Information Sciences 237 (2013) 49–58

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A dynamic programming approach to missing data estimation using neural networks

Fulufhelo V. Nelwamondo [a,b,*], Dan Golding [a], Tshilidzi Marwala [b]

[a] Modelling and Digital Science Unit, Council for Scientific and Industrial Research, P.O. Box 91230, Auckland Park, 2006, Johannesburg, South Africa
[b] Faculty of Engineering and the Built Environment, University of Johannesburg, P.O. Box 524, Auckland Park, 2006, Johannesburg, South Africa

ARTICLE INFO

Article history:
Received 31 July 2008
Received in revised form 30 September 2009
Accepted 6 October 2009
Available online 16 October 2009

Keywords:
Missing data
Neural networks
Data imputation
Dynamic programming
Genetic algorithms

ABSTRACT

This paper develops and presents a novel technique for missing data estimation using a combination of dynamic programming, neural networks and genetic algorithms (GA) on suitable subsets of the input data. The method proposed here is well suited for decision making processes and uses the concept of optimality and the Bellman's equation to estimate the missing data. The proposed approach is applied to an HIV/AIDS database and the results shows that the proposed method significantly outperforms a similar method where dynamic programming is not used. This paper also suggests a different way of formulating a missing data problem such that the dynamic programming is applicable to estimate the missing data.

1. Introduction

Decision making processes are highly dependent on the availability of data, from which information can be extracted. All scientific, business and economic decisions are somehow related to the information available at the time of making such decisions. It is for this reason that the problem of missing data afflicts a variety of research and application areas in fields such as engineering, economics, finance and many more. Most predictive and decision making models designed to use a specified number of inputs will breakdown when one or more inputs are not available. In many such applications, simply ignoring or deleting the incomplete record (known as case deletion) is not a favorable option, as it may bring more harm than good [2]. In a statistical model, case deletion can also lead to biased results and in applications such as machine control, case deletion may result in breakdown of machinery [23]. Many techniques to estimate missing data that are aimed at minimizing the bias or output error of a model have been extensively researched [17,24,25]. Most of these are statistical methods, one of the most successful being Bayesian multiple imputation [25]. It is unfortunate that most decision making tools such as the commonly used neural networks and many other computational intelligence techniques cannot be used for decision making if data are not complete. In such cases, the optimal decision output should nevertheless, still be maintained despite the missing data. The estimation of missing input vector elements 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, as a result, are useful in the imputation of missing data [20].

* Corresponding author. Address: Modelling and Digital Science Unit, Council for Scientific and Industrial Research, P.O. Box 91230, Auckland Park, 2006, Johannesburg, South Africa. Tel.: +27 11 358 0051; fax: +27 11 358 0230.
E-mail address: fnelwamondo@csir.co.za (F.V. Nelwamondo).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2009.10.008

This paper proposes a novel technique for missing data estimation, grounded on the theory of dynamic programing. The novel method proposed here uses neural networks and genetic algorithms (GA) on suitable subsets of the input data, and assumes some model that emits data, some of which are missing. The remainder of this paper is arranged as follows: Section 2 presents a literature review and discusses related methods. The problem is presented in detail in Section 3, followed by the background information in Section 4. Sections 5–7 present in detail, information on the proposed model together with a description of the base model. Lastly, experimental results are given followed by a discussion.

2. Literature review

The problem of missing data in decision making has been a subject of research for over four decades, with more emphasis on statistical methods [24]. There are many applications that some researchers have worked on, with a major interest in making decisions with incomplete information. Dokuchaev [9] presented some qualitative methods and empirical rules for incomplete information in stochastic models in financial investment. Research in the field of missing data also took off in machine learning. Cogill et al. [7] studied a decentralized control problem using dynamic programming. One notable similarity between their study and the one presented in this paper is that they are both motivated by the fact that in many applications, decisions must be made at each time period and often with incomplete information. Cogill et al. [7] and Qiao et al. [22] do not attempt to estimate missing data, but present a model that can work reasonably well in the presence of missing data. Another relevant work was conducted by Geffner and Bonet [11] when solving a high-level planning with incomplete information. In their approach, they use partially observable Markov decision processes, which they transform into controllers using dynamic programming. The goal of their work was to design a shell that accepts partially observable Markov decision processes (POMDP) and produce controllers. Clearly, they were not estimating missing data as it is done in this work.

Some of the researchers who attempted to estimate missing data using some of the tools in machine learning include Twala [28] and He [13] who investigated the problem using decision trees. A thorough review of the literature reveals that there has not been any work on this area using dynamic programming. Some work on this area is in the book by Marwala [18], where he only points out dynamic programming as an emerging technique. Nelwamondo [19] has applied dynamic programming in the missing data problem. The approach used by Nelwamondo breaks the problem into smaller sub-problems, and the sub-problems are broken down into even smaller sub-problems and these smaller sub-problems were solved strategically to solve the bigger problem of missing data. The next section defines the problem and will justify why the problem is solved using dynamic programming.

3. Problem formulation and assumptions

Suppose there exists some dataset D with n variables and some large number of observations of these variables. Further suppose that, at each observation, all n variables are recorded as if observed. When recorded, all the observations are used as input to some decision making system. There certainly will be a problem if at some time t, some variables are missing from the observation. This paper defines a set $S \in \{0, 1, \ldots, n\}$ of possible states of the world. In this case, states can be any variable that is missing. State 0 is the one where all variables are observed and the remaining states correspond to the features. By design, the model is assumed to only be at one state at a time and can jump to any state in the next observation. In other application, the states can also be defined as a combination of the missing variables, but this will not be the focus of this paper. The transition matrix for this process of changing states is considered in this paper to be unknown, but can be derived from the data. We also define a set of corresponding actions that depend on what state the missingness model is. There are three possible actions given as [*do-nothing, recall, estimate*]. *Do-nothing* is the action that will be optimal if no data are missing or the state is 0. *Recall* is taken when a similar case has been observed in the past such that the best policy now will be to recall the action taken then, and *estimate* occurs when the missing data have to be estimated as discussed later in this paper. There is also a reward model R, which offers the highest reward when the estimate is accurate. In this case, the reward is only associated with the states and not the combination of the state and action. What is meant here is that the reward model does not look at what action was taken. It instead, only looks at how close the estimates are to the real answers. It becomes clear that what is being solved is the optimality equation, where we want to derive the best policy, and this is done well using dynamic programming [3]. The next section presents some background theory of the tools used.

4. Background information

4.1. Auto-encoder neural networks

Auto-encoders, also known as auto-associative neural networks, are neural networks trained to recall the input space. Thompson et al. [27] distinguish two primary features of an auto-encoder network, namely, the auto-associative nature of the network 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, auto-encoders are preferred due to their remarkable ability to learn certain linear and non-linear interrelationships such as correlation and covariance inherent in the input space. In this