Time Series Prediction of Telephone Traffic Occupancy using Neural Networks

R. M. Goodman and B. E. Ambrose
California Institute of Technology
Pasadena, CA91125, USA
Ph: (818) 3956811 Fax: (818) 5688670
email: bambrose@micro.caltech.edu

Abstract

Various techniques, including feed-forward neural networks, are applied to the time series prediction problem. The forecasting of occupancy on a telephone trunk group is taken as a case study. The relative performances of the techniques are reported. Theoretical justifications are provided for the results.

1 Introduction

Pacific Bell and Caltech have for some time been working on a real-time traffic management/expert system. This project is called NOAA, Network Operators Advice and Assistance. The task of NOAA is to take information from the Pacific Bell network management computer, use it to isolate and diagnose exceptional events in the network and then recommend the same corrective advice as network management staff would in the same circumstances.

The development of the expert system has been reported in previous papers [1, 2]. The occupancy predictor is one module in this system and is used to aid traffic rerouting. Occupancy is a good indicator of spare capacity on a trunk group.

Although the problems described in this paper are concerned with monitoring telephony traffic, the techniques should be applicable to the monitoring of any large network, with little modification. For example, rather than monitor trunk occupancy, the network manager may be monitoring link throughput, but the same analysis techniques should apply.

2 Data set

The data set for this study consisted of about 1500 observations of occupancy of a single trunk group taken every 5 minutes over 7 days. Occupancy is defined as the moving average of twenty samples of the number of trunks occupied on a route. The samples are taken every 30 seconds and the result is scaled to be between 0 and 1. The occupancy for every trunk group in the network is reported every five minutes. The first 300 points of the data set are shown in Fig. 1. Some key features to note are:

- the traffic level varies according to time of day
- spikes may be present in the time series, e.g. close to example 60
- the variance of the occupancy varies with the traffic level

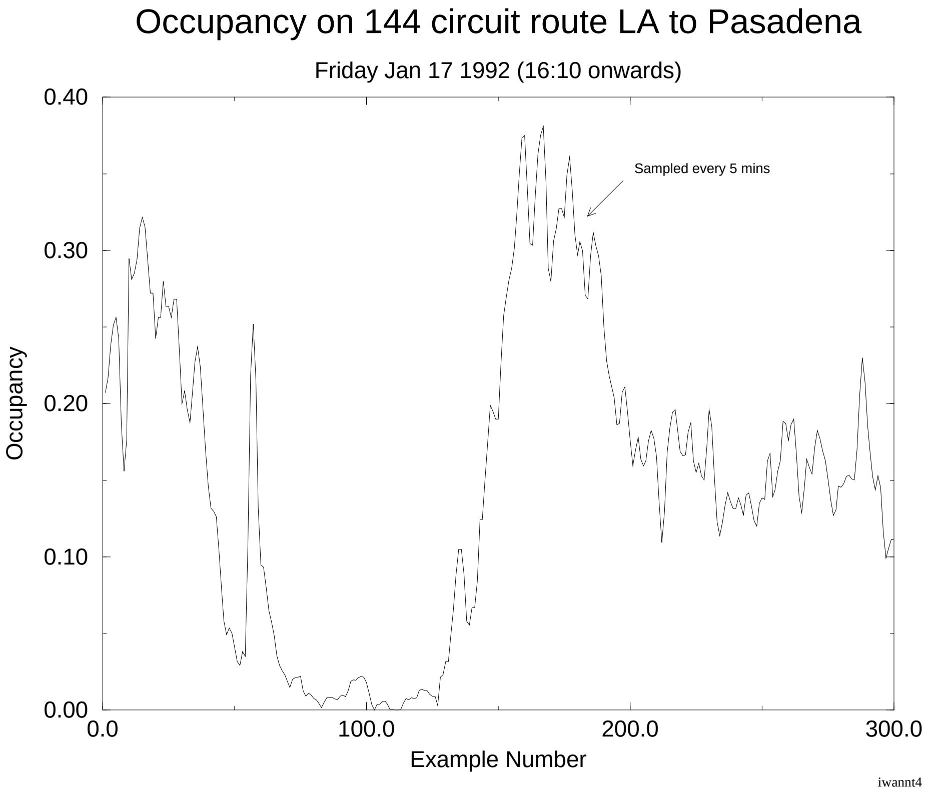

Figure 1: Occupancy Dataset

Method	Error
Using linear predictor	16215
Using non-linear prediction	16167
Using neural network	16119
Using local approximation	16085
Using Hidden Markov Model	16081
Using log transformation	16033

Table 1: Scaled RMS Prediction Error for Various Prediction Methods

3 Cross-validation

In testing the relative merits of prediction techniques, a distinction must be made between learning ability and generalisation ability. A good prediction method will generalize well on examples that have not been seen before. This can be tested for using cross-validation. With v-fold cross-validation and a data set of size N, we carry out v tests using all but N/v of the data set to provide the training set and then testing on the N/v test set examples that were not in the training set. This makes maximum use of the data set and allows us to check the significance of the results.

4 Results

The results are shown in Table 1. All methods predict next observation based on previous 6 observations. The linear predictor can be taken as the baseline performance to beat. Error is RMS prediction error multiplied by 1,000,000, with a difference of 100 being significant. 50-fold cross-validation was used.

5 Linear Predictor

It is known that for a process that has a multivariate Gaussian distribution the Linear Predictor (LP) is the best predictor[3]. Since the Poisson or telephone traffic distribution looks like a Gaussian for large traffic levels, it not surprising that the linear predictor did well. The other nonlinear methods can only give small percentage improvements.

To find the LP weights, a matrix equation can be written for the residual errors r_i of an approximate solution of the prediction problem in terms of the occupancy observations y_i as follows:

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ \vdots \\ r_{1500} \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & \cdots & y_6 \\ y_2 & y_3 & \cdots & y_7 \\ y_3 & y_4 & \cdots & y_8 \\ y_4 & y_5 & \cdots & y_9 \\ \vdots & \vdots & \ddots & \vdots \\ y_{1500} & y_{1501} & \cdots & y_{1506} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_6 \end{bmatrix} - \begin{bmatrix} y_7 \\ y_8 \\ y_9 \\ y_{10} \\ \vdots \\ y_{1507} \end{bmatrix}$$

or in short,

$$r = Aw - b$$

Then $r^T r$ (the total square residual error) can be minimized with respect to the weight vector w by setting

$$r^T r = (Aw - b)^T (Aw - b)$$

and differentiating $r^T r$ with respect to w and setting the result to zero giving

$$A^T A w = A^T b$$

and hence

$$w = (A^T A)^{-1} (A^T b)$$

Thus the LP coefficients can be obtained by a simple 6x6 matrix inversion.

6 Non-Linear Predictor

Non-linear prediction allows the use of $y_i y_j$ crossproduct terms in the A matrix of section 5, but still uses matrix inversion to give the predictor coefficients. This is also known as polynomial regression. A slight improvement in prediction performance was obtained. Trial and error showed that the best results were obtained with pure squares of the input terms.

7 Neural Network

Previous comparisons of neural networks and linear predictors have shown that neural network sometimes can give better results [4, 5]. However the data sets used were not particularly long, so the statistical significance of these comparisons may be questionable.

Moody in [6] gives learning limits for neural networks. See appendix A for some details. A key point in his paper is that too many hidden units combined with a low value for weight regularisation will produce an increase in generalisation error. From this it can be deduced that there is an optimum number of hidden units for a given learning problem.

In our studies a feed-forward neural network, with a single hidden layer was used. Quickprop[7] was used for training as it was advertised as having faster convergence than standard back-prop and was freely available on the internet. For the neural network, trial and error showed that four hidden units and a linear output unit gave best results. This architecture was fixed prior to training.

A plot of hidden unit activations gave valuable insight into the features of the data set. One of the four hidden units reacted strongly to the overall traffic level, one of the units reacted strongly to rate of change of traffic level while the other two reacted strongly to the rate of rate of change of traffic level.

8 Local Approximation

Local approximation techniques work well for time series with no noise[8]. The assumption is that the mapping from input space (the six previous observations of occupancy) to output space (the next observation) is locally linear.

The local approximation technique trains a linear predictor on a subset of the training set which is in some sense close to the example to be predicted. The key point is that a different subset is used for each test example to be predicted. Euclidean distance was used as a measure of closeness. Trial and error showed best performance with using a subset of the training set of size 640. Prediction performance was reasonably good.

9 Hidden Markov Model

Hidden Markov Models (HMM) are popular for word classification in automatic speech recognition[9].

The HMM is defined by a set of states, S, and two matrices A_{ij} and B_{oi}. A_{ij} is the probability of a transition to state i from state j and B_{oi} is the probability of observing an observation of o given that you are in state i. This is illustrated in Fig. 2. The A_{ij} matrix gives rise to the *Markov* in the name, and the B_{oi} matrix gives rise to the *Hidden* in the name.

The Baum-Welsh algorithm can be used to learn the A and B matrices[9]. A different algorithm, the Baum Backward-Forward algorithm can be used to derive the probabilities of the system being in a particular state given the observations observed[9]. In this way, the HMM can be used as a classifier.

In our case, a HMM was used to classify the trunk group as being in one of two modes/states: (i) traffic varying a lot as during a traffic spike or (ii) traffic behaving normally. This classification was used to choose between two linear predictors for training purposes as shown in Fig. 3. For testing purposes, as shown in Fig. 4, the HMM output probability for state 1 (p_1) was used to combine the estimates from the two linear predictors (LP$_i$) to yield the occupancy estimate $\hat{y}$ as follows:

$$\hat{y} = p_1 * \mathrm{LP}_1 + (1.0 - p_1) * \mathrm{LP}_2$$

The observation, $o(n)$, used as input to the HMM model was the prediction error from a 5-

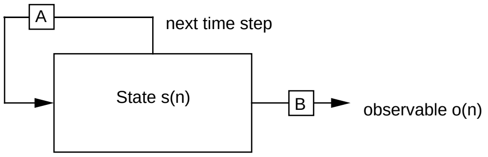

Figure 2: Hidden Markov Model

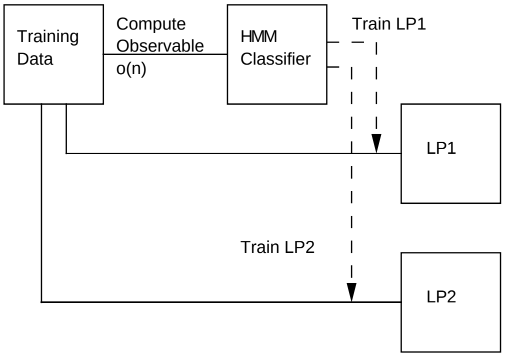

Figure 3: Training a Dual LP

input linear predictor in predicting the most recent occupancy reading. The A-matrix was learned from the data using the Baum-Welsh algorithm. An assumption of a Gaussian distribution of prediction error was used to generate the B-matrix.

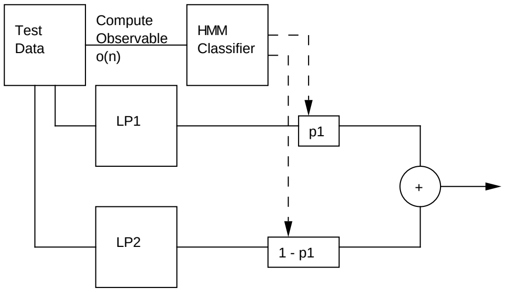

Figure 4: Testing a Dual LP

10 Log Transformation

A log transformation of the data was carried out, prior to training a linear predictor. This was the best prediction method. It is believed that the reason why it was so good is that it tended to give the same output as a linear predictor when the traffic levels were about constant. This was because the log predictor had approximately the same weights as the linear predictor, and averaging in “log space” is the same as averaging in linear space if the points being averaged are close together.

On the other hand, in the event of spikes (as occurred in a small part of the data set) the log predictor gave much better prediction results. This may reflect the geometric averaging process as being better than the arithmetic averaging process following a spike.

It might be argued that heteroskedasticity is the cause of the log predictor’s success. Heteroskedasticity is the change of variance of the occupancy statistic as the traffic level rises. This may interfere with the calculation of the coefficients of the LP. A possible solution is to take a log transformation of the data to flatten the variance as was done here. However heteroskedasticity can be ruled out as a cause of the log predictors success because in this event, a weighted least squares predictor should obtain the same improvements as the log predictor, and this was found not to be the case.

11 Conclusions

The spikes in the telephone traffic occupancy statistic mean that it is possible to do better than use a linear predictor for this data set.

The general nonlinear methods of neural networks and local approximation did well and can be expected to be near optimal as the data set size increases. Some progress in understanding the role of each hidden unit in the neural network predictor was obtained.

The Hidden Markov Model has a useful side effect of giving a classification of the state of the trunk group. This could be useful in a network management context.

Examination of a larger data set will be necessary before concluding that the log occupancy is the best prediction method, since the number of spikes was limited in this data set.

A Learning Limits for Neural Networks

J. E. Moody carried out an analysis of generalisation and regularisation in non-linear learning systems[6].

Assume a set of n real valued input/output data pairs were given and we had to estimate a function to fit the data. The noise was i.i.d. with mean zero and finite variance σ^2. The noise was not necessarily Gaussian.

For a linear predictor, references were given to the following result for the MSE e_{test}:

$$\mathrm{E}(e_{test}) \approx \mathrm{E}(e_{train}) + 2\sigma^2 \frac{p}{n}$$

where p is the number of parameters (weights) being estimated.

For a neural network, a new result correct to second order was given:

$$\mathrm{E}(e_{test}) \approx \mathrm{E}(e_{train}) + 2\sigma^2 \frac{p_{eff}}{n}$$

where p_{eff} is a complicated function of various Jacobians. However for a locally linear model, p_{eff} is a decreasing function of λ the weight decay parameter for the neural network with $p_{eff}(\lambda = 0) = p$.

References

[1] Goodman, R. M., Ambrose, B., Latin, H., Finnell, S., "Network Operations Analyzer and Assistant (NOAA): A real-time traffic rerouting expert system," *Globecom,* Florida, December, 1992.

[2] Goodman, R. M., Ambrose, B., Latin, H., Finnell, S., "Network Operations Analyzer and Assistant (NOAA): A hybrid Neural Network / Expert System for Traffic Management," *IFIP,* San Francisco, April, 1993.

[3] Christensen, R., *Linear Models for Multivariate, Time Series, and Spatial Data,* Springer-Verlag, New York, 1991.

[4] Tang, Z., de Almeida, C., Fishwick, P. A., "Time series forecasting using Neural Networks vs. Box-Jenkins Methodology," *Simulation,* 57:5, pp. 303–310, November, 1991.

[5] Sharda, R., Patil, R., "Neural Networks as Forecasting Experts: an Empirical Test," in *International Joint Conference on Neural Networks,* volume 1, pp. 491–494, Washington, D.C., 1990.

[6] Moody, J. E., "The effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems," *Advances in Neural Information Processing Systems 4,* Morgan Kaufmann, 1992.

[7] Fahlman, S. E., "Faster-Learning Variations on Back-Propagation: An Empirical Study" in *Proceedings of the 1988 Connectionist Models Summer School,* Morgan Kaufman, 1988.

[8] Farmer, J. D., Sidorowich, J. J., "Predicting Chaotic Time Series," *Physical Review Letters,* Vol. 59, No. 8, pp. 845–848, August, 1987.

[9] Levinson, S. E., Rabiner, L. R., Sondhi, M. M., "An Introduction to the Applications of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *Bell System Technical Journal,* Vol. 62, No. 4, pp. 1035–1074, April, 1983.