Missing Data Estimation using Principle Component Analysis and Autoassociative Neural Networks

Jaisheel Mistry
Department of Electrical and Information Engineering, University of Witwatersrand
Johannesburg, 2050, South Africa

Fulufhelo V. Nelwamondo
Council for Scientific and Industrial Research, CSIR,
Johannesburg, 2050, South Africa

Tshilidzi Marwala
Engineering and Built Environment, University of Johannesburg,
Johannesburg, 2050, South Africa

ABSTRACT

Three new methods are used for estimating missing data in a database using Neural Networks, Principal Component Analysis and Genetic Algorithms are presented. The proposed methods are tested on a set of data obtained from the South African Antenatal Survey. The data is a collection of demographic properties of patients. The proposed methods use Principal Component Analysis to remove redundancies and reduce the dimensionality in the data. Variations of autoassociative Neural Networks are used to further reduce the dimensionality of the data. A Genetic Algorithm is then used to find the missing data by optimizing the error function of the three variants of the Autoencoder Neural Network. The proposed system was tested on data with 1 to 6 missing fields in a single record of data and the accuracy of the estimated values were calculated and recorded. All methods are as accurate as a conventional feedforward neural network structure however the use of the newly proposed methods employs neural network architectures that have fewer hidden nodes.

Keywords: Missing Data, Autoencoder Neural Networks, Auto Associative Neural Network, Principal Component Analysis and Genetic Algorithm

1. INTRODUCTION

The problem of having missing data for a database or for a real time system can have adverse effects. Missing data in databases is a common problem when surveys are taken and as a result proper analysis on the data cannot be done. Missing or faulty sensor inputs to a machine's control system is unacceptable because the machine may not operate appropriately and hence a method is required to efficiently estimate the missing values. Various statistical methods discussed in [1, 2, 3, 4] were used to estimate missing data and newer methods include the use of computationally intelligent methods to estimate missing data [5, 6, 7, 8, 9]. In this paper three variations of the proposed autoassociative neural network with genetic algorithm used by [5, 10] is presented. The focus of two of the proposed methods is to reduce the dimensionality of the input data by compressing data by using Principal Component Analysis.

The data used was collected for the 2001 antenatal survey and this data keeps a record of demographic properties of patients. The same data was previously used by [11] and [12] but they were used specifically to predict the HIV/AIDS status of a patient. In this investigation we attempt to estimate any of the missing fields in the database by using the previously mentioned computational intelligent methods. Work presented in [13, 14, 15, 16, 17, 18] for similar types of nonlinear systems justifies the use of Computational Intelligent methods such as Neural Networks and Genetic Algorithms.

The following section gives a brief background on missing data, Neural Networks, Genetic Algorithms and Principal Component Analysis. The newly proposed methods for estimating the missing data is presented and this is followed by the results obtained when testing the new methods.

2. THEORATICAL BACKGROUND

HIV Modeling

In 1982 Root-Berstein defined the Acquired Immunodeficiency Syndrome (AIDS) for unusual immune system failure. It was then found that the Human Immunodeficiency Virus (HIV) was identified as the cause of AIDS. Besides identifying the virus, much research has been done to better understand the virus. The HIV virus has already claimed more than 20 million lives by the end of 2007. The HIV/AIDS virus has spread rapidly in South Africa which currently has the highest prevalence rate in the world. Some of the research on this problem includes

investigating the causes of the HIV virus, predicting the HIV status for risk analysis purposes and to better understand the risks of such a virus. In the field of bioinformatics HIV classifications has been done using neural networks. Using patient data to better understand and model the HIV epidemic is not uncommon. Work by Knorr and Srivastava was done to model the intracellular and intercellular scale HIV dynamics of a person using patient data. Lurie et al. developed a decision analysis model for HIV testing using health workers and hospitals patient information. Other models of the HIV virus that are based on patient data.

Missing Data

Popular methods for dealing with missing data include substitution, hot deck imputation, regression methods and expectation maximization. These methods are briefly explained and discussed.

Substitution Methods: Two types of popular substitution methods include mean substitution or zero substitution. Zero substitution deals with placing a zero as the estimate value Using this method is not sensible because it may have no relevance to the data type that is to be estimated. Suppose a person's age is substituted with a zero because it is not known. For mean substitution the missing value is estimated to be the mean of the variable for all available cases. Mean substitution has the high likelihood of producing biased estimates, and hence it is also not recommended. Mean substitution may also result in values that are not sensible. Suppose there is a variable x which can either hold the value of 1, 2, 3 or 4. Now suppose we have three complete entries which take on the value of 1, 3 and 4. The mean value for x equals 2.33 and this value has no significance because x can only be in 1 of the four possible states .

Hot Deck Imputation: Hot deck imputation is a look up table method which works by finding a similar case as the one with the missing value. Suppose the variable x is missing in a given record, the x value is substituted with the x value of a record that has the same or similar values for the other fields. An advantage of this method is that the estimate values will be more sensible in that categorical variables will remain categorical and continuous variables will remain continuous. A disadvantage of using the hot deck method is that it is difficult to define similarity and the method is inefficient in cases where there is a large amount of uncertainty. This occurs when there are large amounts of similar cases and the missing variable type differs significantly from each other.

Regression Methods: For regression methods, a regression equation based on complete data for a given variable is derived. The missing variable is treated as being dependent on the other variables and hence can be estimated using the regression equation. A polynomial equation may not be sufficient to model non-linear systems; hence more advanced methods are required to model such non-linear systems. The methods presented in this paper for missing data estimation using a neural network can be thought of as a complex regression model used to model non-linear systems.

Expectation Maximization: The Expectation Maximization (EM) Algorithm is an iterative process that consists of 2 steps. The first step, known as the expectation (E) step, computes the value of the complete data log likelihood based upon the complete data cases and the algorithm's best guess as to which are the best statistical functions for the specified model. The second Maximization (M) step substitutes expected values for the missing data obtained in the E step so as to maximize the likelihood function. The E and M steps are repeated iteratively until convergence is obtained. The EM algorithm is thought to be a statistically sound method for estimating missing values . The disadvantage however is that the algorithm does not add any uncertainty component to the estimate data.

Missing Data Mechanisms: Before estimating missing data, it is important to understand why the data is missing. The reason for the data being missing is known as the missing data mechanism. The three main types of missing data mechanisms are Missing at Random (MAR), Missing Completely at Random (MCAR) and the non ignorable case. These mechanisms are explained further.

1. Missing at Random (MAR) . A MAR data is missing value that has a probability of being missing in field X and this missing data is dependent on other fields in the database but not on field X itself. A simple example is that the person's education level is missing in the database because of the age.
2. Missing Completely at Random (MCAR). MCAR occurs if the probability of a missing value that belongs to field X is not related to the field X itself and not to any other field in the dataset. A patient's husband's age may be missing due the fact that the patient does not know. The patient's husband's age is not dependent on any of the other variables in the database.
3. Non-Ignorable. The non-ignorable case is when a missing data in field X is dependent on field X itself. A simple example is that the patient has not gone to school and hence does not want to fill in the field for the maximum education achieved.

Neural Networks

A neural network is an information processing system that is inspired by the way the biological nervous system operates. Hence a neural network process information in similar manner to how the brain would process information. Neural Networks can be thought of as a machine that is designed to simulate a particular way the human brain performs a particular task [20].

There are a variety of neural network architectures and these include:

- Multi layer Perceptron (MLP)
- Radial Basis Function (RBF)
- Recurrent Neural Network (RNN)
- Hierachial Mixture of Experts (HME)

- Self organizing maps (SOM)
- Hybrid Neural Network (HNN)

The Multi-layer Perceptron (MLP) neural network consist of the input, hidden and output layer with each layer having a set of neurons or nodes. The neurons of each layer are interconnected to the proceeding layers neurons by weights.

Neural networks are used extensively for pattern recognition and to model non linear systems [21]. The Neural Network Matlab implementation Netlab [22] is used for neural network implementations in this paper.

Genetic Algorithms

Genetic Algorithms are algorithms that are inspired by the simulation of genetic processes such as inheritance, mutation, selection, and crossover (also called recombination) [23]. The Genetic Algorithm Optimization Toolbox (GAOT) [24] is a black box model that can be used to find the optimum value of a certain function. In the case of this project the GAOT toolbox is used to find the maximum value of a specific evaluation function. The processes used to get this input value and the corresponding output is done by following these steps:

1) Create an initial population of input values
2) Using the evaluation function, obtain the output to each of these values
3) Create a new population by choosing the fittest of the old population
4) Apply some genetic process/algorithm such as mutation and crossover to create a new population
5) Use the evaluation function on the new population
6) Repeat the previous three steps for a number of times that are specified by the user

Principal Component Analysis

In this paper Principal Component Analysis (PCA) [25] will be used to reduce the dimensionality of the input data as well as remove redundant fields in the database. The conventional PCA method that was implemented for this paper is a simple method where principle directions are found by finding the data points with the most variance. These directions are called the principal directions. The data is then projected onto the principal directions without the loss of significant information of the data. A brief outline of the implementation of the above mentioned method of PCA is described.
The first step involves computing a covariance matrix [26]

$$C_{ij} = \frac{\sum_{p=1}^{p}\left(x_i^p - \overline{x_i}\right)\left(x_j^p - \overline{x_j}\right)}{(n-1)} \qquad (1)$$

The covariance matrix C and the superscript P equals number of vectors in the training set; x is the input data; and $i = j = 1,..,N$ where N is the total number of different fields in the database. The next step involves calculating the eigenvalues and eigenvectors of the covariance matrix. Once these values are found the eigenvectors must be arranged in order of largest eigenvalue to the smallest eigenvalue. The first N eigenvalues are then chosen. The data are then projected onto the eigenvectors corresponding to the N most dominant eigenvalues. Hence to reduce the dimensionality of the data by compressing the data the following equation applies

$$CompressedData = Data \times CM \qquad (2)$$

where CM is the Compression Matrix which is represented by the eigenvectors of the N largest eigenvalues of the Covariance matrix. When the compressed data is to be decompressed the following equation is used

$$DecompressedData = CompressedData \times CM^T$$

where CM^T is the transpose of the compression matrix used in equation 2. The selection of the N eigenvectors to construct the compression matrix (CM) is dependent on the acceptable error between decompressed data and the actual data which is calculated as follows .

$$Error = ActualData - DecompressedData \qquad (4$$

3. METHOD

Three new methods that are a variation of the methods used by Abdella's autoassociative Neural Network method [10] are presented. Firstly the data preparation is discussed and this will be followed by a description of autoassociative Neural Networks. After this the 3 variations of the autoassociative neural network are presented.

Data Preparation

Data preparation is an essential part of model fitting and this must be done before the design and training of a neural network. It is necessary to have reliable data that can be representative of the dominant properties of the system. Table 1 shows the different fields of data that are going to be used as inputs to the missing data estimator system. In table 1, the forth column distinguishes the data as either categorical or ordinal variables. The difference between these two types of data is that categorical data represents data that have discrete and definite values. For example the HIV status data field can either be 0 or 1 and a value of 0.5 for HIV status is nonsensical. To obtain a good set of data it is essential that the data be normalized, outliers from the data be removed and the data be randomized.

Data is normalized so that when training the Neural Network each of the inputs and outputs are of equal significance and this prevents a neural network from becoming biased toward a select range of inputs.

In order to make race and province categorical values the use of unary coding is used [27]. Categorical variables such as province and race cannot be compared. Suppose the variable race Asian is represented by integer value 1 and the race African is represented by the integer value 2 (Ordinal System). It is incorrect to say Asian < African and in a similar manor it is incorrect to say that Asian + Asian = African. Hence the use of unary codes has been made to represent the inputs such as race and province. Hence there are 21 inputs for the missing data estimation system because 9 inputs are used to represent the provinces, 5 inputs are used to represent the races and there

are a further 7 inputs to represent the remaining fields as indicated in table 1.

RPR Test	Binary	0 or 1	Categorical
Race	Integer	1-5	Categorical

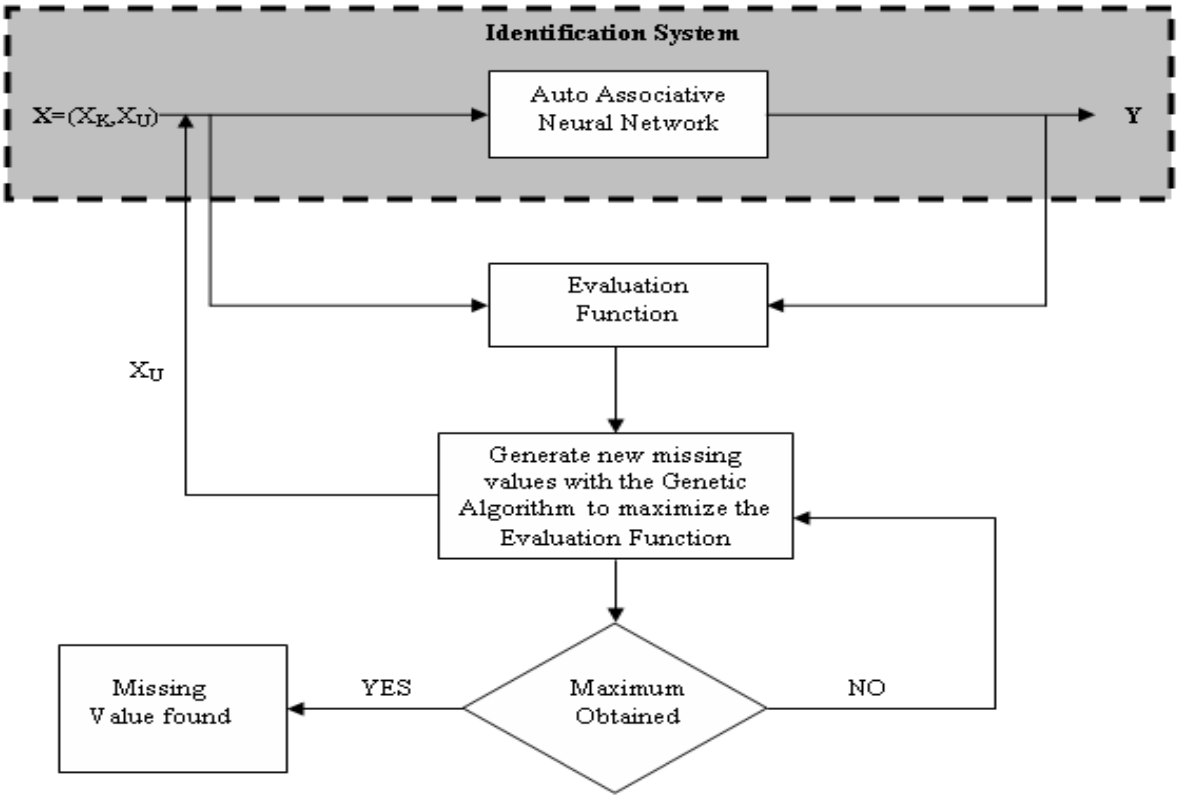

Figure 1. Flow Chart of the autoassociative neural network with genetic algorithm used by [10]

The RPR Test field is represented by a single digit binary number where a binary 1 represents a patient who has taken the RPR test and a binary 0 represents a patient that has not taken the RPR test. In a similar manner the HIV status is classified where 1 represents an HIV positive patient and a 0 represents a HIV negative patient.

The fields that are of an ordinal type are fields that do not have to have discrete values and they are values that can be compared. For example ages and education levels can be compared numerically.

Table 1: Summary of Different Fields of Data

Variable	Type	Range	Variable Type
Province	Integer	1-9	Categorical
Age	Integer	14-50	Ordinal
Education	Integer	0-13	Ordinal
Gravidity	Integer	0-6	Ordinal
Parity	Integer	0-6	Ordinal
Father Age	Integer	14-50	Ordinal
HIV Status	Binary	0 or 1	Categorical

As indicated in table 1 it is necessary to set a range for each of the data fields and each field data is normalized within these limits. It is therefore necessary to remove unusual data that does not occur often and data that is out of range on an intuitive level. The data that have to be removed are called outliers to the system.

The data provided by the South African antenatal survey was sorted according to provinces. It was necessary to randomize the data because the data was to be partitioned into 3 parts for the training, validation and testing of the neural network systems to be designed. The randomization of each record of data was required to prevent the neural network from learning the dynamics of the HIV system for only a few of the provinces.

Autoassociative Neural Networks

An auto associative neural network is a neural network that is trained so that the outputs of the network recall the inputs of the network [28]. The autoassociative neural network model with the use of genetic algorithms to estimate missing data was used by [10] and this method can be summarized by the flowchart found in figure 1.Mathematically, the Autoassociative Neural Network can be written as

$$\vec{Y} = f\{\vec{X}, \vec{W}\} \tag{5}$$

where the neural network is trained to predict an output that is the same as the input.
The error due to the inaccuracy of the autoassociative neural network is given by

$$e = \vec{X} - \vec{Y} \tag{6}$$

Substituting $\vec{Y}$ from 5 into 6 we get

$$e = \vec{X} - f\{\vec{X}, \vec{W}\} \tag{7}$$

Hence to obtain a positive error we square the equation 7 to give us

$$e = (\vec{X} - f\{\vec{X}, \vec{W}\})^2 \tag{8}$$

For the situation where missing data is to be estimated we have both known and unknown (missing) data. Known data is represented by $\vec{X_K}$ and Unknown data is represented by $\vec{X_U}$.

Rewriting equation 8 in terms of $\vec{X_K}$ and $\vec{X_U}$ gives us

$$e = \left(\begin{bmatrix}\vec{X_k}\\ \vec{X_U}\end{bmatrix} - f\left\{\begin{bmatrix}\vec{X_k}\\ \vec{X_U}\end{bmatrix}, \vec{W}\right\}\right)^2 \tag{9}$$

Genetic algorithms find the global maximum and hence it is required that the evaluation must have a maximum and this can be obtained by maximizing the error function as follows

$$EvalFunc = -\left(\begin{bmatrix}\vec{X_k}\\ \vec{X_U}\end{bmatrix} - f\left\{\begin{bmatrix}\vec{X_k}\\ \vec{X_U}\end{bmatrix}, \vec{W}\right\}\right)^2 \tag{10}$$

As suggested in [11], it is better to have an auto-encoder neural network rather than a conventional feedforward Neural Network to perform the task of the Autoassociative Neural Network. An auto-encoder neural network is a 3 layered neural network with an input, hidden and output layer where the input and the output are the same. A key requirement of an autoencoder neural network is for the network to have fewer hidden nodes than the number of input nodes so that the auto-encoder network learns a compressed version of the data.

Another logical reason for having an auto-encoder neural network is so that we do not end up with a situation which is best described with the help of figure 2. A situation can easily arise where the dotted weights in figure 2 can tend to the value 1 and the other weights marked by the solid lines tend toward the value 0. This situation can easily arise if the network is over trained.

Hence if we have an auto-encoder neural network as shown in figure 3, so that we can avoid the situation explained for figure 2.

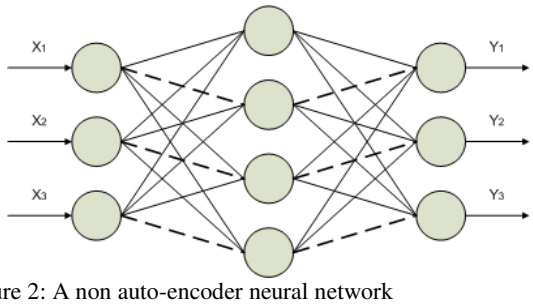

Figure 2: A non auto-encoder neural network

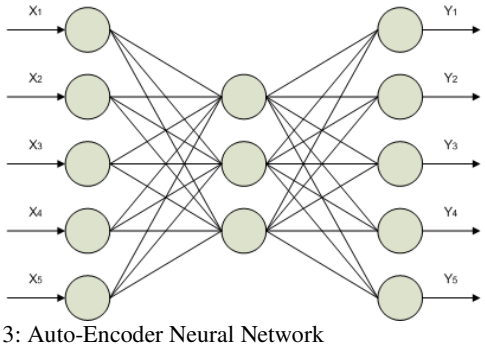

Figure 3: Auto-Encoder Neural Network

Method 1

For this method the paradigm shift of using an autoassociative neural network is broken in that the neural network is not designed to predict the same inputs as the outputs. For this method a neural network that has fewer hidden nodes than input nodes is trained to predict an output that is a linear function of the inputs.

Mathematically the output of the neural network is given by

$$\vec{Y} = g\{\vec{X}\} \tag{11}$$

where $g\{\square\}$ is a linear function such as $\sin\square$ or $\cos\square$. Because of the inclusion of the linear function the evaluation function used by the genetic algorithm is changed to

$$EvalFunc = -\left(g\left\{\begin{bmatrix}\vec{X_k}\\ \vec{X_U}\end{bmatrix}\right\} - f\left\{\begin{bmatrix}\vec{X_k}\\ \vec{X_U}\end{bmatrix}, \vec{W}\right\}\right)^2 \tag{12}$$

It is said in [29] that the best auto-encoder neural network is one with the least number of hidden nodes because the network will be of less complexity and capture the dynamics of the system with avoiding the situation as explained for figure 2. The structure of the network must be similar to that of an autoencoder neural network and hence it is required that the number of hidden nodes be minimal. Hence an investigation was done to find the minimum number of hidden nodes required to model the system. The use of principal component analysis was used to find the minimum number of hidden nodes by investigating the minimum number of eigenvectors required to compress and decompress the data but still ensuring that the error on ordinal variables are low and that the accuracy on categorical outputs are high. Figure 4 shows the results of the investigation and from here we can see that the optimum numbers of hidden nodes required are 16.

Method 2

The second method that is to be used is the same as the one explained in figure 1 but the identification scheme is changed so that the autoassociative neural network part is replaced by an auto-encoder neural network that works on a compressed version of the input data.

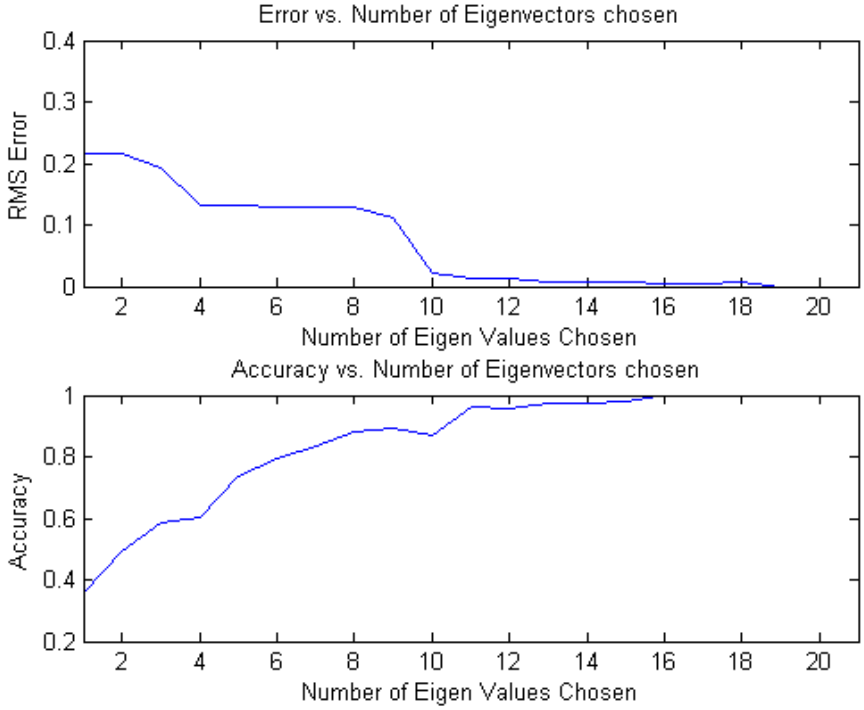

Figure 4: Accuracy and error of data for data compression

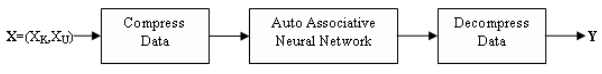

Figure 5: Identification Scheme for model 2

Hence the identification block shown in figure 1 is replaced by the one shown in figure 5. The methods used to compress and decompress the data are executed by making use of PCA as shown in equation 2 and 3. Hence the identification system equation is given as

$$\vec{Y} = f\left\{\vec{X} \times CM, \vec{W}\right\} \times CM^T \quad (13)$$

The evaluation function that is evaluated using the Genetic Algorithms are as follows

$$EvalFunc = -\left(\vec{X} - f\left\{\vec{X} \times CM, \vec{W}\right\} \times CM^T\right)^2 \quad (14)$$

where $\vec{X}$ is representative of known values ($\vec{X_K}$) and unknown values ($\vec{X_U}$).

Testing was done for the identification system explained by equation 13 and it was found that the best model would be one with the data being compressed from 21 to 16 fields and then the dimensionality of the data is further reduced because the autoencoder neural network that works on the compressed data only has 13 hidden nodes. The reduction in dimensionality in the data could probably be accounted for by the redundant number of inputs used in order to implement the unary coding scheme for the race and provincial data fields. Note that the compression matrix (CM) in equation 13 and equation 14 is computed only once on the training data set.

Method 3

This method differs from the model in figure 1 in that the identification system used is one that is described in figure 6. In

this model an ordinary feedforward neural network is designed to model the PCA compression of data. Therefore a Neural Network is trained to model the situation given by equation 15.

$$\vec{Y} = \vec{X} \times CM \tag{15}$$

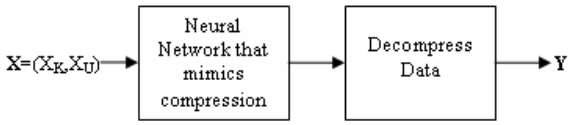

Figure 6: Identification Scheme for method 3

Suppose the mimic Neural Network function is represented by $g\{\square\}$. The output of the neural network is then decompressed using conventional PCA methods as shown in equation 3. Therefore the evaluation function used by the genetic algorithm will be as follows:

$$EvalFunc = -\left(\left[\frac{\vec{X_k}}{\vec{X_U}}\right] - g\left\{\left[\frac{\vec{X_k}}{\vec{X_U}}\right], \vec{W}\right\} \times CM^T\right)^2 \tag{16}$$

The various parameters that can be changed in this model is the number of eigenvectors to construct the compression matrix (CM) and the number of hidden nodes of the neural network that mimics the PCA compression. It was found that the best parameters were when the data was compressed from 21 to 16 fields (16 eigenvectors chosen) and the number of hidden nodes used for the neural network was 12 hidden nodes. Note that the compression matrix CM was only computed once using the training data set.

4. RESULTS AND DISSCUSSION

In order to test the missing data estimation methods that were developed, a test data set with missing inputs was made. The test data set was made by taking a portion of valid data that was not used for the training, validation or testing of the neural networks. Data was randomly removed from this set. This set of data was segmented into six different sets and from each set a fixed amount (ranging from 1 to 6) fields from each records data was removed.

The different missing data estimation methods were applied to find the estimate values. The predicted values from the systems were then compared with the actual data and the following two parameters were computed.

The RMS error for ordinal variables were calculated as follows

$$E_{RMS} = \frac{\sqrt{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}}{n} \tag{17}$$

where x_i is the actual value and $\hat{x}_i$ is the estimated value. The accuracy of the categorical outputs are calculated using the following equation

$$Accuracy = \frac{TP + TN}{TN + FP + TN + FN} \tag{18}$$

where TP is the true positive, TN is the true negative, FP is the false positives and FN are the false negatives. The accuracy of the predicted output for the fields such as race and province are calculated as follows

$$Accuracy = \frac{Correct\ Guesses}{Correct\ Guesses + Incorrect\ Guesses} \tag{19}$$

The graphs in figure 7 and 8 show the performance of the various systems. In addition to the three methods discussed in this paper an auto-encoder neural network with 16 hidden nodes and an autoassociative neural network with 25 nodes were used as the identification system as shown in figure 1. The results for using these identification schemes are also found in figure 7 and 8.

All the three methods have an accuracy of greater than 50% for predicting categorical variables and errors less than 30% for estimating ordinal values. These estimated values would be more accurate than using random selection or using average values.

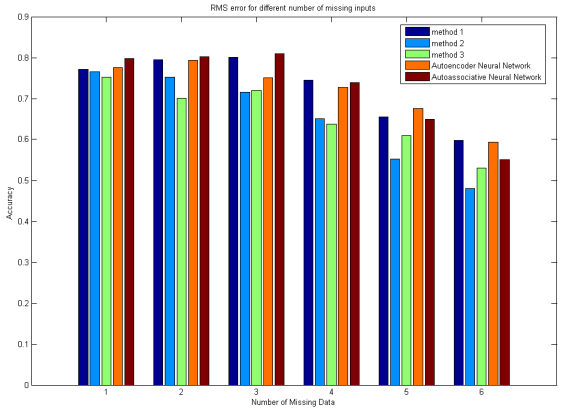

Figure 7: Accuracy of the different estimation methods

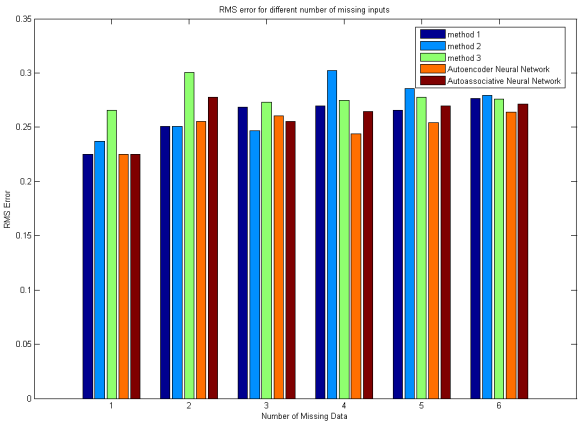

Figure 8: RMS error of different estimation methods

For the first method the use of a linear function as the output of the neural network instead of a conventional autoassociative neural network helps in keeping the weights values more uniformly distributed rather than having some weights tending to the value 1 and others becoming insignificant (tending to the value 0). The second estimation method can be used in fields where the user does not know if there is redundancy in data because the compression of data removes redundancies in data and helps make the neural network architecture simpler. The third model is useful in situations where the missing data must be estimated in real time because this method converges to the correct solution quicker and diverges from the incorrect solution as quickly because the error generated by estimating the wrong input value propagates through the entire system.

5. CONCLUSION

New alternate computational intelligent methods that make use of data compression are used to estimate missing data in a database. The data used to build, model and test the missing data estimator was obtained from the South African antenatal seroprevalence survey. The new methods perform equally as accurately as a conventional feedforward neural network but use simpler neural network architectures. The use of principal component analysis can be used to correctly identify the number of hidden nodes required for an auto-encoder neural network. All three models were used to estimate multiple fields of missing data with a reasonable accuracy.

6. REFERENCES

[1] R.Little and D. Rubin, Statistical Analysis with missing *data*, 2nd ed. New York: John Wiley and Sons, 1987.

[2] P.Allison, "Multiple imputation for missing data: A cautionary tale," In Sociological Methods and Research, vol. 28, pp. 301–309, 2000.

[3] M. Hu, S. Savucci, and M.Choen, "Evaluation of some popular imputation algorithms," *In Proceedings of the* Survey Research Methods Section of the American Statistical *Association*, pp. 308–313, 1998.

[4] D. Rubin, "Multiple imputations in sample surveys – a phenomenological bayesian approach to nonresponse," *In Proceedings of the Survey Research Methods Section of the American Statistical Association*, pp. 20–34, 1978.

[5] M. Abdella and T. Marwala, "The use of genetic algorithms and neural networks to approximate missing data in database," in *Computational Cybernetics , IEEE 3rd International Conference*, ICCC, Apr. 2005, pp. 207–212.

[6] B. Crossingham and T. Marwala, "Bayesian approach to neuro-rough models," *arXiv.0705.0761*, May 2007.

[7] F. Nelwamondo and T. Marwala, "Rough sets computations to impute missing data," *arXiv.0704.363*, Apr. 2007.

[8] F. Nelwamondo, S. Mohamed, and T. Marwala, "Missing data: A comparison of neural networks and expectation maximisation," *arXiv.0704.3474*, Apr. 2007.

[9] M. Scholz, "Non-linear pca: a missing data approach," *BioInformatics*, vol. 21, pp. 3887–3895, 2005.

[10] M. Abdella, "The use of genetic algorithms and neural networks to approximate missing data in database," M. Eng. thesis, University of Witwaterrand, Johannesburg , South Africa, Jan. 2005.

[11] B. Leke, T. Marwala, and T.Tettey, "Autoencoder Networks for hiv classification," *Current Science*, vol. 91, pp.1467–1473, 2006.

[12] T. H. Tim, "Predicting hiv status using neural networks and demographic factors," M. Eng. thesis, University of Witwaterrand, Johannesburg , South Africa, Apr. 2006.

[13] K. Poundstone, S. Strathdee, and D. Celectrano, "The social epidemology of human immunodeficiency virus/acquired immunodeficiency sysndrome," *Epidemiologic Rev*, vol. 26, pp. 22–35, 2004.

[14] P. Lisboa, "A review of evidence of health benefit from artificial neural networks in medical intervention," *Neural Networks*, vol. 15, pp. 11–39, 2002.

[15] M. Fernandez and J. Caballero, "Modeling of activity of cyclic urea hiv-1 protease inhibitors using regularizedartificial neural networks," *J.Bioorg.Med.Chem*, vol. 14, pp. 280–294, 2006.

[16] C. Lee and J. Park, "Assesment of hiv/aids-related health perfomance using an artificial neural network," *J.Inf.Manage*, vol. 38, pp. 231–238, 2001.

[17] S. Sardari and D. Sardari, "Applications of artificial neural networks in aids research and therapy," C*urr.Pharmaceut.Design*, vol. 8, pp. 659–670, 2002.

[18] E. Laumann and Y. Youm, "Racial/ethnic group differences in the prevalence of sexually transmitted diseases in the united states: a network explanation," *Sex Transm*, vol. 26, pp. 250–261, 1999.

[19] Y. Yoon and L. Peterson, "Artificial neural networks: an emerging new technique," *In Proceedings of the 1990 ACM SIGBDP conference trends and directions in expert systems*, pp. 417–422, 1990.

[20] S. Haykin, *Neural Networks*. New York: Prentice-Hall, 1995.

[21] C. Bishop, *Neural Networks for pattern recognition*. U.K: Oxford University Press, 1995.

[22] I. Nabney, *NETLAB: Algorithms for Pattern Recognition*. London: Springer-Verlag, 2003.

[23] D. Goldberg, *Genetic Algorithms in Searcgh Optimization and Machine Learning*. Reading: Addison-Wesley, 1989.

[24] C. Houck, J. Joines, and M. Kay, "A Genetic Algorithm for Function Optimization: A Matlab Implementation," North Carolina State University, Raleigh, NC, Tech. Rep. NCSU-IE-TR-95-09, 1995.

[25] I. Jollife, *Principal Component Analysis*. New York: Springer, 1986.

[26] L. Smith. (2002, Feb.) A tutorial on pricipal component analysis. University of Otago. New Zealand. [Online]. Available: http://csnet.otago.ac.nz/cosc453

[27] T.Marwala. (2006, Jan.) Lectures on introduction to neural networks. Dept. Elect. Eng., University of Witwatersrand. Johannesburg.[Online]. Available: http://dept.ee.wits.ac.za/marwala/ai.htm

[28] P. Lu and T. Hsu, "Applications of autoassociative neural networks on gass-path sensor data validation," *J.Propul,Power*, vol. 18, pp. 879–888, 2002.

[29] M. Kramer, "Non linear pricipal component analysis using autoassociative neural networks," *AlCheJ.*, vol. 37, pp. 233–234, 1991.