks

1. INTRODUCTION

Call centers are nowadays ubiquitous in companies that have to deal with large number of customers, some (like airlines, hotels, and credit card companies) relay almost exclusively in call centers for service providing and customer feedback.

The capacity of a call center is mostly determined by the human resources employed. Since these are expensive, the quality of service is often balanced with capacity so that the call center can provide the best service with minimum costs (Pinedo *et al.*, 1999).

Forecasting techniques can be useful to schedule the number of agents needed to provide a desired quality of service at any time.

Simple time-series methods have been used to forecast call center load, such as in (Sze, 1984). Andrews and Cunningham (Andrews and Cunningham, 1995) develop ARIMA models that estimate the number of daily calls at L.L. Bean. A statistical method for predicting arrival rates can be found in (Jongbloed and Koole, 2001).

Apart from that we have found very few papers describing the application of time series analysis to call centers load forecasting.

The model in (Andrews and Cunningham, 1995) is used to predict daily call volumes for the next three weeks in order to produce efficient agent scheduling. The ARIMA model is a linear combination of lagged values of the independent variable (daily call volumes) and past prediction errors. The number of regressors and their lag were cho-

sen using a 5-year data base. The resulting models yield a mean absolute percent error of about ten in a year of use after construction (ex-post forecasts).

In this paper we present different autoregressive models for forecasting the hourly call load for an information call center of a Spanish phone company. Such forecast is then used to schedule the number of agents needed to attain a prescribed level of efficiency. The paper also discuses different modelling techniques and the results obtained. The work extends the results obtained previously in the same context ((Andrews and Cunningham, 1995)) by using neural models and a systematic procedure for selecting input variables. The paper also shows that for this particular problem, the key issue is an appropriate selection of input variables and not the complexity of the model chosen. Furthermore, the best results from a forecasting point of view are obtained for simple models with few input variables.

Once a phone call is received an agent is designed depending of some characteristics of the call: code area, customer, kind of information requested, language of the caller. A call can be transferred from an agent to another. Each time an agent manages a call an attention counter is increased. Thus a single call can produce many attentions. Thousands of calls and many more attentions are managed each day in the call center.

Lost calls are a figure of merit used by the company to assess the quality of the service. Three are the reasons for a lost call:

- Calls lost while in queue. These are due to lack of agents to manage all incoming calls. The client hangs because the waiting is to long.
- Calls lost by saturation. These are due to lack of space to enqueue more calls. The call is not answered in any way.
- Calls lost not in saturation. Most are due to technical failure of communication. These calls are considered correctly attended because is not a fault in the part of the CAC.

A large data base is available consisting of hourly values of variables such as number of incoming calls, number of internal calls, average length of calls, etc. In order to produce a schedule for the agents the most important variable is the number of incoming calls. In the next section the forecasting problem is posed in a more precise way. Section 3 shows the linear and neural models developed followed by the results obtained in the application. The paper ends with some conclusions.

2. DATA

Let us denote by $v(d,h)$ the number of incoming calls received at the CAC at day d and hour h. Days are numbered consecutively, thus d represents a unique day in the data base. Hours are numbered from 1 to 24. The number of incoming calls will also be referred to as load.

For scheduling reasons, it is paramount to forecast the load with a week in advance. That means that if k is a Monday, the forecast for next-week Monday $k+7$, next-week Tuesday $k+8$, and so on until next-week Sunday $k+13$ are needed. Since the forecast has to be made the very day k that means that information about k is not available yet. This amounts to a lead-time in the predictions ranging from 8 to 14 days.

In order to simplify the problem it is better to consider just the maximum lead time and make predictions with a 14-day horizon. To be more precise: we are to produce $\hat{v}(d+14,h)$ for hours $h = 1,...,24$ using information about past loads $v(k,h)$ for days $k = 1,...,d$.

Data is composed of historical information from the call centers of a Spanish phone company. It consists of hourly load (number of incoming calls every hour) for a number of months. The data base consists of values of load $v(d,h)$ for each hour h and day d. It can also be viewed as a time series $x(t)$ being $t = 24d + h$.

Figure 1 shows the number calls for 31 days. The vertical axis has been scaled (to zero mean and variance unity) for confidentiality reasons.

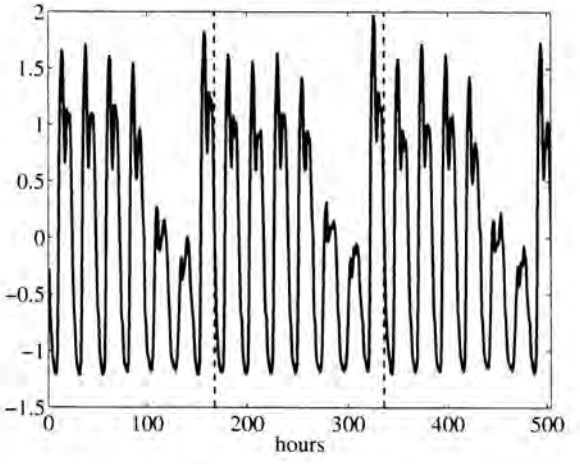

Fig. 1. Scaled number of incoming calls during 21 days at the CAC.

Periodicity in the data is clearly visible. The power spectrum (Figure 2) shows a number of peaks at 8, 12, 24, 28, 84 and 178 hours per cycle. The most prominent is the 24 hour cycle. This observation opens the way for autoregressive models.

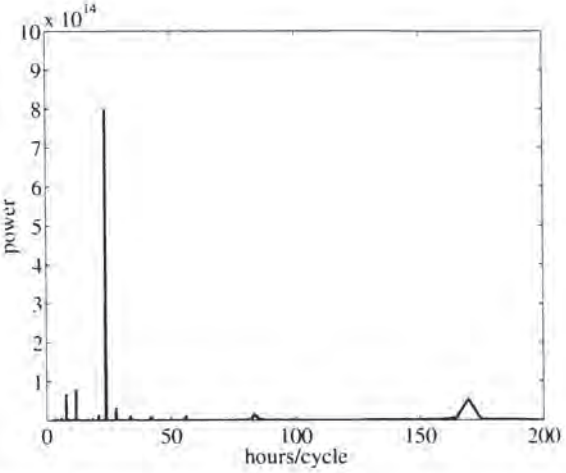

Fig. 2. Power Spectrum of $x(t)$.

3. MODELS

Autoregressive linear models are the first choice for the problem considered. Different orders and time lags can be easily tested using computer programs such as ident for MATLAB.

The general structure for a linear autoregressive (AR) model is

$$\hat{x}(t) = \sum_{k=1}^{na} a_k x(t - \tau_k) \qquad (1)$$

being na the order of the model, $a_k, k = 1, ..., na$ the coefficients and $\tau_k, k = 1, ..., na$ the time lags. The past value $x(t - \tau_k)$ is referred to as regressor. An AR models produces predictions as a linear combination of past values of the observed variable. In order to construct a model, appropriate values for the order and time lags have to be searched for. Then, the coefficients can be obtained by least squares.

In many cases time lags are consecutive multiples of a basic lag-unit T plus a constant dead-time d, that is: $\tau_k = d + (k - 1)T$. In this case, the only choices are na, d and T. Normally d is the lead-time of the prediction, fourteen days in the present case.

Akaike's criterion and others may be used to choose the value of na that achieves a balance between accuracy and complexity of the model.

Regressors can also be selected using a forward inclusion technique. In forward inclusion lagged variables are included in the model one by one, selected according to how much they help reducing the model error (see (Arahal *et al.*, 2002)). The problem arises on when to stop since any new variable will introduce new degrees of freedom that will allow to reduce the error in a controlled set of observations. Validation techniques must be carefully used to perform model selection.

In the application we are tackling the order must be kept as low as possible since we do not have large amounts of data needed to develop higher order models.

The adjustable parameters of the models are obtained minimizing a quadratic criterion of the prediction error. A set of observations is set aside to test the validity of the model. The root mean squared error over this set is defined as:

$$J = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(x(t) - \hat{x}(t))^2} \qquad (2)$$

being N the number of observations in the validation set. J will be used through the paper as a figure of merit to compare different models.

3.1 Linear models

Models of the form:

$$\hat{x}(t) = \sum_{k=1}^{na} a_k x(t - d - (k-1)T) \qquad (3)$$

will be considered first, being $d = 14 \cdot 24 = 336$ the lead time of the prediction. Table 1 shows the root mean squared prediction error for different values of na and T. It can be seen that variables lagged 24 hours ($T = 24$) are the most informative but even in this case a large model order is needed to reduce the error significantly.

$T = 1$		$T = 8$		$T = 24$	
na	J	na	J	na	J
1	0.306	1	0.306	1	0.306
2	0.306	2	0.306	2	0.304
3	0.306	3	0.305	3	0.304
10	0.304	10	0.303	10	0.278
20	0.303	20	0.303	15	0.268
50	0.301	25	0.276	30	0.247

Table 1. Root mean squared error for different AR linear models.

Forward inclusion of variables can be easily tested. Models now have the form

$$\hat{x}(t) = \sum_{k=1}^{na} a_k x(t - d - l_k) \qquad (4)$$

being l_k selected lags not necessarily consecutive. The first lag l_1 is selected comparing the correlation coefficients between the corresponding regressor $x(t - d - l_1)$ and the observed variable $x(t)$. The variable with the highest correlation is chosen because it is the best suited to approximate $x(t)$.

The model with just one regressor yields predictions $\hat{x}_1(t) = a_1 x(t - d - l_1)$, being a_1 a coefficient estimated via least squares. The error of this model is $e_1(t) = x(t) - \hat{x}_1(t)$. The selection

of a new input variable is done as previously but looking for high correlations with $e_1(t)$.

Table 2 shows the root mean squared prediction error for different models. The first column na is the number of input variables used by the model. The second column shows the values of lags l_k (see 4). It can be seen that the more variables the better results are obtained. We must keep in mind that these results are not forecasting but estimation results since data has been used up in building the models.

na	lags l_k	J
1	0	0.306
2	0, 504	0.274
3	0, 504, 215	0.271
4	0, 504, 215, 599	0.270
5	0, 504, 215, 599, 583	0.269

Table 2. Root mean squared for linear AR models with selected non consecutive lags.

3.2 Neural models

The latter strategy of forward inclusion can be of use for nonlinear models such as neural networks. One-hidden layer neural networks with linear output node have been selected for the sake of simplicity. A neural net computes a nonlinear function of its input vector $\mathbf{z}$ as:

$$\mathrm{NN}(\mathbf{z}) = \sum_{n=1}^{nn} w_n^o s_n(\mathbf{z}) \tag{5}$$

being nn the number of nodes in the hidden layer and $s_n(\mathbf{z})$ the output of the n-th node, calculated as

$$s_n(\mathbf{z}) = \tanh(\sum_{k=1}^{na} w_k^i z_k) \tag{6}$$

being na the dimension of the input vector. Coefficients w_n^o and w_k^i are the adjustable parameters of the network and are referred to as weights. Training is the procedure (gradient based in most cases) for assigning a value to weights so that the approximation error is made small.

In order to use the net as a AR model, the input vector must contain lagged values of $x(t)$ such that $z_k = x(t - d - l_k)$.

Forward inclusion now poses a problem. Since the expected nature of relationship between input variables and observed output is nonlinear it does not make sense to look for linear correlations among them. The procedure to select variables should be done in a different manner. A possible approach is to use brute force and compute models for all possible combinations of two input variables, then of three and so on. Although time consuming this approach should be taken into account. Another possible way of proceeding is to use a nonlinear measure of correlation such as the one proposed in (Yuan and Fine, 1998).

Table 3 shows the results obtained at different stages of the forward inclusion procedure.

na	lags l_k	J
1	0	0.308
2	0, 336	0.249
3	0, 336, 48	0.245
4	0, 336, 48, 288	0.242
5	0, 336, 48, 288, 480	0.237

Table 3. Root mean squared error for neural AR models with lags selected using a nonlinear correlation measure among variables.

The neural network training has been done using the MATLAB neural toolbox. The data base of input/output pairs has been split in two random separate sets of equal cardinality. One set (training set) is used for adjusting the weights and the other (validation set) to avoid overtraining. The value of J presented in table 3 is calculated after training using both sets.

3.3 Ad hoc models

Results obtained so far are somehow poor. The estimated error in the prediction is about 20 % of the mean volume of incoming calls. In order to obtain sharper scheduling operations it is necessary to improve the forecasting ability of the models.

A study of the data reveals some facts:

- Week-days and week-ends have very different loads.
- Load in any particular day is more similar to load of previous days of the same kind.
- Holidays in the middle of a week have a load similar to that of a Sunday.

According to this it seems sensible to modify the way regressors are obtained. Instead of using fixed values for lags, ones should always try to use past values from similar kind of days. That is, to forecast a Monday data from past Mondays should be used and the same holds for Tuesdays to Sundays and Holidays.

The new model structure can be mathematically described as:

$$\hat{x}(t) = \sum_{k=1}^{na} a_k x(t - d - 24r_k(t)) \tag{7}$$

where $r_k(t)$ represents a variable number of days. The correct value for $r_k(t)$ is obtained searching the database for days similar to the forecasted day.

This can be clarified with an example. Suppose that today is a Monday and we are to forecast next week. In order to forecast the load for next Monday we can use load from previous Monday $r_1 = 0$, and the one before $r_2 = 7$, etc. Unless one of these Mondays happened to be a holiday. Then this day is skipped and the previous one is considered.

With this in mind the r_1 can be described as the minimum "distance" (in days) that one has to regress to obtain a day of the same type as the forecasted day (discounting the lead time of fourteen days). Similarly r_2 is the distance to the second closest day of the same kind, and so on.

Note that in this way holidays are never used as regressors to forecast a day that is not a holiday. In the same way data from a Monday is not used to forecast load of a day that is not a Monday.

Model order can be selected just trying out increasing values for na . Table 4 shows the results obtained.

na	J
1	0.210
2	0.197
3	0.188
4	0.186

Table 4. Root mean squared error for linear AR models with variable lags.

Neural networks can also be used with this new way of selecting regressors. The new model structure can be mathematically described as $\hat{x}(t) = \mathrm{NN}(\mathbf{z})$, being NN() a nonlinear function implemented by a neural net with input vector $\mathbf{z}$. The na components of this vector are selected past values of $x(t)$ obtained as $z_k = x(t - d - 24r_k(t))$ as explained above. Table 5 shows the results obtained.

na	J
1	0.195
2	0.189
3	0.180
4	0.176

Table 5. Root mean squared error for neural AR models with variable lags.

It can be seen that results have improved, being now possible to achieve root mean squared errors of about 15% of the mean load with low order models.

4. APPLICATION RESULTS

The models generated in the previous section have been tested during a period of 41 days. Table 6 shows the results in terms of root mean squared error of normalized variables J^n as in the previous sections. Also, for the sake of comparison with other papers, other measures of forecasting error are given. In the third column the root mean squared error as a percentage of the mean load is shown (J^p). This figure of merit is nothing but the root mean error of the non-normalized variable divided by the mean load and multiplied by one hundred. In some applications the arithmetic mean of the absolute value of the error is used instead. This is shown in the fourth column as MAE again as a percentage of the mean load.

A) linear models with lags $d + 24(k-1), k = 1, \ldots, na$

na	J^n	J^p	MAE
1	0.095	7.73 %	5.41 %
2	0.093	7.54 %	5.17 %
3	0.094	7.63 %	5.36 %
4	0.095	7.72 %	5.45 %
5	0.095	7.66 %	5.42 %
6	0.094	7.58 %	5.34 %
10	0.076	6.15 %	**4.31** %

B) nonlinear models with lags $d + l_k$

na	l_k	J^n	J^p	MAE
1	0	0.110	8.89 %	6.42 %
2	0, 336	0.069	5.62 %	**4.00** %
3	0, 336, 48	0.087	7.05 %	4.23 %
4	0, 336, 48, 288	0.080	6.51 %	4.37 %
5	0, 336, 48, 288, 480	0.088	7.12 %	4.95 %

C) linear models with variable lags

na	J^n	J^p	MAE
1	0.087	7.07 %	4.86 %
2	0.078	6.29 %	**4.45** %
3	0.101	8.16 %	5.59 %
4	0.116	9.41 %	6.40 %

D) nonlinear models with variable lags

na	J^n	J^p	MAE
1	0.090	7.28 %	5.06 %
2	0.085	6.88 %	**4.72** %
3	0.109	8.79 %	5.50 %
4	0.194	15.73 %	8.02 %

Table 6. Summary of application results of different models.

As can be seen in Table 6 the best results (typed in boldface) are achieved by very low order models. The performance is in some cases deteriorated by the use of new input variables. This was not previously detected although a large validation set was used to avoid overtraining.

Figure 3 shows the normalized number of incoming calls and the forecast made by the best model of each group during a typical week. A portion of the curve has been amplified to show the differences among models.

To make the comparison more apparent the error has been also drawn for each model in Figure 4.

It is also noticeable that simple but intuitive models, can perform as well as other derived by more sophisticate procedures.

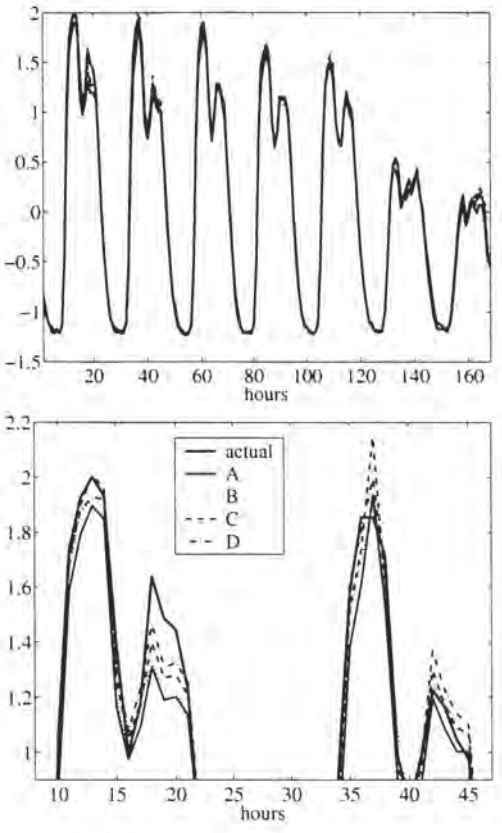

Fig. 3. Normalized load and forecast for the bests models of each group (see table 6 and text for details).

Going back to the scheduling of agents, any of the proposed models produce predictions with root mean squared error under 10% of the mean load, allowing to attain a balance between lost calls and staffing.

5. CONCLUSIONS

It has been shown that techniques of system identification and neural networks can be used to obtain models to forecast the number of incoming calls to a CAC.

The results obtained in this application shows the importance of selecting input variables and model order for forecasting. In this particular case it has been shown that overestimation of model order has a negative effect on model forecasting performance.

REFERENCES

Andrews, B. and S.M. Cunningham (1995). L.L. bean improves call-center forecasting. *Interfaces* **25**, 1-13.

Arahal, M.R., Alfonso Cepeda and E.F. Camacho (2002). Input variable selection for forecasting models. *Preprints of the 15th triennial world congress of the IFAC*.

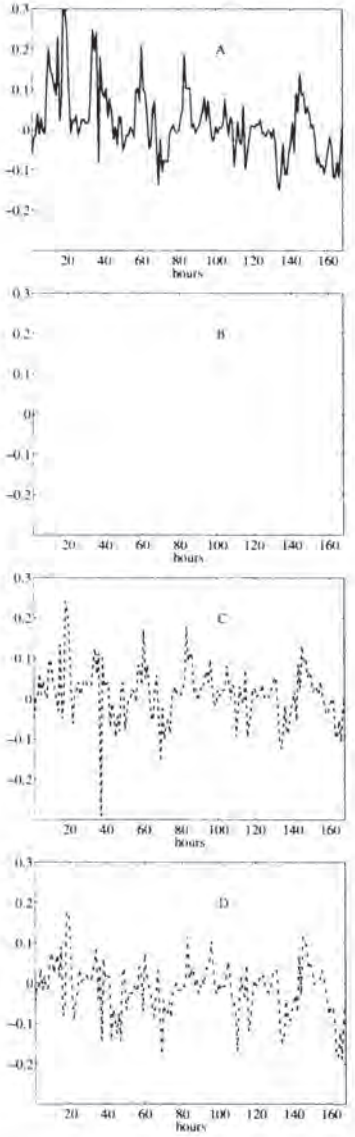

Fig. 4. Normalized forecasting error for the bests models of each group (see table 6 and text for details).

Jongbloed, G. and G.M. Koole (2001). Managing uncertainty in call centers using poisson mixtures. *Applied Stochastic Models in Business and Industry* **17**, 307–318.

Pinedo, M., S. Seshadri and J.G. Shanthikumar (1999). Call centers in financial services: strategies, technologies, and operations. In: *Creating Value in Financial Services: Strategies, Operations and Technologies* (E.L.Melnick, P. Nayyar, M.L. Pinedo and S. Seshadri, Eds.), Chap. 18, pp. 357–388. Kluwer.

Sze, D.Y. (1984). A queuing model for telephone operator staffing. *Operations Research* **32**, 229–249.

Yuan, J.-L. and T.L. Fine (1998). Neural-network design for small training sets of high dimension. *IEEE Transactions on neural networks* **9**, 266–280.