Evolutionary Regression and Neural Imputations of Missing Values

Chapter *in* Studies in Fuzziness and Soft Computing · January 1970

DOI: 10.1007/978-3-540-77465-5_8

3 authors, including:

Pawan Lingras
Saint Mary's University

Ming Zhong
Wuhan University of Technology

Evolutionary Regression and Neural Imputations of Missing Values

Pawan Lingras[1], Ming Zhong[2], and Satish Sharma[3]

[1] Saint Mary's University, Halifax, Nova Scotia, Canada
pawan@cs.smu.ca
[2] University of New Brunswick, Fredericton, New Brunswick, Canada
ming@unb.ca
[3] University of Regina, Regina, Saskatchewan, Canada
Satish.Sharma@uregina.ca

1 Introduction

While the information age has made a large amount of data available for improved industrial process planning, occasional failures lead to missing data. The missing data may make it difficult to apply analytical models. Data imputation techniques help us fill the missing data with a reasonable prediction of what the missing values would have been.

The generic techniques available for short term forecasting can also be used for data imputations. Researchers have explored the possibilities of applying advanced techniques to fully utilize available information for more accurate predictions. Various data mining functionalities have been applied to summarize data, analyze missing value distribution, classify and mine patterns, and visualize data (Cabena et al. 1998; Han and Kamber 2001). Sophisticated models based on genetic algorithms (GAs), time delay neural networks (TDNN), and locally weighted regression (LWR) analysis have been used for predictions in various fields. GAs are adaptive algorithms for finding practical solutions among an enormous number of possibilities. GAs have been widely explored in machine learning, scientific modeling, optimization, and engineering (Mitchell 1999). TDNN is a variant of neural network, which is particularly useful for modeling time series. Neural networks are good at recognizing patterns, generalizing and predicting trends. They are fast, tolerant to imperfect data and do not need formulas and rules. Neural networks have been used to make predictions and understand data in investments, medicine, science, engineering, marketing, manufacturing, and management (Lawrence 1993). LWR is a nonparametric regression analysis, which uses local variables for forecasting. Recent studies (Gorinevsky and Connolly 1994; Schaal and Atkeson 1994) have shown that LWR achieved higher accuracy than other parametric approaches. LWR have been increasingly applied in control and prediction (Atkeson et al. 1997).

The techniques described above can be used for short term forecasting as well as data imputations. However, the modeling process for forecasting and imputations needs to be different for a number of reasons. The short term forecasting problem usually has a fixed prediction window. For example, prediction of electricity demand for next two hours, which can help in resource planning in a power distribution

B. Prasad (Ed.): Soft Computing Applications in Industry, STUDFUZZ 226, pp. 151–163, 2008.
springerlink.com

company. The failure of data recorders, on the other hand, can be of variable lengths. The data imputation process needs to be prepared for prediction of data for a variable time period. Another important difference between forecasting and imputations is the available data. The forecasting involves prediction of future values based on past values in real-time, whereas the data imputation models have access to the data from before and after the failure. The data imputation is usually a necessary step prior to the offline data mining analysis.

This chapter desribes how the techniques described above, namely, GAs, LWR, and TDNN can be used to develop sophisticated models for estimating missing data. The hourly volumes of traffic counts are used to test the effectiveness of these models. Traffic volume is one of the most useful traffic data. Highway agencies usually employ permanent traffic counters, seasonal traffic counters, and short-period traffic counters to collect traffic volume data. Permanent traffic counters are used to monitor the traffic for 24 hours per day, 365 days per year. Seasonal traffic counters and short-period traffic counters are used to monitor the traffic for a short period ranging from a few hours to a few days (Garber and Hoel 1988). Accurate estimation of missing hourly volumes is of importance for highway agencies to carry out traffic analysis at both operation and planning level.

2 Review of Literature

Many researchers have studied the problem of missing values (Beveridge 1992; Bole et al. 1990; Gupta and Lam 1996; Little and Rubin 1987; Singh and Harmancioglu 1996; Wright 1993; Rubin, 1987, 1996; Schafer, 1997, 1999; Scahfer and Olsen, 1998). Ratner (2003) reported that there were four popular imputation methods for updating missing values:

1. Deductive Imputation. Missing values are deduced with certainty, or with high probability, from other information on the case.
2. Hot-deck Imputation. For each imputation class, missing values are replaced with values from "closest matching" individuals.
3. Mean-value Imputation. The sample mean of each imputation class is used to fill in missing values.
4. Regression-based Imputation. Missing values are replaced by the predicted values from a regression analysis.

This study focuses on neural and regressive imputations of missing values in a traffic data time series.

2.1 Locally Weighted Regressive Analysis

Regression analysis consists of graphic and analytic methods for exploring relationships between one variable, referred to as a response variable or dependent variable, and one or more other variables, called predictor variables or independent variables. Regression analysis is distinguished from other types of statistical analyses in that the goal is to express the response variable as a function of the predictor variables. Once such an expression is obtained, the relationship can be utilized to predict values of the response variable, identify which variables most affect the response, or verify hypothesized causal models of the response (Mendenhall and Sincich 1995). The regression finds a linear combination of input variables to minimize the sum of square errors between observed and predicted values.

A variation of regression analysis called locally weighted regression (LWR) is used in this study. Locally weighted regression is a form of instance-based (or memory-based) algorithm for learning continuous mappings from real-valued input vectors to real-valued output vectors. In locally weighted regression, points are weighted by proximity to the current **x** in question using a kernel. Local methods assign a weight to each training observation that regulates its influence on the training process. The weight depends upon the location of the training point in the input variable space relative to that of the point to be predicted. Training observations closer to the prediction point generally receive higher weights (Friedman 1995). A regression is then computed using the weighted points. This is achieved by weighting each data point according to its distance to the query point: a point very close the query gets a weight of one, and a point far away gets a weight of zero.

Locally weighted regression has been increasingly used in control and prediction. Zografski and Durrani (1995) explored the use of locally weighted regression in robot control and modeling time series, and also compared it to neural networks and other methods. Gorinevsky and Connolly (1994) compared several different approximation schemes, such as neural nets, Kohonen maps, radial basis functions, to local polynomical fits on simulated robot inverse kinematics with added noise. They found that local polynomical fits were more accurate than all other methods. One recent study demonstrated that locally weighted regression was suitable for real-time control by constructing a locally-weighted-regression-based system that learned a difficult juggling task (Schaal and Atkeson 1994).

2.2 Time Delay Neural Network

The variant of neural network used in this study is called time delay neural network (TDNN) (Hecht-Nielsen 1990). Fig. 1 shows an example of a TDNN, which are particularly useful for time series analysis. The neurons in a given layer can receive delayed input from other neurons in the same layer. For example, the network in Fig. 1 receives a single input from the external environment. The remaining nodes in the input layer get their input from the neuron on the left delayed by one time interval. The input layer at any time will hold a part of the time series. Such delays can also be incorporated in other layers. Neurons process input and produce output. Each neuron takes in the output from many other neurons. Actual output from a neuron is calculated using a transfer function. In this study, a sigmoid transfer function is chosen because it produces a continuous value in the range [0,1].

It is necessary to train a neural network model on a set of examples called the training set so that it adapts to the system it is trying to simulate. Supervised learning is the most common form of adaptation. In supervised learning, the correct output for the output layer is known. Output neurons are told what the ideal response to input signals should be. In the training phase, the network constructs an internal representation

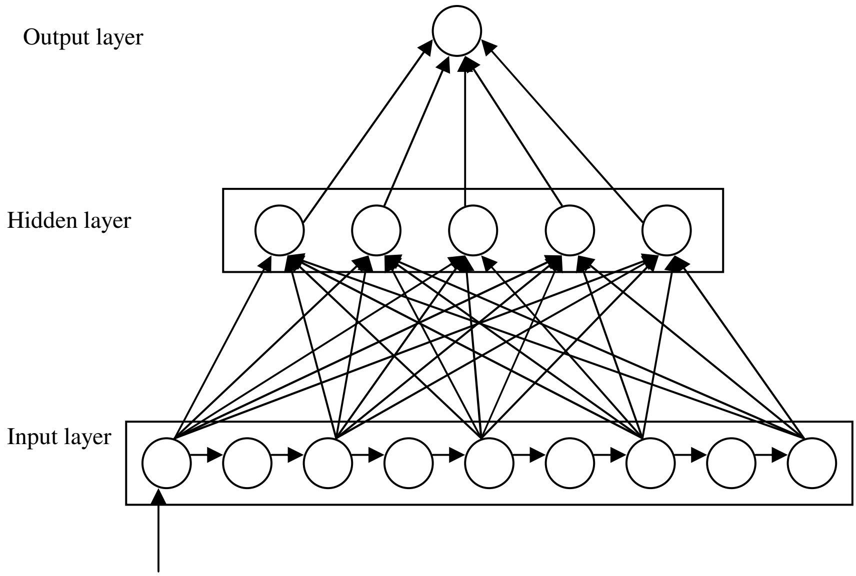

Fig. 1. Time Delay Neural Network (TDNN)

that captures the regularities of the data in a distributed and generalized way. The network attempts to adjust the weights of connections between neurons to produce the desired output. The backpropagation method is used to adjust the weights, in which errors from the output are fed back through the network, altering weights to prevent the repetition of the error.

2.3 Genetic Algorithms

Genetic algorithms are heuristic searching algorithms for finding near optimum solutions from vast possibilities. The origin of genetic algorithms (GAs) is attributed to Holland's work (Holland 1975) on cellular automata. There has been significant interest in GAs over the last two decades (Buckles and Petry 1994). The range of applications of GAs includes such diverse areas as: job shop scheduling, training neural networks, image feature extraction, and image feature identification. Previous research shows that GAs consistently outperform both classical gradient search techniques and various forms of random search on more difficult problems, such as optimizations involving discontinuous, noisy, high-dimensional, and multimodal objective functions (Grefenstette 1986).

The genetic algorithm is a model of machine learning, which derives its behavior from a metaphor of the processes of evolution in nature. In practice, the genetic model of computation can be implemented by having arrays of bits or characters to represent the chromosomes $(c_i \mid 1 \le i \le n)$, where c_i is called a *gene*. Simple bit manipulation operations allow the implementation of crossover, mutation and other operations. The crossover operator creates new individuals called *offspring*, by recombining the genetic material of two individuals, deemed the *parents*. Individuals with higher fitness scores are selected with greater probability to be parents and “pass on” their genes to the next generation. The mutation operator randomly alters one or more genes in an individual. Mutations add genetic diversity to the population.

The GAs attempt to construct a good individual by starting with a population of randomly generated individuals. From there on, the genetic operations, in concert with the fitness measure, operate to improve the population.

2.4 GAs for Designing Neural Network and Regression Models

Many researchers have used GAs to determine neural network architectures. Harp, et al. (1989) and Miller, et al. (1989) used GAs to determine the best connections among network units. Montana and Davis (1989) used GAs for training the neural networks. Chalmers (1991) developed learning rules for neural networks using GAs.

Hansen, et al. (1999) used GAs to design time delay neural networks (TDNN), which included the determination of important features such as the number of inputs, the number of hidden layers, and the number of hidden neurons in each hidden layer. Hansen, et al. (1999) applied their networks to model chemical process concentration, chemical process temperatures, and Wolfer sunspot numbers. Their results clearly showed advantages of using GAs configured TDNN over other techniques including conventional autoregressive integrated moving average (ARIMA) methodology as described by Box and Jenkins (1970).

Hansen et al.’s approach (1999) consisted of building neural networks based on the architectures indicated by the fittest chromosome. The objective of the evolution was to minimize the training error. Such an approach is computationally expensive. Another possibility that is used in this study is to choose the architecture of the input layer using genetic algorithms.

Time series modeling is based on the assumption that the historical values of a variable provide an indication of its value in the future. Based on the available information and pattern analysis, it is possible to make a reasonable assumption about the appropriate length of the history that may be useful for predicting the value of a variable. If all the historical values in the input layer were used in TDNN calculations, it would lead to an unwieldy network. Such a complex network may hinder the training process. A solution to such a problem is selective pruning of network connections. Lingras and Mountford (2001) proposed the maximization of correlation between input variables and the output variable as the objective for selecting the connections between input and hidden layers. Since such an optimization is not computationally feasible for large input layers, GAs were used to search for a near optimal solution. It should be noted here that since the input layer has a section of time series, it is not possible to eliminate intermediate input neurons. They are necessary to preserve their time delay connections. However, it is possible to eliminate their feedforward connections. Lingras and Mountford (2001) achieved superior performance using the GAs designed neural networks for the prediction of inter-city traffic.

Lingras et al. (2002) also showed that GAs designed regression models had more accurate short-term traffic predictions than simple regression models on a recreational road. For example, the average errors for simple regression models used by Kalyar (1998) are usually more than 4% in the training sets, whereas those for genetically

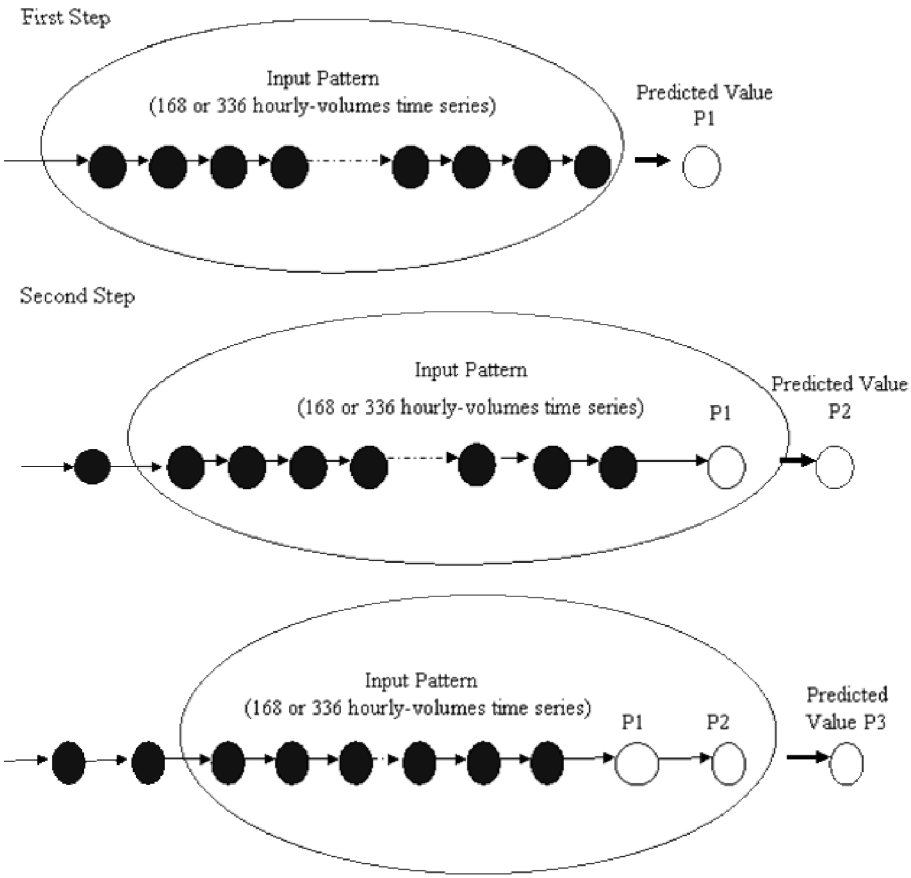

Fig. 2. Prototype of genetically designed models

designed regression models are less than 0.6%. For test sets, the 95th percentile errors for normal regression models are usually more than 30%. The errors for some hours are as high as 54%. However, for genetically designed regression models, they are all lower than 8% and most of them are less than 4%.

The present study summarizes the extensive experimentation reported by Zhong (2003). It uses the same objective function for the development of regression and neural network models. These models are enhanced to estimate successive missing hourly volumes for 12 hours and up to a week (168 hours). The models are improved to be able to use the data from both before and after the failure period. The developed models are used to update missing values of traffic counts.

3 The Prototype of Genetically Designed Models

Two types of genetically designed models are developed in this study. The first type is short-term prediction models, which only use the data from before the failure as the input. For this type of models, a week-long ($7 \times 24 = 168$) hourly volumes before the first missing value are used as the candidate inputs. The second type is both-side models, which use the data from both before and after the failure as the input. For models using the data from before and after the failure, one week-long hourly volumes from each side of the occurrence of missing value(s) are used as the candidate inputs. Totally $168 \times 2 = 336$ hourly volumes are presented to GAs for selecting final inputs. The number of final input variables is decided to be 24 based on experiments with different numbers. For neural network models, there are 168 or 336 neurons in the input layer, but only 24 of them have the connections with hidden layer. In addition, there are 12 neurons in the hidden layer and 1 neuron in the output layer. For regression models, there are 168 or 336 independent variables. However, only 24 of them are selected by GAs and used to predict dependent variable.

Fig. 2 shows the prototype of genetically designed models used in this study. First, assuming there is only one missing value P_1 in traffic counts, candidate inputs of models are presented to GAs for selecting 24 final input variables. These 24 hourly volumes are chosen because they have the maximum correlation with the traffic volume of the missing hour, among all the combinations of 24 variables from candidate inputs. The GAs selected variables are used to train neural network and regression models for traffic prediction for the missing hour. The trained neural network or regression models are used to estimate the missing hourly volume P_1. If there were more than one successive missing value, same techniques would be used to predict missing value of the next hour P_2. However, at this stage, the candidate pattern presented to GAs for selecting final inputs includes estimated volume of the first hour P_1, as shown in Fig.2. P_1 may or may not be chosen as a final input because there are different input selection schemas for different hourly models. Fig. 3 shows a TDNN model with inputs selected from a week-long hourly-volume time series. Corresponding regression model also used same inputs for prediction.

A top-down design was used to search for the models with reasonable accuracy. First 24-hour universal models were established to test their ability, then they were further split into 12-hour universal models, single-hour models, seasonal single-hour models, day-hour models, and seasonal day-hour models. The characteristics of these models are as follows:

1. 24-hour universal models: This approach involved a single TDNN and a single regression model for all the hours across the year. The advantage of such models is the simplicity of implementation during the operational phase.
2. 12-hour universal models: In order to improve models' accuracy, 12-hour universal models were developed based on 12-hour (from 8:00 a.m. to 8:00 p.m.) observations. In other words, the observations from late evenings and early mornings were eliminated from the models.
3. Single-hour models: 12-hour universal models were further split into 12 single-hour models. That is, every hour had a separate model.
4. Seasonal single-hour models: Seasons have definite impact on the travel. Further dividing single-hour models into seasonal models may improve models' accuracy. In this study, yearly single-hour models were further split into May-August single-hour models and July-August single-hour models.

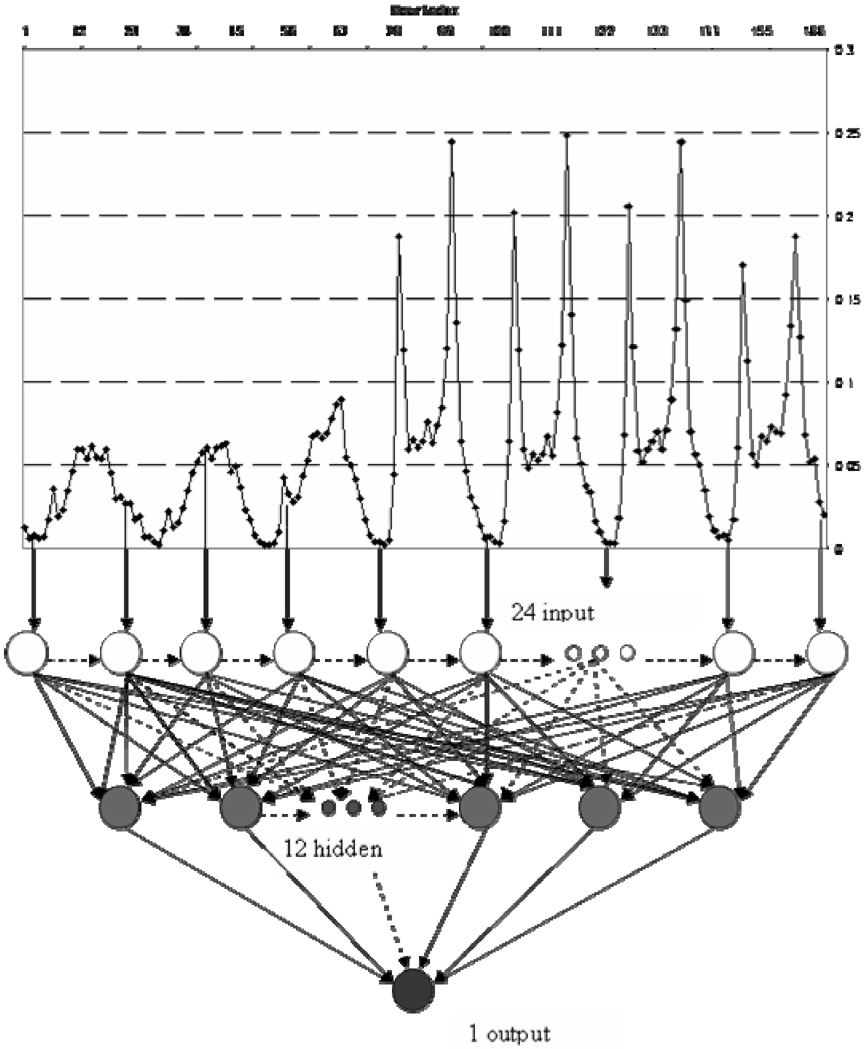

Fig. 3. TDNN model used for prediction

5. Day-hour models: Travel patterns also vary by the day of the week. Further classification of observations into groups for different days (e.g., Wednesday or Saturday) may improve models' accuracy.
6. Seasonal day-hour models. Day-hour models were further split into seasonal day-hour models (e.g., July-August day-hour models) to explore the models.

4 Experiments

Two large traffic volume databases are used to discover the nature of missing values. The missing values discussed in this study are restricted to missing hourly volumes.

The analyses show that these data are stored in various formats and have significant missing portions. The data are cleaned and integrated into a data warehouse. The 1998 data from Alberta are used to study the efficiency distribution of traffic counting program in detail. Cluster analysis is used to classify complete traffic counts into trip pattern groups. Sample counts from different pattern groups and functional classes are selected for testing imputation models proposed in this chapter.

4.1 The Nature of Missing Values

The data sets from two highway agencies in North America are used in this study. The data set from the Alberta Transportation spans 17 years, and the other from the Minnesota Department of Transportation has one-year data. The data are in the form of hourly volumes for one-way or both directions. Zhong (2003) provide detailed description of the format of the data. The analyses for missing values were applied to these data sets. Missing values are usually represented by zero hourly volumes or blanks in counter files. The machine failures result in blanks in the records. There are many reasons for this, including power surges, lightening, loss of battery power, solar panel failure, vandalism, and environmental fatigue, such as storms and frost heaves. Long-range zero hourly volumes indicate that the connections between sensors and counting machines were cut off, while the machines were still working normally. Lightening is a major reason for this kind of disconnection.

Analysis of missing data for Alberta indicated that large numbers of permanent traffic counts (PTCs) had missing values. Over seven years, more than half of total counts have missing values. For some years, the percentage is as high as 70% to 90%. Minnesota data also showed more than 40% counts having missing values. Detailed analyses of the missing data can be found in (Zhong 2003). The analysis clearly shows the extent of missing values and a need for data imputation.

The PTCs were clustered into various categories based on the predominant traffic on the corresponding highway sections. Six PTCs were selected from 4 out of 5 groups: two from the commuter group, two from the regional commuter group, one from the rural long-distance group, and one from the summer recreational group. Due to insufficient data, no counts were selected from the winter recreational group. These counts were used to test the accuracy of imputation models described earlier.

4.2 Analysis of Results

The performance of various models used in this study is expressed in terms of absolute percentage errors (APE). Previously, Sharma et al. (1999) used APE to evaluate the performance of neural networks and factor methods for estimating AADTs from short-period traffic counts. Lingras et al. (2002) used APE to assess the accuracy of genetically designed regression and neural network models for forecasting short-term traffic. The APE used in this chapter is defined as:

$$APE = \frac{|actual \quad volume - estimated \quad volume|}{actual \quad volume} \times 100 \tag{1}$$

The key evaluation parameters consist of the average, 50^{th}, 85^{th}, and 95^{th} percentile errors. These statistics give a reasonable idea of the error distributions by including (e.g., when calculating average errors) or excluding (e.g., when calculating the percentile errors) large errors caused by outliers or special events.

A detailed analysis of results for all the counters is described in Zhong (2003). Due to space limitations we only describe the summary of results and detailed reauslts for a regional commuter highway section in Alberta in this chapter.

It was found that the regression models tended to perform better than the TDNN models. Among the six categories of models described in the previous section, the refined models in the hierarchy tended to perform better. For example, the seasonal day hour models performed the best, while the 24 hour universal model performed the worst. The prediction accuracy for stable traffic patterns such as commuter traffic was higher than that for more variable recreational traffic patterns. As expected, models that were based on data from before and after the failure performed better than the models that only used the historical data. Table 1 shows an example of the prediction errors for updating 12 successive missing values with July-August day-hour models for regional commuter count C022161t. First column in the table shows the hour of the day. The remaining columns give various prediction errors. The column entitled "B" correspond to the data only from before the failure, while those entitled "B-A" correspond to the data from before and after the failure. It is clear that in this case the use of data from before and after the failure is advantageous. It is also interesting to note that the overall errors are very small. For example, the average errors for models based on data from before and after failure range from 0.8 to 2.1%. Even the 95th percentile errors are fairly low, ranging from 1.5% to 5.7%. That means less than 5% of the errors will be more than 6%.

Table 1. Errors for imputing 12 successive missing traffic volumes with GA-LWR

Hour	**Prediction Errors**							
	Average		**50th %**		**85th %**		**95th %**	
(1)	**B** (2)	**B-A** (3)	**B** (4)	**B-A** (5)	**B** (6)	**B-A** (7)	**B** (8)	**B-A** (9)
07-08	1.66	1.20	1.59	0.86	2.51	2.00	4.18	2.51
08-09	3.26	1.49	2.37	1.32	5.33	2.88	9.05	3.05
09-10	2.03	1.24	2.25	1.23	2.92	1.91	2.94	2.23
10-11	1.93	1.36	1.62	1.40	3.28	196	3.61	2.83
11-12	1.29	0.82	0.80	0.37	2.71	1.79	3.00	2.24
12-13	2.76	2.11	2.82	1.74	3.99	3.08	4.97	5.66
13-14	2.83	0.83	2.68	0.75	4.38	1.44	4.80	1.79
14-15	1.82	1.54	1.64	1.66	2.76	1.81	4.38	3.06
15-16	0.82	1.15	0.73	0.66	1.37	1.78	1.70	3.00
16-17	2.12	1.91	1.69	1.55	3.53	2.79	4.33	3.69
17-18	1.47	0.82	1.49	0.80	2.66	1.14	2.95	1.49
18-19	2.91	1.00	2.71	1.14	5.95	1.71	6.68	1.82

5 Summary and Conclusions

In order to make meaningful use of the available data, it may be necessary to provide a reasonably accurate estimation of missing data. Otherwise, it will be difficult to apply many of the data mining and knowledge discovery models. The analysis of traffic data (used in this study for illustration) shows that a significant number of data sets have some missing values.

While the techniques for short-term forecasting can be adapted for data imputation of missing values, there are inherent differences between the two problems. First of all, the duration for which the estimations are needed is variable in imputations, but fixed for forecasting. This fact makes the imputation problem a little more complicated. This chapter shows how imputed data values can be used for the estimation of the subsequent values. The second important difference between forecasting and imputation is the availability of data. The forecasting is limited by the fact that only historical data can be used for estimating future values. The accuracy of imputations, on the other hand, can be improved by the use of data from before and after failures.

The chapter describes how genetically designed regression and neural network models can be used for data imputation. The extensive study reported in detail in Zhong (2003) used a top down approach to improve generic universal models to more refined models specific to a particular time period. The results show that the genetically designed regression models outperform the genetically designed neural network models. The difference between the results is especially significant for stable temporal patterns from commuter highways as opposed to more variable recreational highways. It is possible that the neural networks may perform better for noisy and large data sets.

References

Allison, P.D.: Missing data. Sage Publications, Thousand Oaks (2001)

Atkeson, C., Moore, A., Schaal, S.: Locally Weighted Learning for Control. Artificial Intelligence Review 11, 75–113 (1997)

Beveridge, S.: Least Squares Estimation of Missing Values in Time Series. Communications in Statistics: Theory and Methods 21(12), 3479–3496 (1992)

Bole, V., Cepar, D., Radalj, Z.: Estimating Missing Values in Time Series. Methods of Operations Research 62 (1990)

Box, G., Jenkins, J.: Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco (1970)

Buckles, B.P., Petry, F.E.: Genetic Algorithms. IEEE Computer Press, Los Alamitos (1994)

Cabena, P., et al.: Discovering Data Mining: From Concept to Implementation. Prentice Hall, New Jersey (1998)

Chalmers, D.: The Evolution of Learning: An Experiment in Genetic Connectionism. In: Touretzky, D., et al. (eds.) Connectionist Models: Proceedings of the 1990 Summer School, pp. 81–90. Morgan Kaufmann, San Mateo (1991)

Friedman, J.H.: Intelligent Local Learning for Prediction in High Dimensions. In: International Conference on Artificial Neural Networks, Paris, France (1995)

Garber, N.J., Hoel, L.A.: Traffic and Highway Engineering. West Publishing Company, New York (1998)

Grefenstette, J.: Optimization of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man, and Cybernetics 16(1), 122–128 (1986)

Gorinevsky, D., Connolly, T.H.: Comparison of Some Neural Network and Scatter Data Approximations: The Inverse Manipulator Kinematics Example. Neural Computation 6, 521–542 (1994)

Gupta, A., Lam, M.S.: Estimating Missing Values using Neural Networks. Journal of the Operational Research Society 47(2), 229–239 (1996)

Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco (2001)

Harp, S., Samad, T., Guha, A.: Towards the Genetic Synthesis of Neural Networks. In: Shaffer, D. (ed.) Proceedings of the Third International Conference on Genetic Algorithms, pp. 360–369. Morgan Kaufmann, San Mateo (1989)

Hansen, J.V., McDonald, J.B., Nelson, R.D.: Time Series Prediction with Genetic Algorithm Designed Neural Networks: An Experimental Comparison with Modern Statistical Models. Computational Intelligence 15(3), 171–184 (1999)

Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)

Kalyar, I.: Prediction of Sunday Afternoon Traffic Using Neural Network and Regression Analysis. Unpublished M.A.Sc. thesis. Faculty of Engineering, University of Regina, Regina, Saskatchewan, Canada (1998)

Lawrence, J.: Introduction to Neural Networks: Design, Theory, and Applications. California Scientific Software Press, Nevada City (1993)

Lingras, P.J., Mountford, P.: Time Delay Neural Networks Designed Using Genetic Algorithms for Short Term Inter-City Traffic Forecasting. In: Proceedings of the Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Budapest, Hungary, pp. 292–299 (2001)

Lingras, P.J., Sharma, S.C., Zhong, M.: Prediction of Recreational Travel using Genetically Designed Regression and Time Delay Neural Network Models. Transportation Research Record 1805, pp. 16–24 (2002)

Little, R.J.A., Rubin, D.B.: Statistical analysis with missing data. John Wiley & Sons, New York (1987)

Mendenhall, W., Sincich, T.: Statistics for Engineering and Science. Prentice-Hall, NY (1995)

Miller, G., Todd, P., Hedge, S.: Designing Neural Networks using Genetic Algorithms. In: Shaffer, D. (ed.) Proceedings of the Third International Conference on Genetic Algorithms, pp. 53–60. Morgan Kaufmann, San Mateo (1989)

Mitchell, M.: An Introduction to Genetic Algorithms. The MIT Press, Boston (1999)

Montana, D., Davis, L.: Training Feedforward Networks using Genetic Algorithms. In: Sridhara, N. (ed.) Proceedings of Eleventh International Joint Conference on Artificial Intelligence, Detroit, Michigan, pp. 762–767 (1989)

Pickles, A.: Missing data, problems and solutions. In: Kempf-Leonard, K. (ed.) Encyclopedia of social measurement, pp. 689–694. Elsevier, Amsterdam (2005)

Ratner, B.: CHAID as a Method for Filling in Missing Values. In: Statistical Modeling and Analysis for Database Marketing: Effective Techniques for Mining Big Data, Chapmall & Hall/CRC Press (2003)

Rubin, D.B.: Multiple imputation for nonresponse in surveys. John Wiley & Sons, New York (1987)

Rubin, D.B.: Multiple imputation after 18+ years. Journal of the American Statistical Association 91, 473–489 (1996)

Schaal, S., Atkeson, C.: Robot Juggling: An Implementation of Memory-based Learning. Control Systems 14, 57–71 (1994)
Schafer, J.L.: Analysis of incomplete multivariate data. Book No. 72, 72th edn. Chapman & Hall series Monographs on Statistics and Applied Probability. Chapman & Hall, London (1997)
Schafer, J.L.: Multiple imputation: A primer. Statistical Methods in Medical Research 8, 3–15 (1999)
Schafer, J.L., Olsen, M.K.: Multiple imputation for multivariate missing-data problems: A data analyst's perspective. Multivariate Behavioral Research 33, 545–571 (1998)
Singh, V.P., Harmancioglu, N.B.: Estimation of Missing Values with Use of Entropy. In: NATO Advanced Research Workshop, Izmir, Turkey (1996)
Wright, P.M.: Filling in the Blanks: Multiple Imputation for Replacing Missing Values in Survey Data. In: SAS 18th Annual Conference, New York (1993)
Zhong, M.: Data mining applications for updating missing values of traffic counts. Unpublished Ph.D. thesis. Faculty of Engineering, University of Regina, Regina, Saskatchewan, Canada (2003)
Zografski, Z., Durrani, T.: Comparing Predictions from Neural Networks and Memory-based Learning. In: Proceedings of ICANN '95/NEURONIMES '95: International Conference on Artificial Neural Networks, Paris, France, pp. 211–226 (1995)