Automating Time Series Analysis - A Case-based Reasoning Approach and Web Services

Branko Pecar, PhD

University of Gloucestershire
bpecar@glos.ac.uk

Abstract. The method developed and described in this paper departs from the traditional time series analysis approach. The starting premise is that any time series can be broken down into a number of characteristic cases, each of which potentially holds the key for indicating the value of the subsequent observation. The case that constitutes the beginning of the forecasting horizon (the reference case) is compared with all the past cases and the best-case match is identified. The differences between the best historical case and the reference case are used for predicting the value through the forecasting horizon. This approach, generally considered to be a case-based reasoning approach, preserves the dynamics and the texture of the original series. It enables development of a fully automated and computerised system, free of any models and assumptions. In addition to this, if coupled with the web services technology, it provides an ideal collaborative tool in a distributed environment.

Keywords. Case-based reasoning approach, time series analysis, pattern recognition, APRE method, collaborative forecasting, web services

1. Introduction

Most of the methods deployed for time series analysis and extrapolation fall in the rules-based category. A number of different approaches have been successfully applied, such as classical decomposition [18], exponential smoothing [5] [11] [26], stochastic modelling [4], state space modelling [10] [14] and Bayesian models [13], to mention some of the traditional approaches. More contemporary methods have roots in neural networks [20] and fuzzy logic [2] [24] [29], i.e. artificial intelligence domain, or genetic algorithms [8]. Pattern recognition as a subset of artificial intelligence also contributed to this subject area. However, the common thread that flows through all the above approaches is some sort of a rule approach. An assumption is made that a time series belongs to one of the general processes, or probability distribution is assumed, or errors are measured and used as a rule for correction, etc. This paper will attempt to depart from this generic approach to time series analysis and introduce a different treatment of time series, namely a case-based reasoning approach. Once a new approach has been developed, a web services approach to deployment of the solution will be discussed.

An alternative approach to rigorous time series analysis could be traced back to Yakowitz's nearest neighbour method [28], or Singh's PMRS[1] [22], which has origins in the paper published by Sket-Motnikar, Pisanski & Cepar [23]. Although there is some resemblance between the method described in this paper and other methods such as the pattern-imitation method by Sket-Motnikar, Pisanski & Cepar, PMRS by Singh and the original *K* nearest neighbour method, the APRE method described in this paper has been developed completely independently and contains some unique features.

[1] Pattern Modelling and Recognition System.

2. Case-based Reasoning as an Alternative to Rules Approaches

A time series is a snapshot, contained in a time window, of the historical behavioural pattern of the observed variable. The clues about the direction and the dynamics of the future behaviour of the variable are hidden in the time series. If a time series represents a historical behavioural pattern, than it inevitably consists of miniature instances that define this pattern. Predictably, these instances could be called cases. Every instance, i.e. every case, holds the key for the future behaviour of the time series. In other words, if we know the case that currently characterises the variable and there is a precedent case, we can predict what the next move is likely to be.

This line of thinking inevitably leads towards the case-based reasoning (CBR) approach to problem solving. In this paradigm, specific knowledge about the behaviour of the variable is implicitly embedded in individual cases. This approach can be successfully deployed for time series extrapolation purposes, but before we define specific interpretation of cases in the time series context, we need briefly to remind ourselves of some of the starting premises of CBR.

CBR assumes that the library of past cases holds the expertise about the system behaviour, rather than encoding this behaviour by a series of rules. If we can identify (match) past cases with the current case, we have the foundation for predicting the future outcomes. CBR usually follows the process of **retrieving** similar cases, **reusing** the retrieved cases, **revising** the solution and **retaining** the solution. It is irrelevant whether a specific approach to CBR is based on trivial syntactic similarities, or more complex semantic ones. The process is the same.

3. Definition of Cases in the Time Series Context

In CBR terminology a case is a problem situation [1], which effectively implies that we could split a time series into smaller pattern sequences. For example, we could break the series down into a sequence of three rolling observation patterns. In other words, the last interval in a series of, say, three observations, could contain x_n, x_{n-1} and x_{n-2}, the one before the last one x_{n-1}, x_{n-2} and x_{n-3}, and so forth until we reach x_3, x_2 and x_1. However, in order to generalise the method and make it capable of handling and comparing various patterns (to accommodate for the presence of nonstationarity and heteroscedacity, among other things), we need to define cases in somewhat more general terms. Equally, case features need to be defined.

Rather than handling actual observations from the series, we could use a linguistic equivalent to describe the series dynamics. We say that every observation in the pattern, in relation to the previous observation, can go up (P for positive move), down (N for negative move) or stay on the same level (Z for a zero move)[2]. This implies that we can identify intervals that consist of a series of three-observation patterns, something like PPN, PNN, NNZ, NZZ, ZZP, ZPP, etc. Each of these patterns constitutes a case. If we analyse all the identical or similar cases, we'll probably discover that they are usu-

[2] P, N and Z are equivalent to Singh's binary patterns [22].

ally followed by a similar move. This implies that some form of similarity measure needs to be established.

The similarity measure we intend to use in this paper is defined as the minimum distance between two vectors:

$$dj = \min \left\| y - y_j \right\|$$

Computationally this measure is implemented as:

$$d_{i,j}^{m} = \sum_{k=1}^{r} (y_{i,k} - y_{j,k})^{m}$$

Providing that $m=1$, and the above differences are taken as absolute values, this measure becomes a standard Manhattan distance. For $m=2$, the measure is conventional Euclidean distance. However, as we are potentially dealing with nonstationary time series, the measure was standardised by dividing the differences by their standard deviation.

Returning to the issue of the case formation, we have to say that there is no rigorous method to define what a typical case is, in other words, whether it should consist of only 2 rolling observations, 3, 4 or more. In accordance with the CBR approach the best way is to suspend judgement and allow the coexistence of different cases. In practical terms this implies forming a library of 2, 3, 4, 5, etc. pattern observations and treating every group as a case category. For the sake of convenience, to speed up the computing time and restrict the storage requirements, we restricted ourselves to a maximum of 12 observations in a pattern. As this corresponds with the number of months in a year, the assumption is that cases are also capable of detecting seasonal variations.

4. Algorithmic Approach to Case Matching in Time Series

To deploy the ideas we initiated above, an alternative method can be developed. A form of an **A**lgorithmic approach to time series **P**attern **R**ecognition and **E**xtrapolation[3], founded on the case-based reasoning approach. The variables are defined as follows:

n = number of observations in the series
x_n = the n^{th} observation in the series
j = total number of cases containing r elements
r = number of elements in every case
$t_{j,r}$ = specific historical case
c = forecasting origin (the beginning of the ex-post forecasts)
$t_{c,r}$ = reference case
$d_{c,j}$ = case similarity measure (case distance between identical case patterns)

The method (algorithm) consists of several steps.

[3] We will refer to it in the text as the APRE method

1. Break the series down into r-interval patterns consisting of sequential rolling observations. Each pattern is used as a basis for the case formation.

For j=2, … , n we first find differences:

$$m_j = x_j - x_{j\text{-}1} \tag{1}$$

The letters P, N or Z are assigned according to the value of difference:

$$m_j = \begin{cases} \text{"P"} & & m_j > 0 \\ \text{"N"} & \text{for} & m_j < 0 \\ \text{"Z"} & & m_j = 0 \end{cases} \tag{2}$$

For j = 2, 3, …, (n - r + 1) and r = 2 to 12, let $t_{j,r}$ represent cases with a different number of observation differences:

$$t_{j,r} = \{m_j\} \tag{3}$$

2. Store all the cases and case features.

3. Decide the forecasting origin c and take the last case preceding the forecasting origin as the reference case. The case has an arbitrary number of observation differences in the interval.

$$t_{c,r} = \{m_j\} \tag{4}$$

4. Retrieve all the similar cases from the past, providing strictly that the pattern is identical to the reference case pattern, i.e. for c = 1, … , n and j = 2 to c-1 where:

$$t_{c,r} = t_{j,r} \tag{5}$$

Where index c represents the beginning of the ex-post forecasting horizon

5. Establish relevant distances (maximum similarity between the cases) by calculating:

$$d^2_{c,j} = \sum_{k=1}^{r} \left(\frac{x_{c,k} - x_{j,k}}{s_c} \right)^2 \tag{6}$$

6. Where the retrieved matching case has the value of $d_{c,j}$ = MIN, the value of the observation difference m_j, succeeding the case t_j can be used for predictions.

$$x_c = x_{c\text{-}1} + m_j \tag{7}$$

7. Once we have reached the end of the actual series, the future forecasting horizon is calculated in the same fashion. The only difference is that newly generated observations (ex-ante forecasts) become part of the new reference case. However, only the stored cases coinciding with the actual time series are used for comparison.

The above steps incorporate all the features of a classical case-based reasoning approach, i.e., how the cases and case features are defined, what the similarity criterion is, how the cases are retrieved, how the learning takes place, the reusability of the cases, the revision and finally the prediction principles.

5. An Example of the APRE Method

A detailed VB code that was compiled to automate the APRE method[4] and a comparative evaluation of this method vs. several other well-established time series analysis methods was rendered[5]. The APRE method figured favourably and showed remarkable ability to extrapolate the underlying dynamics of the series for a very long ex-ante forecasting horizon.

To demonstrate the characteristics of the APRE method, an example using the Lorenz attractor as one of the best-known chaotic time series was considered here. Only 200 values of Lorenz time series were generated and ex-post forecasts starting from the 150th observation were made. The cases were varied from 2 to 12 patterns per case, and the best one was selected by calculating the Mean Error. Fig. 1 below shows how well the original series was approximated and the future 20 observations were rendered.

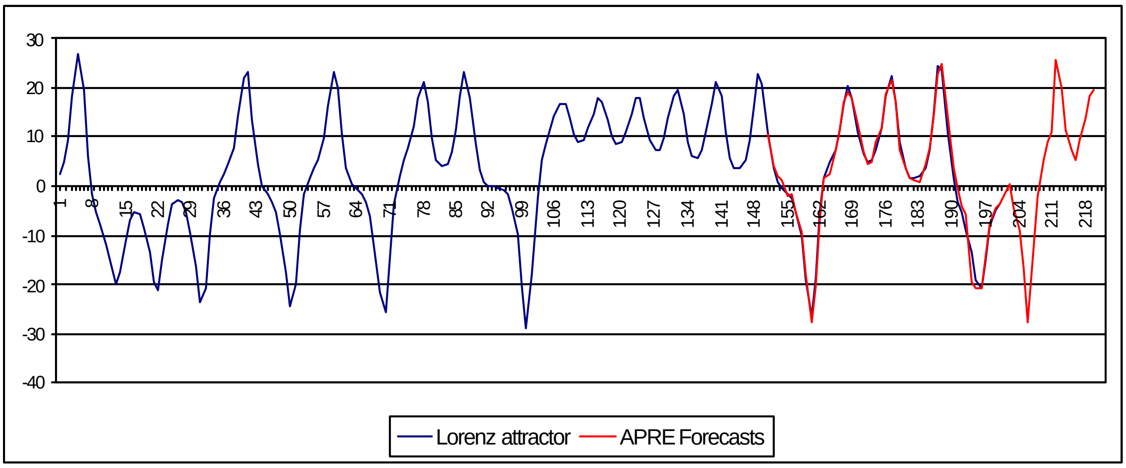

Fig. 1. Forecasts using the APRE method

The case that best approximated the original Lorenz series was based on cases with four patterns (observation differences) only. A further 20 values of Lorenz attractor were calculated to compare them with the APRE method extrapolations. This comparison is shown in Fig. 2.

Just a visual inspection indicates several conclusions. The method shows good accuracy, but fails to produce precise forecasts. It has to be said that here only 200 observations were used. There is no doubt that the method would perform much better if several

[4] The author welcomes any interest in the code (embedded in an Excel spreadsheet) and can be contacted via email.

[5] For details see [19]

thousands of observations were used. However, to make this example more comparable with business cases, where it is more likely there will be several hundred observations present rather than several thousand, it was felt that this was a fair representation.

A more striking conclusion from observing the above graph is that the dynamics of the Lorenz attractor seem to be very well captured by the APRE method and successfully extrapolated in the future. This is a rather unique feature. Virtually all the rules based methods fail to extrapolate adequately the dynamics of the series. They usually stick to either a linear extrapolation, or follow some of the smooth curves (polynomial of varying degree, sinusoid, etc.). The APRE method seems to display a unique ability to 'mimic' the appearance of the series.

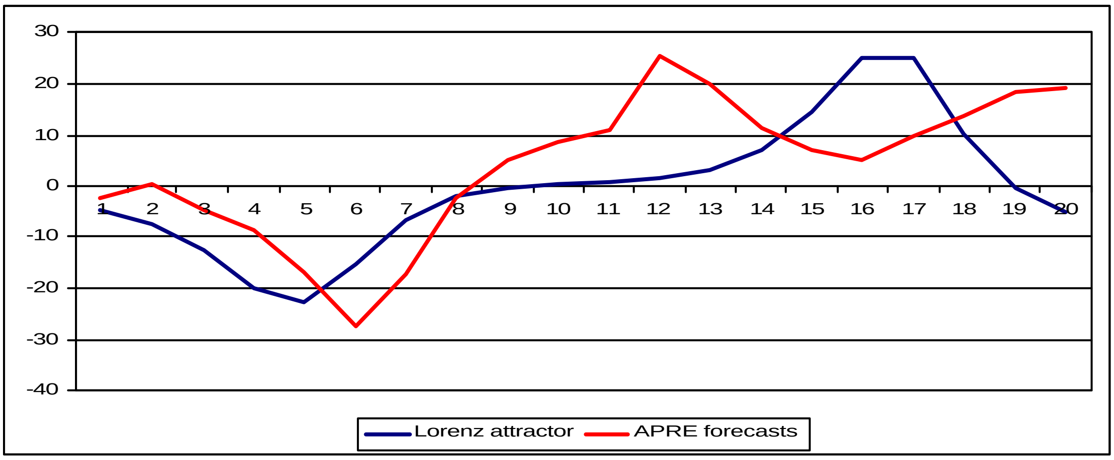

Fig. 2. Comparison between the actual future values of Lorenz attractor and APRE forecasts

6. Applying Case-based Reasoning Approach to Forecasting

As indicated earlier, to apply any of the rules-based methods, an intimate knowledge of the method's know-how is necessary. Equally, certain assumptions about the series characteristics are often made. Although most of the methods have been automated and computerised, the fact remains that such software packages are often restricted to certain families of methods only, or they require significant human intervention and judgement to get valid results.

Case-based reasoning is based on remembering previous experiences [16], implying that no explicit rule or assumption is necessary. In general, past cases hold the key to solving the new cases, despite the fact that they might not be identical, only similar. This also implies that some form of learning is embedded in case-based reasoning and this learning can take place without any human intervention (i.e. machine-based learning).

Using a specific algorithm to this approach of reasoning, which we called the APRE method, some additional benefits surfaced. The APRE method displays all the advantages of case-based reasoning and, specifically in the context of time series analysis, enables the propagation of the internal dynamics of the series to its ex-ante forecasts. This feature is quite unique among the time series analysis methods. Most of the forecasting methods are capable of fitting reasonably well the past movements of the series (ex-post forecasts). However, when extrapolated more than one observation in the future, they resort to smooth lines (linear or any of the polynomials). In practise, this makes the medium to long-term forecasts only marginally indicative.

Accepting that case-based reasoning approach to forecasting does not require any forecasting methodology knowledge, or any prior knowledge of the time series as a prerequisite, this makes it an ideal candidate for automated forecasting in the collaborative environment. Hundreds of series could be handled without manual intervention. Typically, participants in the supply chain would all use their own proprietary forecasting techniques, implying that even if they all handled the same time series, the results are more than likely to be different from one participant to another. With the neutral approach based on historical performance, such as the one advocated in this paper, the uniformity of results is guaranteed. However, to deploy this approach in a collaborative environment, a new architecture is needed.

A new architecture, called web services, promising to enable collaboration between disparate systems started to emerge not so long ago. In 2002 The Web Services Interoperability Organisation (WS-I) was formed, whose mission is broadly defined as "… an open industry effort chartered to promote web services interoperability across platforms, applications, and programming languages" [27]. The founding members of WS-I are some of the best-known names in the information technology community, such as Microsoft, IBM, SAP, Sun Microsystems, Intel, HP etc. Currently more than 135 industry leaders are members of the WS-I community, actively participating in the work of this organisation.

One can think about web services as software components that operate as either web objects or web applications. What is characteristic for them is that they are self-contained, self-describing and modular. They can be published, located and invoked across the web. Once web service is deployed, other applications (and other web services) can discover and invoke the deployed service [25]. This makes them ideal candidates for implementing case-based reasoning forecasting in a collaborative environment.

7. Web Services Technologies

World Wide Web Consortium (W3C) defines web service as a software system identified by a URI (Uniform Resource Identifier), whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols [30].

Unlike current component technologies, web services are not accessed via object-model-specific protocols, such as the distributed Component Object Model (DCOM), Remote Method Invocation (RMI), or Internet Inter-ORB Protocol (IIOP). Instead, web services are accessed via ubiquitous web protocols and data formats, such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). Furthermore, a web service interface is defined strictly in terms of the messages the web service accepts and generates. Consumers of the web service can be implemented on any platform in any programming language, as long as they can create and consume the messages defined for the web service interface [31]. In order to achieve the objectives stated by the above definitions, some fundamental technologies are necessary. These fundamental technologies come in the guise of XML, as well as the SOAP, WSDL and UDDI protocols.

XML stands for eXtensible Markup Language, which is a more versatile version of the HTML language. HTML is a static language that can only display information in a predefined format. XML does the same, i.e. is a standard way to represent data, but it goes beyond presentation. It also provides means of describing the data. In a way, it is a meta-language for document and programme definition and exchange. It has been developed since 1996 and launched in 1998 by the Worldwide Web Consortium (W3C).

SOAP, or Simple Object Access Protocol[6], is a protocol that enables exchange of structured information via XML encoding. SOAP defines the rules for how to use XML to represent data as well as how to represent remote procedure calls (RPC), which enable disparate applications to interact with one another. Currently SOAP is bound only to HTTP protocol, to maximise its usage and encourage the adoption, but in the future other protocols will be used too. Briefly, SOAP is a standard format for communicating with web services.

WSDL stands for Web Services Description Language. This is an XML based language that describes what messages and requests a web service will accept and how it will respond to them. To an extent, this is a standardised XML vocabulary description layer[7].

UDDI, or Universal Description, Discovery and Integration[8], is a method of discovering the existence of a web service. It describes how the discovery document format is structured (in XML) and where to find the service. This specification is often described as the Yellow Pages, or a public registry, for advertising and locating a service.[9]

The technologies described above enable the facilitation of a new approach called Service-Oriented Architecture.

[6] More recently referred to as Service Oriented Access Protocol
[7] Examples of WSDL files can be fount at http://www.xmethods.net, a public repository of web services
[8] Sometimes the acronym is interpreted as Universal Directory Discovery Interface
[9] Details about individual specification are available from a number of web sites, primarily from: http://www.w3.org/2002/ws/; http://www.w3.org/XML/, http://www.w3.org/TR/SOAP/; http://www.w3.org/TR/wsdl/; http://www.uddi.org/ and http://www.webservices.org/

8. Service-Oriented Architecture

Software architecture is an abstraction of the run-time elements of a software system during some phase of its operation. A system may be composed of many levels of abstraction and many phases of operation, each with its own software architecture [7]. The way web services have emerged as a new service-oriented architecture (SOA) is often depicted as per Fig.3 [12].

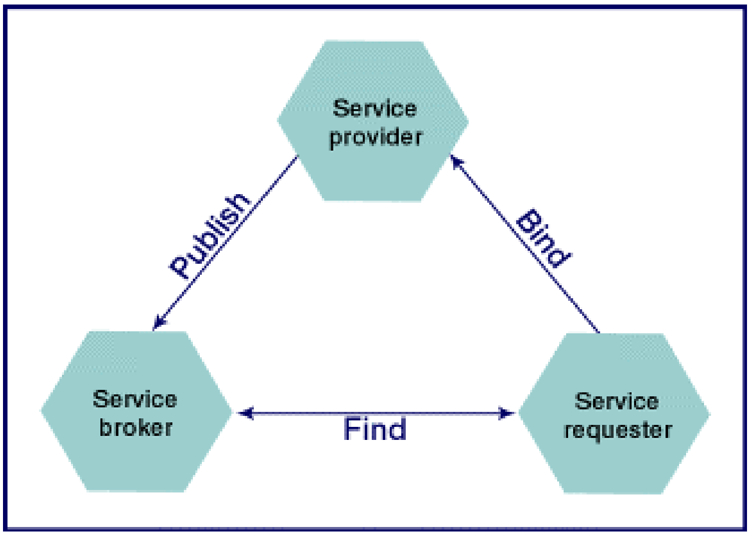

Fig. 3. Web services architecture

The diagram in Fig.3 communicates a possibility of placing the web services into one of the three available categories. The Service Provider provides a service interface for a software asset and publishes its capabilities using the WSDL document. The Service Broker consists of a registry listing the existence of the Service Provider. This is established by using the UDDI documents. Service Requester is an application that, using the SOAP protocol, binds and invokes the service offered by the Service Provider.

This effectively means that the whole Internet with all the publicly available applications that are 'wrapped' in web services, could be used by anybody who uses the same specification. Without any special coding, users do not have to run their applications just from their desktops, or their LANs, but the whole Internet becomes a repository of numerous and diverse, yet publicly available, applications.

9. Conclusion

The APRE method departs from the usual rule-based approach to time series analysis and introduces a particular form of case-based reasoning. The APRE method provides a valid alternative to most commonly used rule-based or model-free methods for time series analysis. The forecasting horizon is not limited by the properties of the method, as with most rule-based methods, but by the richness of cases and decision-making context. The ex-ante forecasts produced using the APRE method preserve the texture and the complexity of the original data set, making it unique among the methods. It contains automated learning ability, common to other case-based reasoning methods, making it an ideal candidate for a neutral forecasting platform in a collaborative environment.

Recent developments on the web services front indicate that such case-based reasoning methods could be easily deployed in a collaborative environment. Without a major integration effort, this technology enables applications to talk to one another without human intervention. This enables any database or application to send time series to a web based collaborative forecasting engine and receive the results, ready to process them for other internal purposes.

Further investigation in how to synchronise and optimise activities between various supply chain participants sharing such a collaborative forecasting engine would seem appropriate.

References

1. Aamodt Agnar & Enric Plaza: *Case Based Reasoning: Foundational Issues, Methodological Variations and Systems Approach*, AICom – Artificial Intelligence Communications, Vol. 7, No. 1, March 1994

2. Altrock von Constantin: *Fuzzy Logic and Neuro Fuzzy Applications Explained*, Prentice Hall, Englewood Cliffs, 1995

3. Barnsley M.F., Devaney R.L., Mandelbrot B.B., Peitgen H.-O., Saupe D. and Voss R. F.: *The Science of Fractal Images*, Springer-Verlag, New York, 1988.

4. Box E.P. George & Gwilym M. Jenkins: *Time Series Analysis: Forecasting and Control*, Holden Day, San Francisco, 1970.

5. Brown Robert: *Statistical Forecasting for Inventory Control*, McGraw Hill Book Co., New York, 1958.

6. Dooley Kevin: “Complexity in Time Series Modelling”, Society for Chaos Theory in Psychology and Life Sciences, 2(3), 1994.

7. Fielding, Roy Thomas: “Architectural Styles and the Design of Network-based Software Architectures”, University of California, Irvine, USA, Doctoral dissertation, 2000, available from <URL:http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

8. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

9. Grassberger Peter & Procaccia Itamar: “Measuring the Strangeness of Strange Attractors”, *Physica* 9D, pp. 189-208, 1983.

10. Harvey C. Andrew: *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge, 1989.

11. Holt C.C.: *Forecasting Seasonal and Trends by Exponentially Weighted Moving Averages*, Carnegie Institute of Technology, Pittsburgh, Pennsylvania, 1957.

12. IBM Web Services Architecture Team "Web Services architecture overview", 2000, available from <URL:http://www-106.ibm.com/developerworks/webservices/library/w-ovr/>

13. Jensen V. Finn: *An Introduction to Bayesian Networks*, UCL Press, London, 1996.

14. Kalman R.E.: "A new Approach to Linear Filtering and Prediction Problems", *Journal of Basic Engineering*, D. 82, 1960.

15. Kantz Holger & Thomas Schreiber: *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge, 2000.

16. Kolodner Janet: "Introduction to Special Issues on Case-Based Reasoning", *Machine Learning*, Vol. 10, Kluwer Academic Publishers, Boston, 1993

17. Mandelbrot B. Benoit: *The Fractal Geometry of Nature*, W H Freeman, New York, 1983.

18. McLaughlin R.L.: "Time Series Forecasting", *Marketing Research Technique*, Series No. 6, AMA, 1962.

19. Pecar, B: "Case-Based Algorithm for Pattern Recognition and Extrapolation", 7th UK Workshop on Case Based Reasoning, 22nd SGES/SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence, Cambridge, 2002

20. Rumellhart, Hinton & Williams: "Learning representations by back-propagating errors", *Nature*, Vol. 323, 1986.

21. Schroeder M.: *Fractals, Chaos, Power Laws*, W H Freeman, New York, 1991.

22. Singh Sameer: "Multiple Forecasting using Local Approximation", *Pattern Recognition*, Vol. 33, 2000.

23. Sket-Motnikar B., Pisanski T. & Cepar D.: "Time-series forecasting by pattern imitation", *OR Spektrum* 18, pp. 43-49, 1996.

24. Tsoukalas H. Lefteri & Uhrig E. Robert: *Fuzzy and Neural Approaches in Engineering*, John Wiley, New York, 1997.

25. Vasuvedan, Venu: A Web Services Primer, O'Riley XML.com, April 04, 2001, available from <URL:http://www.xml.com/pub/a/2001/04/04/webservices/index.html?page=1>

26. Winters R. Peter: ”Forecasting Sales by Exponentially Weighted Moving Averages”, *Management Science,* April 1960.

27. WS-I FAQ, December 2002, available from <URL:http://www.ws-i.org/ FAQ.aspx#A02>

28. Yakowitz, S: “Nearest-neighbour method for time series analysis”, *Journal of Time Series Analysis* 8, 235-247

29. Zadeh A. Lotfi, Fu King-Sun, Tanaka Kokichi & Shimura Masamichi (eds): *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, Academic Press, New York, 1975.

30. <URL:http://www.w3.org/TR/2002/WD-wsa-reqs-20021114>

31. <URL:http://msdn.microsoft.com/library/default.asp?url=/library/enus/dnwebsrv/ html/ websvcs_platform.asp>