Modelling Multivariate Time Series

X Liu, S Swift, A Tucker, G Cheng and G Loizou
Department of Computer Science,
Birkbeck College, University of London,
Malet Street, London WC1E 7HX, United Kingdom

Abstract

Multivariate time series (MTS) data are widely available in different fields including medicine, finance, science and engineering. Modelling MTS data effectively is important for many decision-making activities. In this paper, we will describe some of our efforts in modelling these data for numerous tasks such as outlier analysis, forecasting and explanation. Through the analysis of various kinds of MTS data to achieve those tasks, we hope to trigger discussions in the workshop about what "I" could mean in "IDA" for this type of application.

1 Introduction

Recent developments in computing have provided the basic infrastructure for fast access to vast amounts of online data; processing power and storage devices continue to become cheaper and more powerful, networks are providing more bandwidth and higher reliability, personal computers and workstations are widespread, and On-Line Analytic Processing (OLAP) allows rapid retrieval of data from data warehouses. This is especially true for the recording of time series data, for example in medical and financial sectors.

Clinicians evaluate patients' condition over time. The analysis of large quantities of time-stamped data will provide doctors with important information regarding the progress of a disease. Therefore systems capable of performing temporal abstraction and reasoning become crucial in this context. Shahar has developed a general framework for this purpose [28]. Although the use of temporal abstraction and reasoning methods requires an intensive knowledge acquisition effort, such methods have found many successful medical applications [19], including data validation in intensive care [11], the monitoring of child growth [8], the modelling of medical concepts [14], and the monitoring of heart transplant patients [18]. In particular, Bellazzi et al. [3] discuss the use of temporal abstraction to transform longitudinal data into a new time series containing more meaningful information, whose features are then interpreted using statistical and probabilistic methods. Their approach was successfully applied to the analysis of diabetic patients' data.

Much research has gone into the development of ways of analysing multivariate time series (MTS) data in both the statistical and artificial intelligence communities. Statistical MTS modelling methods include the Vector Auto-Regressive process, the Vector Auto-Regressive

Moving Average process, and other non-linear and Bayesian approaches [4, 23, 26], while various AI methods have been developed for different purposes. These include dependence detection in MTS of categorical data [24], knowledge-based temporal abstraction [13, 28], Bayesian clustering of similar MTSs [27], and forecasting [1, 31]. Over the last few years, we have looked at several MTS modelling tasks in public health and process industry and have started accumulating experiences in analysing MTS data.

In this paper, we shall look at three MTS modelling tasks in some detail: outlier analysis, short MTS forecasting, and explanation in dynamic processes. We hope that by examining how these tasks are approached we could trigger discussions on what "I" could mean in "IDA" for this type of application.

2 Outlier analysis

The handling of anomalous or outlying observations in a data set is important for the following reasons. First, outlying observations can have a considerable influence on the results of an analysis. Second, although outliers are often measurement or recording errors, some of them can represent phenomena of interest, something significant from the viewpoint of the application domain. Third, for many applications, exceptions identified can often lead to the discovery of unexpected knowledge.

There are two principal approaches to outlier management [2]. One is outlier *accommodation*, which is characterised by the development of a variety of statistical estimation or testing procedures which are *robust* against, or relatively unaffected by, outliers [12]. In these procedures, the analysis of the main body of data is the key objective and outliers themselves are not of prime concern. This approach is, however, unsuitable for those applications where explicit identification of anomalous observations is an important consideration, e.g. suspicious credit card transactions. The other approach is characterised by identifying outliers and deciding whether they should be retained or rejected. Many statistical techniques have been proposed to detect outliers and these techniques range from informal methods such as the ordering of multivariate data, the use of graphical and pictorial methods, and the application of simple test statistics, to some more formal approach in which a model for the data is provided, and tests of hypotheses that certain observations are outliers are set up against the alternative that they are part of the main body of data. The identification of outliers has also received much attention from the computing community [15].

However, there appears to be much less work on how to decide whether outliers should be retained or rejected. In the statistical community, a commonly-adopted strategy when analysing data is to carry out the analysis both including and excluding the suspicious values. If there is little difference in the results obtained then the outliers had minimal effect, but if excluding them does have an effect, options need to be sought to analyse these outliers further.

In order to successfully distinguish between noisy outlying data and noise-free outliers, different kinds of information are normally needed. These should not only include various data characteristics and the context in which the outliers occur, but also relevant domain knowledge. The procedure for analysing outliers has been experimentally shown to be subjective, depending on the above mentioned factors [6]. The analyst is normally given this task of judging which suspicious values are obviously impossible and which, while physically possible, should be viewed with caution. However, in the context of data mining where a large number of cases are normally involved, the number of suspicious cases would also be sizable and manual analysis would become too labour-intensive.

We have conducted much research on how to analyse outliers using domain knowledge [21, 32]. Two general strategies for distinguishing between phenomena of interest and measurement noise have been proposed. The first strategy attempts to model "real" measurements, namely how measurements should be distributed in a domain of interest, and rejects values that do not fall within the real measurements. The other strategy uses knowledge regarding our understanding of noisy data points instead, so as to help reason about outliers. Noisy data points are modelled, and those outliers are accepted if they are not accounted for by a noise model.

Figure 1 illustrates how the strategy based on noise models works. Suppose that a set of training data, by using relevant domain knowledge, can be labelled into two classes: "noisy" and "rest". Class "noisy" indicates that the corresponding outliers in this data set are noisy data points, while class "rest" says we have good reason to believe the outliers in the data set are not due to measurement noise. Given sufficient amounts of training data, one can use any supervised machine learning techniques to build a "noise model" and this model, after validation, can then be used to help distinguish between the two types of outliers.

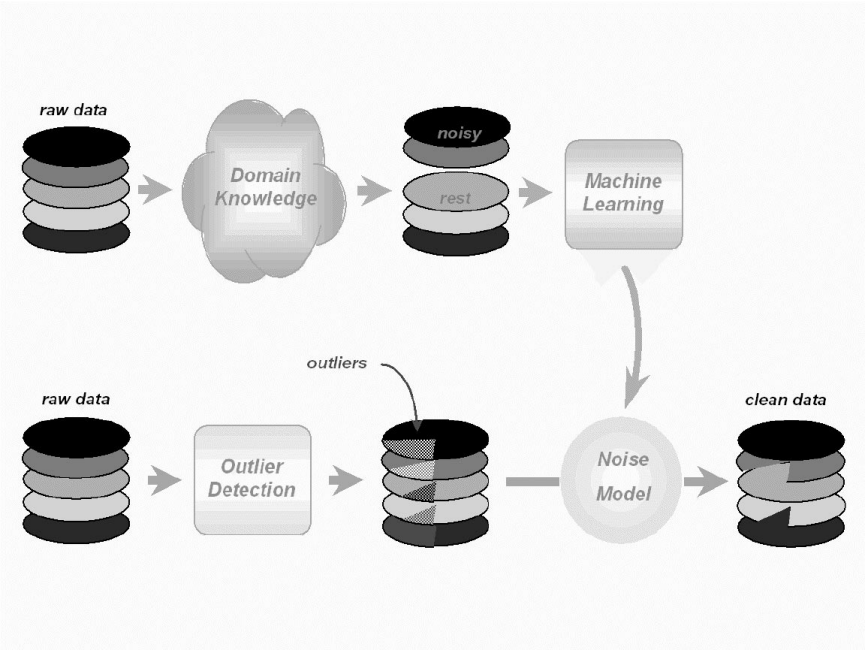

Figure 1: The Process of Outlier Analysis

Note that the labelling of training instances is not always easy, especially with multi-dimensional time-series data. To assist in this process, we used Self-Organising Maps [16] to visualise and compress data into a two-dimensional map. Data clusters and outliers then become easy to spot, and data are then relatively easily interpreted using the meaning of the map and relevant domain knowledge. So given a data set, outliers may be detected and can then be tested using the noise model. As a result, noisy outliers can then be deleted, while the rest of outliers are kept in the data set for further analysis.

This strategy has been successfully applied to glaucoma screening data, collected from subjects at various public environments [22]. Measurements were repeatedly taken over time from a number of fixed testing locations in the visual field. Let us look at a particular glaucoma study in some detail. All patients aged 40 years or older who routinely attended a GP clinic in North London for a three-month period were offered the test. A total of 925 patients were screened and 78 of them were later assessed clinically in the practice by an ophthalmologist; this sample included all people failing the test and a randomly sampled age-matched control group. Among these, 22 eyes were later assessed as glaucoma, 81 were confirmed as normal eyes without any disease, and the rest were diagnosed as other types of ocular abnormalities. The noise model was applied to these 103 test records, resulting in a dataset with selected outliers deleted. This dataset was then used to compare the performance of the test with those using the original (raw) dataset and another dataset obtained with all the outliers deleted. Figure 2 summarises the results of examining the *discriminating power* of the test in terms of its glaucoma detection rate versus false alarms using these three different datasets. The decision threshold used for discriminating between normal and abnormal eyes is the average percentage of positive responses within the test.

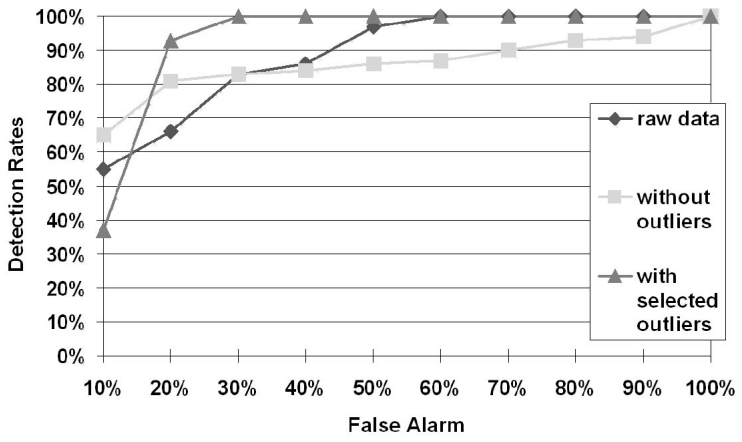

Figure 2: The Results

It is clear that the dataset with selected outliers performs better than the other two (the raw dataset and the dataset without any outliers). Although there are areas where using the dataset without outliers leads to better detection rates, the dataset with selected outliers performs better in most areas, especially in the top left corner. This area is the most

important since we are trying to find out which dataset allows for the maximisation of the detection rate, and at the same time, the minimisation of false alarms. This comparison demonstrates the usefulness of deleting noisy outliers from the test data.

3 Forecasting of short MTS

Although much research has been performed in the modelling of MTS data for forecasting purposes, one area that has been largely overlooked is the particular type of time series where the data set consists of a large number of variables but with a small number of observations.

We have been researching into the key issues in the modelling and analysis of the *short* MTS for prediction purposes. In particular, we have been looking into statistical MTS modelling methods [10] since these have the desirable feature of *interpretability* in that it is relatively easier to understand the internal constructs of the model. This feature is lacking in many modern methods such as neural networks. However, there are difficulties in using traditional statistical methods to model short MTS data. Let's have a look at the visual field data collected from a standard testing machine, crucial for the forecasting of visual deterioration in glaucoma patients. The patients were tested approximately every six months for between five and 22 years. The particular test used with this dataset examines 76 points in each eye, thereby producing a multivariate irregular time series of length between 10 and 44.

The visual-field dataset exhibited features that made the standard methods difficult to apply. The value of each of the 76 field points ranges exclusively between 0 and 60, and the length of this MTS is rather short. A suitable way of modelling this data appears to be the Vector Auto-Regressive Process, usually denoted as VAR(P) for a model of order P. The standard statistical methods for fitting a VAR process to a set of data often consist of two steps: order selection and parameter estimation.

Order selection is commonly performed through the use of information theory based metrics such as AIC (Akaike's Information Criterion) [23]. Many of these metrics will impose a restriction on N, the minimum length of a MTS, based on the number of degrees of freedom of the model being estimated, namely $N > KP + 1$, where K is the number of variables being modelled, and P is the order of the VAR process. For example, for the visual-field MTS involving 76 variables, to find the most appropriate order of a VAR process with a maximum order of five under consideration, N must be at least 382 in length. This restriction is unacceptable for modelling the visual-field type of data.

Even if the above problems were to be ignored, the parameter estimation step can experience some difficulties when dealing with a short MTS. The standard methods for parameter estimation include maximum likelihood (ML) methods, the Yule-Walker (YW) equations method, and the least squares (LS) method. With ML methods, there must be some distribution assumptions on the noise vector. Since the range of the visual field values is exclusively within a set interval, an appropriate continuous distribution is difficult to find. The YW method can involve matrix inversion, which is computationally expensive with large matrices and can fail if the matrix is singular. The LS method is often used

in preference to the YW equations, but it too can involve matrix inversion, and can also impose the degree of freedom restriction mentioned above (this is involved in computing an unbiased estimator for the covariance matrix of the associated noise vector).

However, a genetic algorithm can be used to find the parameters and order of a VAR process without making any of the assumptions outlined above. In addition, a genetic algorithm (GA) may reduce the length restriction to $N > P$, which makes it possible to model many short MTSs [29]. The general idea is illustrated in Figure 3. Below are some of the details.

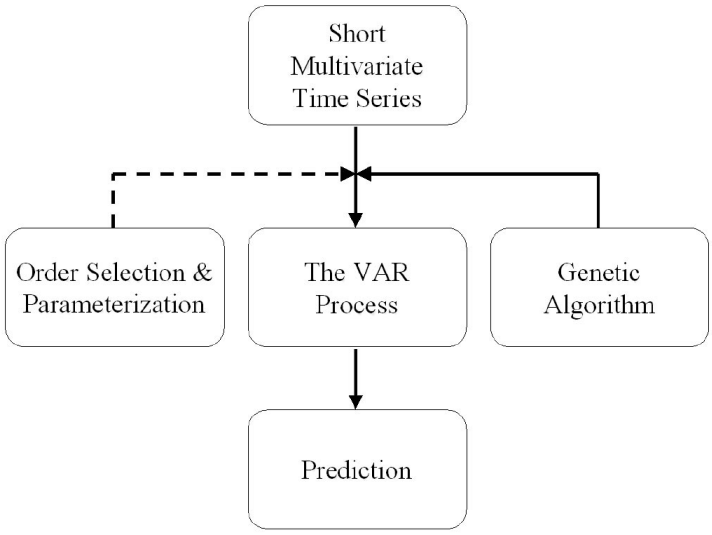

Figure 3: Modelling Short MTS for Forecasting

A VAR process of order P, written VAR(P), is defined in equation 1.

$$\underline{x}(t) = \sum_{i=1}^{p} A_i \underline{x}(t-i) + \underline{\varepsilon}(t) \tag{1}$$

Where $\underline{x}(t)$ is the next data vector of size K (the number of variables in the model), A_i is a $K \times K$ coefficient matrix at time lag i, and $\underline{\varepsilon}(t)$ is a K length noise vector at time t (usually Gaussian) with zero mean. The value of each element in A_i is usually a real number in the range [-1, 1]. To use equation 1 for prediction purposes the parameter matrices A_i must be estimated from the data.

VARGA is a genetic algorithm designed to find the order and associated parameter matrices for a VAR(P) process best suited to fitting a dataset. The level of accuracy for the GA (the fitness function) is defined in equations 2 and 3, namely

$$\hat{\varepsilon}(t) = \underline{x}(t) - \sum_{i=1}^{p} \hat{A}_i \underline{x}(t-i) \tag{2}$$

and

$$\varepsilon = \sum_{j=1}^{k} |\hat{\varepsilon}_j(t)| \,. \tag{3}$$

Where $\hat{\varepsilon}(t)$ is the estimation of the noise vector, $\hat{\varepsilon}_j(t)$ is the jth element of $\hat{\varepsilon}(t)$, $\hat{A}_i$ is the estimation of the ith parameter matrix, and ε is a scalar that represents the level of noise. All other variables were defined earlier. The model with the smallest ε value is deemed to be the best for forecasting since it is assumed that the best estimation for any unobserved noise vector is the zero vector.

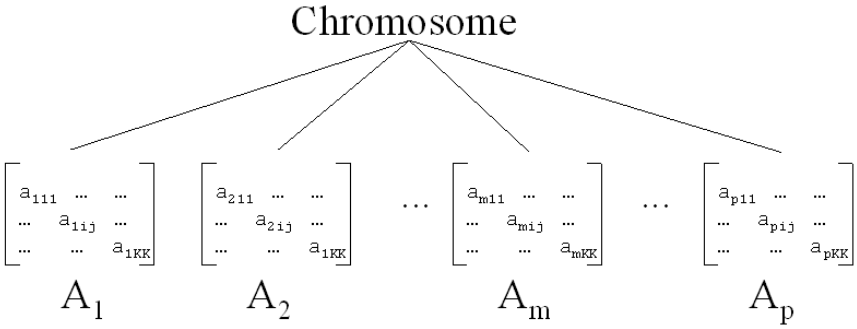

Figure 4: The Chromosome Representation

The chromosome representation is a list of $K \times K$ matrices, whose elements are integers over the set [0, 20000). A simple scaling is done to map each value into the set [-1, 1]. The order of each matrix in the list corresponds to the equivalent coefficient matrix for the VAR process being represented. The visual field data is mean-adjusted before being used in this method. This chromosome representation is shown in Figure 4. The VARGA algorithm essentially follows the standard Holland genetic algorithm, however, crossover is different and there are two mutation operators: one for *gene mutation* and the other for *order mutation* [29].

The models found using the VARGA method are compared with those produced by the conventional way of finding a VAR process, i.e. the solution of the Yule-Walker equations using S-Plus. To provide more insight into the accuracy of the VARGA method, it is further compared with the results from two other techniques: the Holt-Winters model (a commonly-used univariate forecasting model) and the noise model. Using a set of visual field data supplied by the Moorfields Eye Hospital, the VARGA method has been found to perform better than the other three methods for relatively long MTS, but also is able to predict the short MTS where other methods such as S-Plus may fail to do so [29]. The fact that it performs better than the Holt-Winters model appears to suggest that the VARGA method is able to take into consideration the dependencies among related time series variables. However, one should note that these results are obtained from only one real-world application, and many more experiments need to be performed on applications of different kinds to have a general picture regarding the applicability of VARGA. Work is indeed under way to obtain a deep understanding of various characteristics of the method.

3.1 Explanation in Dynamic Processes

The learning of appropriate models for diagnosis, or causal explanation, in MTS data is an important issue for many AI problems, including patient monitoring, robot navigation, data-mining for temporal sequencies and learning to control a complex plant. For example, many complex chemical processes record multivariate time-series data every minute. This data is characterised by a large number of interdependent variables, though some may have no substantial impact on any others. There can be large time lags between causes and effects (over 120 minutes in some chemical processes). In many situations certain anomalous events have a significant adverse economic impact, whether in terms of reduced yield, excessive equipment stress, or violation of environmental constraints. The identification of these events is important but of greater importance still are adequate diagnoses of them, which could then be used to modify operating practices, retrain operators or conduct anticipatory planning.

In order to perform diagnosis, some method is required for reasoning about relationships between these variables back in time. For example, the reason for a particular temperature becoming extremely high may be that a flow rate dropped ten minutes ago and the flow dropped because, one minute before that, a valve was closed by a control engineer. A number of diagnostic approaches have been proposed by different AI communities over the years. Early proposals include the use of *rule-based* and *model-based* systems. A more recent paradigm for performing causal inference is the Bayesian Network [25], and its dynamic counterpart can model a system over time [7]. Most of the research on dynamic networks, however, has focused on small models or models with small time lags. It would be desirable to learn a Dynamic Bayesian Network (DBN) for large datasets with large possible time lags such as oil refinery data.

The search for a causal model from a large multivariate time series with large time lags is a daunting task, particularly if it must be found quickly. In some applications such as diagnosis in an oil refinery, the causal explanation may be required in a very short space of time. We have suggested a general methodology which attempts to overcome some of the key problems associated with learning such a model. In particular, a representation that produces a DBN is proposed, based on a reasonable assumption. An algorithm for finding a good structure in as short a time as possible using evolutionary computation and relevant knowledge is then developed.

The assumption is that a DBN contains no links within the same time slice (contemporaneous links). A DBN with only non-contemporaneous links can be represented by a selection of $K+Q$ nodes, where K is the number of variables at a single time slice, and Q is the collection of nodes at previous time slices up to some maximum lag $MaxT$, with $Q \leq K \cdot MaxT$, where all members of Q have a direct dependency on nodes at time slice 0. We can use a list of triples to represent a possible network: (a,b,l) where a is the parent variable, b is the child variable and l is the time lag. Therefore each triple maps directly to a link in the network. So a list for $K = 5$, $MaxT = 5$ and $Q = 9$ such as (2,4,5), (4,3,4), (0,1,3), (2,1,2), (0,0,1), (1,1,1), (2,2,1), (3,3,1) and (4,4,1) would represent the DBN in Figure 5.

In many applications, where data is recorded frequently this assumption may well be true - that is, all variables take at least one time slice to have any effect on another variable.

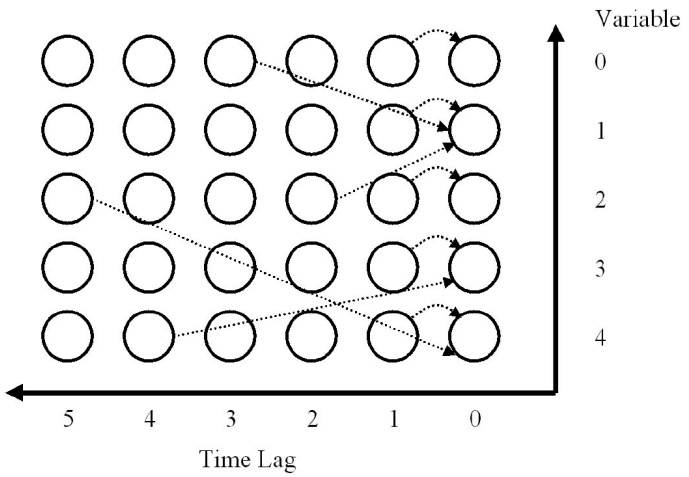

Figure 5: An Example DBN Using the Triple Representation

Although this assumption has already significantly reduced the search space, the number of possible network structures is still very large. For K variables and $MaxT$ possible time lags, this number will be:

$$2^{MaxT \cdot K^2}. \tag{4}$$

Two more heuristics are therefore used to contrain the search space even further: the exploitation of the Description Length [17] of each single link in the network, and the decomposition of the network into a group of simple tree structures [30]. These led to a general evolutionary algorithm for learning DBNs with the Description Length metric, outlined below.

Given a multivariate time series with K variables we can generate a DBN with a maximum time lag of $MaxT$. This is achieved firstly by applying Evolutionary Programming methods to a random selection of single triples in order to produce a list of *good* links. A random selection of triples from this list is then used to seed the initial generation of a GA where each individual chromosome comprises a selection of triples and the fitness function is the Description Length of the network. After a number of generations, the fittest individual comprises those triples which correspond to the DBN learned from the multivariate time series. The detailed algorithm is documented in [30].

To assess the performance of the algorithm we initially tested it on 10 synthetic datasets generated from different DBNs. Each DBN consisted of 11 variables and a maximum time lag of 60 minutes. The generated data contained 10000 data points for each variable and each variable could be in one of two possible states. We wanted to see if each original DBN could be learned quickly from its generated dataset using this algorithm. The next assessment step involved measuring how accurately the learned network structure represents the sort of relationships an expert would find in an oil refinery dataset. We do this by asking

a control engineer, who has extensive knowledge of the refinery process and data, to produce some dependency diagrams that represent the expected relationships between the variables in an oil refinery time series. We then compare these diagrams to explanations generated using the network learned from the same time series.

The limited space prevents us presenting all the experimental results from those two assessments, so we just summarise the key results here. The algorithm does indeed manage to find all the original DBN structures that were used to generate the synthetic data, and does so much more quickly than a standard GA algorithm. In the experiment involving the oil refinery data, the algorithm manages to detect all of the suggested relationships correctly; however, it generates more explanations than those specified in the dependency diagrams. This is to be expected since the diagrams are not meant to be exhaustive in that they only capture some of the obvious relations that should exist in the dataset. It must also be pointed out, however, that there are a few relationships found within the network structure that were known to be false, leading to some incorrect explanations. A full analysis of this phenomenon is given in [30]. Further research is under way to explore, on the evolutionary front, different parameters and operators; and on the Bayesian network front, different discretisation policies, scoring metrics and search strategies.

4 Concluding remarks

Data analysis is performed for a variety of reasons by scientists, engineers, medical and government researchers, business communities and so on. Statistical methods have been the primary analysis tool, but many new computing developments have been applied to challenging real-world problems. This is a truly exciting time. Questions have been constantly raised: how can one perform data analysis most effectively –intelligently– to improve the quality of life, to gain new scientific insights, to capture bigger portions of the market, and so on? And is there a set of established guiding principles to enable one to do so? Statistics has laid some important foundations, but the evolution of computing technology and the ever-increasing size and variety of data sets have led to new challenges as well as new opportunities for both the computing and statistical communities [9, 20].

Although basic IDA issues are beginning to be understood, this interdisciplinary field appears to be too young to allow for a precise definition of what is the "intelligent" way of performing data analysis. What is certain, though, is that IDA requires careful thinking at every stage of an analysis process, intelligent application of relevant domain expertise regarding both data and subject matters, and critical assessment, selection or integration of relevant analysis methods. In any case, we need to obtain further, substantial experience in analysing complex real-world data before we can have a definitive understanding of the intelligent data analysis process.

In this paper, we have described three pieces of work on the modelling of multivariate time series data. First, outlier analysis is one of the key *data quality* issues, and we have experimented with various ways of incorporating relevant domain knowledge to help distinguish between noisy outliers and noise-free outliers. Second, model selection is arguably the most important and most difficult aspect of *model building*, and yet is the one where there is least

help [5]. This situation is even worse for modelling *short* MTS data. We have developed a method that bypasses the size restrictions of traditional statistical methods, makes no distribution assumptions, and also locates the order and associated parameters as a whole step. Third, we have accumulated experiences about how evolutionary computation techniques, when incorporating relevant heuristics, may significantly reduce the search space, thereby helping make the algorithm for learning dynamic Bayesian networks *scalable*. A common feature in addressing all of the three issues is that AI techniques have been used to enhance the capabilities of traditional statistical methods for problem-solving.

5 Acknowledgement

We thank our research partners at Moorfields Eye Hospital, Institute of Ophthalmology, BP-Ameco, and Honeywell for their contributions to the work reported in the paper.

References

[1] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Bidirectional dynamics for protein secondary structure prediction. In *Proc. of the IJCAI-99 Workshop on Neural, Symbolic, and Reinforcement Methods for Sequence Learning*, pages 77–83. 1999.

[2] V. Barnet and T. Lewis. *Outliers in Statistical Data*. Wiley, 1994.

[3] R. Bellazzi, C. Larizza, and A. Riva. Interpreting longitudinal data through temporal abstractions: An application to diabetic patients monitoring. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis: Reasoning about Data, LNCS 1280*, pages 287–298. Springer-Verlag, 1997.

[4] M. Casdagli and S. Eubank. *Nonlinear Modeling and Forecasting*. Addison Wesley, 1992.

[5] C. Chatfield. Model uncertainty, data mining and statistical inference (with discussion). *Journal of the Royal Statistical Society, Series A*, 158:419–466, 1995.

[6] D. Collett and T. Lewis. The subjective nature of outlier rejection procedures. *Applied Statistics*, 25:228–37, 1976.

[7] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic betworks. *Proc. of UAI-98*, pages 139–147, 1998.

[8] I. Haimowitz and Kohane. Automated trend detection with alternate temporal hypotheses. *Proc. of the 13th International Joint Conference on Artificial Intelligence*, pages 146–151, 1993.

[9] D. J. Hand. Intelligent data analysis: Issues and opportunities. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis: Reasoning about Data, LNCS 1280*, pages 1–14. Springer-Verlag, 1997.

[10] E. J. Hannan. *Multiple Time Series*. Wiley, 1970.

[11] W. Horn, S. Miksch, G. Egghart, C. Popow, and F. Paky. Effective data validation of high frequency data: Time-point, time-interval and time-based methods. *Computers in Biology and Medicine*, 27:389–409, 1997.

[12] P. J. Huber. *Robust Statistics*. Wiley, 1981.

[13] M. Kadous. Learning comprehensible descriptions of multivariate time series. In *Proc. of the International Conference on Machine Learning*, pages 454–463. Morgan Kaufmann, 1999.

[14] E. Keravnou. Modelling medical concepts as time objects. In *Artificial Intelligence in Medicine - LNAI, P Barahona, M Stefanelli ands J Wyatt, editors*, pages 67–90. Springer-Verlag, 1995.

[15] E. Knorr and R. Ng. A unified notion of outliers: Properties and computation. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Proc of the 3rd Int. Conf. on Knowledge Discovery and Data Mining*, pages 219–22. AAAI Press, 1997.

[16] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 1995.

[17] W. Lam and F. Bachus. Learning bayesian networks: An approach based on the mdl principle. *Computational Intelligence*, 10(3):269–293, 1994.

[18] C. Larizza, A. Moglia, and A. Riva. M-htp: A system for monitoring heart transplant patients. *Artificial Intelligence in Medicine*, 4:111–126, 1992.

[19] N. Lavrac, E. Keravnou, and B. Zupan. *Intelligent Data Analysis in Medicine and Pharmacology*. Kluwer, 1997.

[20] X. Liu. Intelligent data analysis: issues and challenges. *The Knowledge Engineering Review*, 11:365–371, 1996.

[21] X. Liu, G. Cheng, and J. Wu. Noise and uncertainty management in intelligent data modeling. *Proc. of the 12th National Conference on Artificial Intelligence*, pages 263–268, 1994.

[22] X. Liu, G. Cheng, and J. Wu. Ai for public health: Self-screening for eye diseases. *IEEE Intelligent Systems*, 13:5:28–35, 1998.

[23] H. Lutkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, 1993.

[24] T. Oates, M. Schmill, and P. Cohen. Efficient mining of statistical dependencies. *Proc. of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[25] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[26] A. Pole, M. West, and P. Harrison. *Applied Bayesian Forecasting and Time Series Analysis*. Chapman-Hall, 1994.

[27] P. Sebastiani, M. Ramoni, P. Cohen, J. Warwick, and J. Davis. Discovering dynamics using bayesian clustering. In D. J. Hand, J. Kok, and M. Berthold, editors, *Advances in Intelligent Data Analysis (IDA-99) LNCS 1642*, pages 199–209. Springer-Verlag, 1999.

[28] Y. Shahar. A framework for knowledge-based temporal abstraction. *Artificial Intelligence*, 90:79–133, 1997.

[29] S. Swift and X. Liu. Modelling and forecasting of glaucomatous visual fields using genetic algorithms. In *Proc of the Genetic and Evolutionary Computation Conference*, pages 1731–1737. Morgan Kaufmann, 1999.

[30] A. Tucker, X. Liu, and A. Ogden-Swift. Learning dynamic probabilistic models with large time lags. In *Techical Report BBKCS-9904, Computer Science, Birkbeck College*. London, 1999.

[31] A. S. Weigend and N. A. Garshenfeld. *Time Series Prediction*. Addison-Wesley, 1994.

[32] J. Wu, G. Cheng, and X. Liu. Reasoning about outliers by modelling noisy data. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis (IDA-97) LNCS 1280*, pages 549–558. Springer-Verlag, 1997.