Missing data: A comparison of neural network and expectation maximization techniques

Fulufhelo V. Nelwamondo*, Shakir Mohamed and Tshilidzi Marwala

School of Electrical and Information Engineering, University of the Witwatersrand, Private Bag 3, Wits, 2050, South Africa

Two techniques have emerged from the recent literature as candidate solutions to the problem of missing data imputation. These are the expectation maximization (EM) algorithm and the auto-associative neural network and genetic algorithm (GA) combination. Both these techniques have been discussed individually and their merits discussed at length in the available literature. However, they have not been compared with each other. This article provides a comparison of the two techniques using datasets of an industrial power plant, an industrial winding process and HIV seroprevalence survey data. Results show that the EM algorithm is more suitable and performs better in cases where there is little or no interdependency between the input variables, whereas the auto-associative neural network and GA combination is suitable when there are inherent nonlinear relationships between some of the given variables.

Keywords: Expectation maximization algorithm, genetic algorithm, missing data, neural network.

DATABASES such as those which store measurement or medical data may be subjected to missing values either in the data acquisition or data-storage process. Problems in a sensor, a break in the data transmission line or non-response to questions posed in a questionnaire are prime examples of how data can go missing. The problem of missing data poses difficulty in the analysis and decision-making processes which depend on these data, requiring methods of estimation that are accurate and efficient. Various techniques exist as a solution to this problem, ranging from data deletion to methods employing statistical and artificial intelligence techniques to impute for missing variables. However, some statistical methods, like mean substitution have a high likelihood of producing biased estimates[1] or make assumptions about the data that may not be true, affecting the quality of decisions made based on the data.

The estimation of missing input vector elements in real-time processing applications requires a system that possesses the knowledge of certain characteristics such as correlations between variables, which are inherent in the input space. Computational intelligence techniques and maximum likelihood techniques do possess such characteristics and thus are important for imputation of missing data. This article compares two approaches to the problem of missing data estimation. The first technique is based on the current state-of-the-art approach to this problem, i.e. the use of maximum likelihood (ML) and expectation maximization (EM)[2]. The second approach is the use of a system based on auto-associative neural network and genetic algorithm (GA)[3]. The estimation ability of both of these techniques is compared, based on three datasets and conclusions are made.

*For correspondence. (e-mail: f.nelwamondo@ee.wits.ac.za)

Background

Missing data

Real time processing applications that are highly dependent on data often suffer from the problem of missing input variables. Various heuristics of missing data imputation such as mean substitution and hot deck imputation also depend on the knowledge of how datapoints become missing. There are several reasons why the data may be missing. As a result, missing data may follow an observable pattern. Exploring the pattern is important and may lead to the possibility of identifying cases and variables that affect the missing data[4]. Having identified the variables that predict the pattern, a proper estimation method can be selected.

According to Little and Rubin[5], and Burk[6], there are three types of missing data mechanisms. These can be distinguished by considering Figure 1, which shows a data pattern with variables $X = \{X_1, X_2, \ldots, X_p\}$, and Y which has some missing data. Variables X and Y represent columns of a table in the database, with Y being the column with missing data. Considering X and Y as random variables, the three categories of missing data are as follows:

Missing completely at random (MCAR): This occurs if the missing value for the input vector does not depend on any other variable in the database, such that inputs with missing entries are the same as the complete inputs. That is, the probability of data Y being missing is not dependent on either X or Y, i.e. is not dependent on either missing or complete values in the same record or any other record in the database.

Missing at random (MAR): This occurs if the missing value for the input vector depends on other variables in the dataset, such that the pattern in which the data becomes missing is traceable. That is, the probability of data Y being missing is dependent only on X, the existing values in the database and not on any missing data.

Missing not at random (MNAR): This occurs when the missing value for the input vector depends on the other missing values, such that the existing data in the database cannot be used to approximate the missing values. This is also known as the non-ignorable case. The probability that Y is missing is dependent on the missing data.

This work assumes that data are missing at random. This implies that we expect the missing values to be deducible in some complex manner from the remaining data.

Autoencoder neural networks

Autoencoders, also known as auto-associative neural networks, are neural networks trained to recall the input space. Thompson *et al.*[7] distinguish two primary features of an autoencoder network, namely its auto-associative nature and the presence of a bottleneck that occurs in the hidden layers of the network, resulting in a butterfly-like structure. In cases where it is necessary to recall the input, autoencoders are preferred due to their remarkable ability to learn certain linear and nonlinear interrelationships, such as correlation and covariance inherent in the input space. Autoencoders project the input onto some smaller set by intensively squashing it into smaller details. The optimal number of the hidden nodes of the autoencoder, though dependent on the type of application, must be smaller than that of the input layer[7]. Autoencoders have been used in various applications, including the treatment of missing data problem by a number of researchers[3,8–10].

In this article, autoencoders are constructed using the multi-layer perceptrons (MLP) networks and trained using back-propagation. MLPs are feed-forward neural networks with an architecture comprising the input layer, hidden layer and output layer. Each layer is formed from smaller units known as neurons. Neurons in the input layer receive the input signals $\vec{x}$ and distribute them forward to the network. In the next layers, each neuron receives a signal, which is a weighted sum of the outputs of the nodes in the previous layer. Inside each neuron, an activation function is used to control the input. Such a network determines a nonlinear mapping from an input vector to the output vector, parameterized by a set of network weights, which we refer to as the vector of weights $\vec{W}$. The structure of an autoencoder constructed using an MLP network is shown in Figure 2.

The first step in approximating the weight parameters of the model is finding the approximate architecture of the MLP, where the architecture is characterized by the number of hidden units, the type of activation function, as well as the number of input and output variables. The second step estimates the weight parameters using the training set[11]. Training estimates the weight vector $\vec{W}$ to ensure that the output is as close to the target vector as possible. The problem of identifying the weights in the hidden layers is solved by maximizing the probability of the weight parameter using Bayes' rule[7] as follows:

$$P(\vec{W} \mid D) = \frac{P(D|\vec{W})P(\vec{W})}{P(D)}, \tag{1}$$

where D is the training data, $P(\vec{W}|D)$ the posterior probability and $P(D|\vec{W})$ is called the likelihood term that balances between fitting the data well and helps in avoiding overly complex models, whereas $P(\vec{W})$ is the prior probability of $\vec{W}$. $P(D)$ is the evidence term that normalizes the posterior probability. The input is transformed from x to the middle layer, a, using weights W_{ij} and biases b_j as follows[7]:

$$a_j = \sum_{i=1}^{d} W_{ij} x_i + b_j, \tag{2}$$

where $j = 1$ and $j = 2$ represent the first and second layer respectively. The input is further transformed using the activation function such as the hyperbolic tangent (tanh)

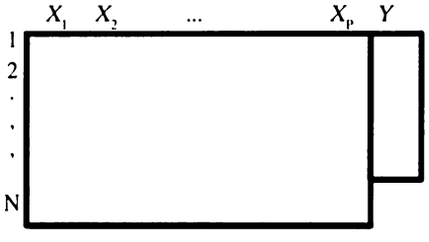

Figure 1. Pattern of missing data in a rectangular dataset. Rows correspond to records in the database and columns correspond to variables or fields of the dataset[2].

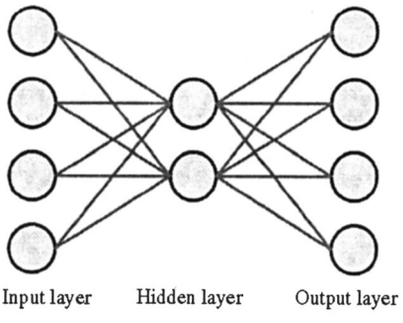

Figure 2. Structure of a four-input four-output autoencoder.

or the sigmoid in the hidden layer. More information on neural networks can be found in Bishop[12].

Genetic algorithms

There are various optimization techniques such as fast simulated annealing, ant colony optimization, GAs and particle swam optimization that are all aimed at optimizing some variables to adhere to some target function. Some of these methods converge at local optimal solutions than the required global optimal solutions. Although stochastic in nature, GAs converge at a global optimal solution. GAs use the concept of survival of the fittest over consecutive generations to solve optimization problems[13]. As in biological evolution, the fitness of each population member in a generation is evaluated to determine whether it will be used in the breeding of the next generation. In creating the next generation, techniques (such as inheritance, mutation, natural selection, and recombination) common in the field of evolutionary biology are employed. The GA implemented in this article uses a population of string chromosomes, which represent a point in the search space[13]. In this article, all these parameters were empirically determined using exhaustive search methods. The GA was implemented by following three main procedures which are selection, crossover and mutation.

Neural networks and GA for missing data

The method used here combines the use of auto-associative neural networks with GA to approximate missing data. This method has been used to approximate missing data in a database by Abdella and Marwala[3]. A GA is used here to estimate the missing values by optimizing an objective function. The complete vector combining the estimated and observed values is fed into the autoencoder as input (Figure 3). Symbols X_k and X_u represent the known and the unknown or missing variables respectively. The combination of X_k and X_u represents the full input space.

Considering that the method uses an autoencoder, one would expect the input to be similar to the output for a well chosen architecture of the autoencoder. This is, however, only expected on a dataset similar to the problem space from which the inter-correlations have been captured. The difference between the target and the actual output is used as the error and this error is defined as follows:

$$\varepsilon = \vec{x} - f(\vec{W}, \vec{x}), \tag{3}$$

where $\vec{x}$ and $\vec{W}$ are input and weight vectors respectively. To make sure that the error function is always positive, the square of the equation is used. This leads to the following equation:

$$\varepsilon = (\vec{w} - f(\vec{W}, \vec{x}))^2. \tag{4}$$

Since the input and output vectors consist of both X_k and X_u entries, the error function can be written as follows:

$$\varepsilon = \left(\begin{Bmatrix} X_k \\ X_u \end{Bmatrix} - f\left(\begin{Bmatrix} X_k \\ X_u \end{Bmatrix}, w \right) \right)^2. \tag{5}$$

Equation (5) is used as the objective function that is minimized using GA in order to estimate X_u.

Maximum likelihood

The maximum likelihood approach to approximating missing data is a popular technique[14,15] and is based on a precise statistical model of the data. The model most commonly used is the multivariate, Gaussian mixture model, while the maximum likelihood method is applied for the task of imputing the missing values. Likelihood methods may be categorized into 'single imputations' and 'multiple imputations'[4,14,15]. Here we consider only single imputations. As a result, the EM algorithm was used.

Background to the EM for missing data

The EM algorithm was originally introduced by Dempster *et al.*[16] in 1977 and was aimed at overcoming problems associated with maximum likelihood methods. The EM algorithm combines statistical methodology with algorithmic implementation and has gained much attention recently in various missing data problems. The algorithm has also been proven to work better than methods such as listwise, pairwise data deletion, and mean substitution, because it assumes that incomplete cases have data missing at random rather than missing completely at random[17,18].

The EM algorithm is a general technique for fitting models to incomplete data. It capitalizes on the relationship between missing data and the unknown parameters of a data model. If we knew the missing values, then estimating the model parameters would be straightforward.

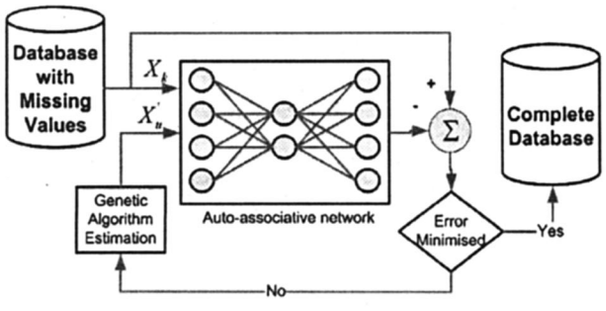

Figure 3. Autoencoder and genetic algorithm-based missing data estimator structure.

Similarly, if we knew the parameters of the data model, then it would be possible to obtain unbiased predictions for the missing values. This interdependence between model parameters and missing values suggests an iterative method where we first predict the missing values based on assumed values for the parameters, use these predictions to update the parameter estimates, and repeat the process. The sequence of parameters converges to maximum likelihood estimates that implicitly average over the distribution of the missing values.

Description of EM for missing data imputation

The EM algorithm is aimed at estimating a parameter θ, such that $P(X|\theta)$ is maximum. The variable X can be defined as a random vector from a parameterized family. To estimate the parameter adequately, a log likelihood function is introduced and the likelihood of θ, denoted as $L(\theta)$ is defined by[16]

$$L(\theta) = \ln P(X|\theta). \tag{6}$$

Due to the ln function, $L(\theta)$ is an ever-increasing function such that the θ that will maximize $L(\theta)$ will also maximize $P(X|\theta)$. As mentioned earlier, the EM algorithm is an iterative procedure aimed at maximizing $L(\theta)$. The iteration is performed such the most recent value of $L(\theta)$ is better than the ones obtained in previous iterations. As a result,

$$L(\theta_n) > L(\theta_{n-1}), \tag{7}$$

where n represents the nth iteration. The algorithm is also aimed at reaching to the maximum value much faster. This can be achieved by maximizing the difference between two consecutive values of $L(\theta)$. The EM algorithm provides a natural framework that enables missing data to be reestimated. The missing data may be viewed as some hidden variables that, if available, would make the maximization process easier. By letting z denote a hidden variable, the maximum probability can be defined as[16,17]:

$$P(X|\theta) = \sum_z P(X|z,\theta)P(z|\theta). \tag{8}$$

The difference, $L(\theta_n) - L(\theta_{n-1})$ can be written as:

$$L(\theta_n) - L(\theta_{n-1}) = \ln\left(\sum_z P(X|z,\theta)P(z|\theta)\right) - \ln P(X|\theta_{n-1}). \tag{9}$$

It can be shown mathematically that

$$L(\theta_{n-1}) \geq L(\theta_{n-1}) + \delta, \tag{10}$$

where δ is some difference between the current and the previous likelihoods that have been observed. More formally, the E-step is aimed at determining the conditional expectation $E_{z|X,\theta_n}\{\ln P(X,z|\theta\}$, whereas the maximization step maximizes this expression with respect to θ.

Experimental evaluation

Data analysis

The EM and neural network coupled with GA approaches for approximating missing data were compared in three different datasets. In both models under investigation in this article, model parameters were obtained based on the training dataset. Missing data were introduced by randomly removing some data and the model was tested to determine how well the missing data are predicted. The datasets used are briefly described as follows:

Power plant data: The first dataset used was the data of a 120 MW power plant in France[19], under normal operating conditions. This dataset comprises five inputs, namely gas flow, turbine valves opening, super heater spray flow, gas dampers and air flow. Sampling of the data was done every 1228.8 s and a total of 200 instances were recorded. An extract of the data without any missing values is shown in Table 1.

The data were split into training and testing datasets. Due to the limited data available, one-seventh of the data was kept as the test set, with the remaining being considered for training. For easy comparison with the neural network and GA combination, the training and testing data for the EM were combined into a single file, with the testing data appearing at the end of the file. This separation ensured that both the EM, and the neural network and GA combination approach testing were compared using the same amount of testing data and that their respective models built from the same amount of 'training' data. The data were transformed using a min–max normalization to [0, 1] before use, to ensure that they fall within the active range of the activation function of the neural network.

HIV database: The data used in this test were obtained from the South African antenatal sero-prevalence survey

Table 1. Set of power plant measurements under normal operating conditions

Gas flow	Turbine	Heater	Gas dampers	Air flow
0.11846	0.089431	0.11387	0.6261	0.076995
0.10859	0.082462	0.11284	0.6261	0.015023
0.099704	0.19919	0.14079	0.62232	0.061972
0.092794	0.19164	0.12733	0.6261	0.059155
0.0888845	0.30023	0.13768	0.6261	0.028169
0.0875858	0.63182	0.074834	0.63052	0.079812

of 2001. The data for this survey were obtained from a questionnaire answered by pregnant women visiting selected public clinics in South Africa. Only women participating for the first time in the survey were eligible to answer the questionnaire.

Data attributes used in this study were HIV status, education level, gravidity, parity, age group and age gap. The HIV status is indicated in a binary form, where 0 and 1 represent negative and positive respectively. The education level was measured using integers representing the highest grade successfully completed, with 13 representing tertiary education. Gravidity is the number of pregnancies, complete or incomplete, experienced by a female, and this variable is represented by an integer between 0 and 11. Parity is the number of times the individual has given birth and multiple births are considered as one birth event. Both parity and gravidity are important, as they show the reproductive activity as well as the reproductive health state of a woman. Age gap is a measure of the age difference between the pregnant woman and the prospective father of the child. A sample of this dataset is shown in Table 2. The data consist of 5776 instances and were divided into two subsets, namely, training and testing datasets as training was done in the Bayesian framework. Testing was done with 776 instances.

Data from an industrial winding process: The third dataset used here represents a test set-up of an industrial winding process and the data can be found in De Moor[19]. The main part of the plant is composed of a plastic web that is unwound from the first reel (unwinding reel), goes over the traction reel and is finally rewound on the rewinding reel as shown in Figure 4.

As shown in Figure 4, reels 1 and 3 are coupled with a DC-motor that is controlled with input set-point currents I_1 and I_3. The angular speed of each reel (S_1, S_2 and S_3) and tension in the web between reels 1 and 2 (T_1) and between reels 2 and 3 (T_3) are measured by dynamo tachometers and tension meters. The full dataset has 2500 instances, sampled every 0.1 s. In this study, testing was done with 500 instances while the training and validation sets for the neural network consisted of 1500 and 500 instances respectively. The inputs to the winding system are the angular speed of reel 1 (S_1), reel 2 (S_2), reel 3 (S_3) and the set point current at motor 1 (I_1) and motor 2 (I_3), as shown in Figure 4. A more detailed description of the data can be found in Bastogne *et al.*[20].

Table 2. Extract of HIV database used, without missing values

HIV	Education level	Gravidity	Parity	Age	Age gap
0	7	10	9	35	5
1	10	2	1	20	2
1	10	6	5	40	6
0	5	4	3	25	3

Performance analysis

The effectiveness of the missing data system was evaluated using the correlation coefficient and the relative prediction accuracy. The correlation coefficient will be used as a measure of similarity between the prediction and actual data. The correlation coefficient, r was computed as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x}_i)(\hat{x}_i - \hat{\bar{x}}_i)}{\left[\sum_{i=1}^{n}(x_i - \bar{x}_i)^2 \sum_{i=1}^{n}(x_i - \bar{x}_i)(\hat{x}_i - \hat{\bar{x}}_i)^2\right]^{1/2}}, \quad (11)$$

where $\hat{x}$ represents the approximated data, x the actual data and $\bar{x}$ represents the mean of the data. The relative prediction accuracy is defined as:

$$\text{Error} = \frac{n_r}{N} \times 100\%, \quad (12)$$

where n_r is the number of predictions within a certain tolerance percentage of the missing value. In this article, a tolerance of 10% is used. The 10% was arbitrarily chosen with an assumption that it is the maximum acceptable margin for error in the applications considered. This error analysis can be interpreted as a measure of how many of the missing values are predicted within the tolerance. The tolerance can be made to have any value depending on

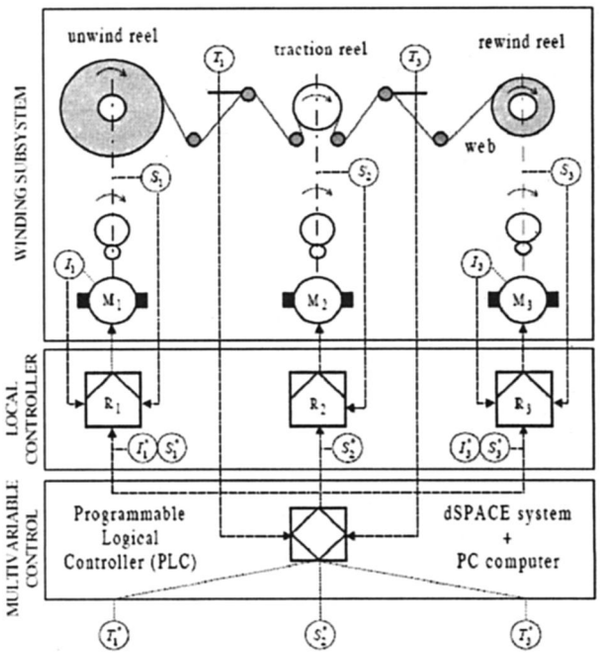

Figure 4. Graphic representation of the winding plot system[20].

Table 3. Results of comparative testing using power plant data

Variable	Correlation		10%	
	Expectation maximization (EM)	Neural network and genetic algorithm (NN–GA)	Expectation maximization (EM)	Neural network and genetic algorithm (NN–GA)
Gas flow	–	0.9790	–	21.43
Turbine	0.7116	0.8061	14.29	14.29
Heater	0.7218	0.6920	7.14	28.57
Gas dumper	–0.4861	0.5093	3.57	10.71
Air flow	0.6384	0.8776	10.71	7.14

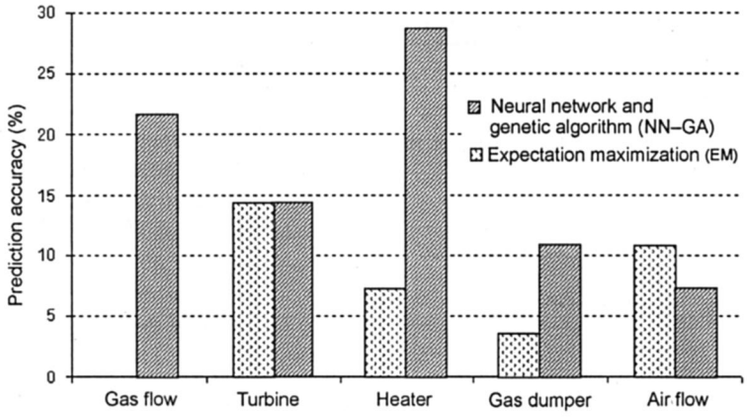

Figure 5. Graphical comparison of estimation accuracy with 10% tolerance using power plant data.

the sensitivity of the application. During evaluation, 2000 values were deleted from one variable and the deleted values were predicted using the proposed methods. This was repeated for all variables in a dataset.

Experimental results and discussion

This section presents the experimental results obtained using both of the approaches described earlier. We evaluate predictability within 10% of the target value. The evaluation was computed by determining how much of the test sample was estimated within the given tolerance. We first present the results of the test done using the power plant dataset.

For the experiment with the power plant data, the neural network and GA system was implemented using an autoencoder network trained with four hidden nodes for 200 training epochs. The GA was implemented using the floating point representation for 30 generations, with 20 chromosomes per generation. Mutation rate was set to a value of 0.1. As mentioned earlier, the GA parameters were empirically determined. The correlation coefficient and the accuracy within 10% of the actual values are given in Table 3.

It can be seen from the results that EM failed to make a prediction for column 1 in this dataset. The reason is that for EM to make a prediction, the prediction matrix needs to be positive definite[21]. The major cause of this is when one variable is linearly dependent on another. This linear dependency may sometimes exist not between the variables themselves, but between elements of moments such as the mean, variance, covariance and correlation[21]. Other reasons include errors while reading the data, initial values, etc. This problem can be solved by deleting variables that are linearly dependent on each other or using principal components to replace a set of collinear variables with orthogonal components. Seminal work on dealing with 'not positive definite matrices' was done by Wothke[21].

The results show that the neural network and GA method is able to impute missing values with higher accuracy of prediction for most cases, and this is shown in Figure 5. The lack of high accuracy predictions for both estimation techniques suggests some degree of difficulty in estimating the missing variables based on the given set of data.

For neural networks, it is observable that the quality of estimation in each input variable depends on the existence of some form of correlation between variables in the input space, such that this linear or nonlinear relationship can be discovered by the neural networks and used to give higher accuracy imputations. The EM algorithm also requires that the data not to be linearly dependent on some variables within the input space, as demonstrated by the need for positive definite matrices. Before commenc-

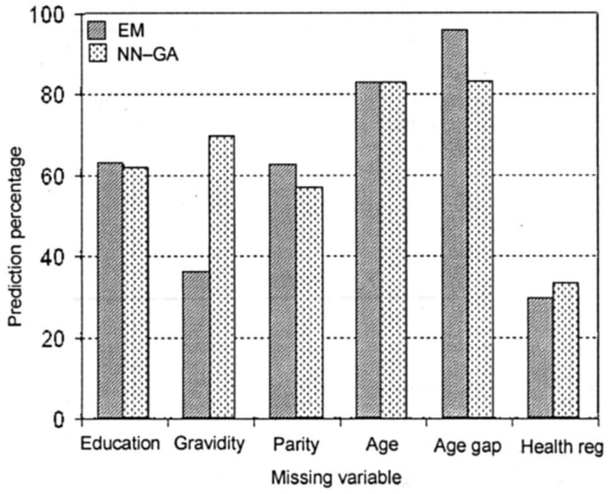

Figure 6. Prediction within 10% tolerance of missing variable in the HIV data.

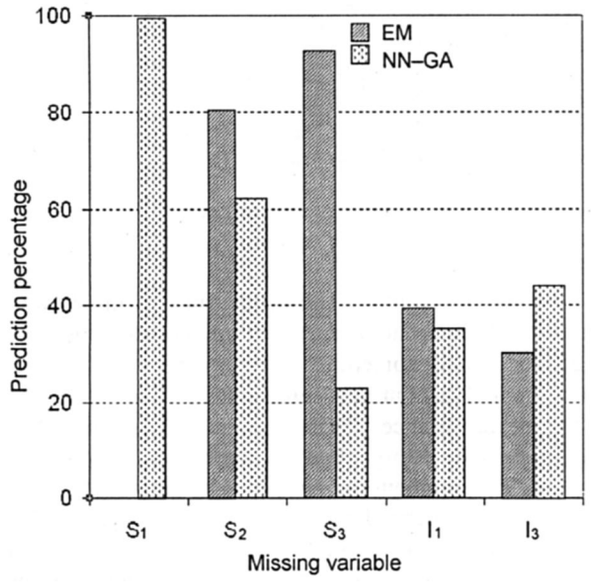

Figure 7. Prediction within 10% tolerance of missing variable in the industrial winding process.

Table 4. Correlation coefficients between actual and predicted data for the HIV database

	Education level	Gravidity	Parity	Age	Age gap
NN–GA	0.10	0.71	0.67	0.99	0.99
EM	0.12	0.21	0.91	0.99	1

Table 5. Correlation coefficients between actual and predicted data for the winding process database

	S_1	S_2	S_3	I_1	I_2
NN–GA	0.203	0.229	0.159	0.038	0.117
EM	–	–0.003	0.009	–0.05	–0.0007

ing the experiment, data were tested for correlation. This testing involved finding out if any variable in the data is somehow strongly related to any other variable in the data.

The results obtained using the HIV database are presented below. Table 4 presents the correlation coefficients between the actual and the predicted data whereas Figure 6 shows the results obtained when predicting missing variables on the HIV dataset within 10% tolerance. Results here clearly show that EM algorithm performs better than the neural network and GA combination algorithm method for the prediction of variables such as education, parity, age and age gap. Unlike the power plant database, results here show that the EM algorithm is better for prediction of variables in the HIV dataset in this study. Since this is a social science database, the reason for poor performance of the neural network and GA combination can be either that the variables are not sufficiently representative of the problem space to produce an accurate imputation model, or that people were not honest in answering the questionnaire, leading to less dependability of variables on each other.

We lastly show the results obtained from the industrial winding process. The EM and neural network and GA combination approaches were compared and the results are shown in Figure 7. For some variables the EM algorithm produces better imputed results, while in others the neural network and GA combination system was able to produce a better imputation results. From the observed data the predicted values were not correlated to the actual missing variables. The possible explanation to this is that the missing data are not interdependent on themselves, but to other variables in the data. Table 5 shows the correlation coefficients. As for the other datasets, the problem of the non-positive definite matrix when imputing values for column 1 prevented the EM algorithm from being used to estimate the missing data.

Conclusion

We have studied and compared the maximum likelihood approach with the neural network and GA combination approach for missing data approximation. In one approach, an auto-associative neural network was trained to predict its own input space. GAs were used to approximate the missing data. On the other hand, the EM algorithm was implemented for the same problem. The results show that for some variables the EM algorithm is able to produce better imputation accuracy, while for the other variables the neural network and GA system is better. Thus the imputation ability of one method over another seems highly problem-dependent. Findings also showed that the EM method seems to perform better in cases where there is little dependency among the variables, which is contrary to the performance of the neural network approach.

1. Tremp, V., Neuneier, R. and Ahmad, S., Efficient methods of dealing with missing data in supervised learning. In *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, 1995, vol. 7.
2. Schafer, J. L. and Graham, J. W., Missing data: Our view of the state of the art. *Psychol. Methods*, 2002, **7**, 147–177.
3. Abdella, M. and Marwala, T., The use of genetic algorithms and neural networks to approximate missing data in database. *Comput. Inf.*, 2005, **24**, 577–589.
4. Schafer, J. L., *Analysis of Incomplete Multivariate Data*, Chapman & Hall, New York, 1997.
5. Little, R. J. A. and Rubin, D. B., *Statistical Analysis with Missing Data*, John Wiley, NY, 2000, 2nd edn.
6. Burk, S. F., A method of estimation of missing values in multivariable data suitable for use with an electronic computer. *J. R. Stat. Soc.*, 1960, **B22**, 302–306.
7. Thompson, B. B., Marks, R. J. and Choi, J. J., Implicit learning in autoencoder novelty assessment. In IEEE Proceedings of the 2002 International Joint Conference on Neural Networks, 2002, vol. 3, pp. 2878–2883.
8. Frolov, A., Kartashov, A., Goltsev, A. and Folk, R., Quality and efficiency of retrieval for Willshaw-like auto-associative networks. *Comput. Neural Syst.*, 1995, **6**, 535–549.
9. Dhlamini, S. M., Nelwamondo, F. V. and Marwala, T., Condition monitoring of HV bushings in the presence of missing data using evolutionary computing. *WSEAS Trans. Power Syst.*, 2006, **1**, 296–302.
10. Mohamed, S. and Marwala, T., Neural network based techniques for estimating missing data in databases, 16th Annual Symposium of the Patten Recognition Association of South Africa, Langebaan, 2005, pp. 27–32.
11. Japkowicz, N., Supervised learning with unsupervised output separation. In International Conference on Artificial Intelligence and Soft Computing, 2002, pp. 321–325.
12. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.
13. Goldberg, D.-E., *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley, Reading, Mass, 1989.
14. Little, R. J. A. and Rubin, D. B., *Statistical Analysis with Missing Data*, John Wiley, New York, 1987.
15. Schafer, J. L. and Olsen, M. K., Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate Behav. Res.*, 1998, **33**, 545–571.
16. Dempster, A. P., Laird, N. M. and Rubin, D. B., Maximum likelihood for incomplete data via the EM algorithm. *J. R. Stat. Soc., Ser. B*, 1977, **39**, 1–38.
17. Allison, P. D., *Missing Data: Quantitative Applications in the Social Sciences*, Sage Publications, London, 2002.
18. Rubin, D. B., Multiple imputations in sample surveys – A phenomenological Bayesian approach to nonresponse. In Proceedings of the Survey Research Methods Section, Alexandria, VA, American Statistical Association, 1978, pp. 20–34.
19. De Moor, B. L. R. (ed.), DaISy: Database for the identification of systems. Department of Electrical Engineering, ESAT/SISTA, K. U. Leuven, Belgium; http://www.esat.kuleuven.ac.be/sista/daisy, last accessed on 15 August 2006.
20. Bastogne, T., Noura, H., Richard, A. and Hittinger, J. M., Application of subspace methods to the identification of a winding process. In Proceedings of the 4th European Control Conference, Brussels, 2002, vol. 5.
21. Wothke, W., Nonpositive matrices in structural modeling. In *Testing Structural Equation Models* (eds Bollen, K. A. and Long, J. S.), 1993, pp. 256–293.

Received 12 March 2007; revised accepted 7 September 2007