Graph Based Framework for Time Series Prediction

Vivek Yadav[*]
Durga Toshniwal[**]

Abstract

Purpose: *A time series comprises of a sequence of observations ordered with time. A major task of data mining with regard to time series data is predicting the future values. In time series there is a general notion that some aspect of past pattern will continue in future. Existing time series techniques fail to capture the knowledge present in databases to make good assumptions of future values.*

Design/Methodology/Approach: *Application of graph matching technique to time series data is applied in the paper.*

Findings: *The study found that use of graph matching techniques on time-series data can be a useful technique for finding hidden patterns in time series database.*

Research Implications: *The study motivates to map time series data and graphs and use existing graph mining techniques to discover patterns from time series data and use the derived patterns for making predictions.*

Originality/Value: *The study maps the time-series data as graphs and use graph mining techniques to discover knowledge from time series data.*

Keywords: *Data mining; Time Series Prediction; Graph Mining; Graph Matching*
Paper Type: *Conceptual*

Introduction

Data mining is the process of extracting meaningful and potentially useful patterns from large datasets. Nowadays, data mining is becoming an increasingly important tool by modern business processes to transform data into business intelligence giving business processes an informational advantage to make their strategic business decisions based on the past observed patterns rather than on intuitions or beliefs **(Clifton, 2011)**. Graph based framework for time series prediction is a step towards exploring new efficient approach for time series prediction where predictions are based on patterns observed in past.

Time Series data consists of sequences of values or events obtained over repeated instances of time. Mostly these values or events are collected at equally spaced, discrete time intervals (e.g., hourly, daily, weekly, monthly, yearly etc.). When there is only one variable upon which observations with respect to (w.r.t) time are made, is called univariate time series. Data mining on Time-series data is popular in many applications, such as stock market analysis, economic and sales forecasting, budgetary analysis, utility studies, inventory studies, yield

[*] *Department of Electronics & Computer Engineering, IIT Roorkee.*
email: viv20pec@iitr.ernet.in
[**] *Assistant Professor. Department of Electronics & Computer Engineering, IIT Roorkee.*
email: durgafec@iitr.ernet.in

projections, workload projections, process and quality control, observation of natural phenomena (such as atmosphere, temperature, wind, earthquake), scientific and engineering experiments, and medical treatments **(Han & Kamber, 2006)**.

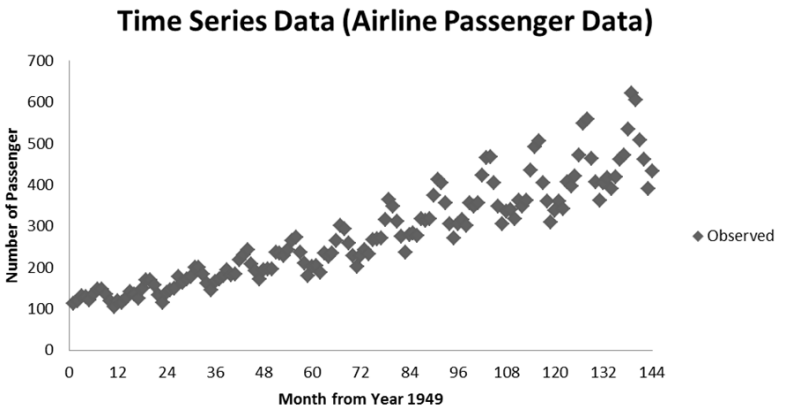
Time series dataset constitutes of $\{Y_1, Y_2, Y_3, \ldots, Y_t\}$ values, where each Y_i represent the value of variable under study at time *i*. One of the major goal of Data mining in the time series is forecasting the time series i.e., to predict the future value Y_{t+1}. The successive observations in time series are statistically dependent on time and time series modeling is concerned with techniques for analysis of such dependencies. In time series analysis, a basic assumption is made that is (i.e.) some aspect of past pattern will continue in future. Under this assumption time series prediction is assumed to be based on past values of the main variable *Y*. The time series prediction can be useful in planning and measuring the performance of predicted value on past data against actual observed value on the main variable Y.

Time series modeling is advantageous, as it can be used more easily for forecasting purposes since the historical sequences of observations upon study on main variable are readily available as they are recorded in the form of past observations & can be purchased or gathered from published secondary sources. In time series modeling, the prediction of values for future periods is based on the pattern of past values of the variable under study, but the model does not generally account for explanatory variable which may have affected the system. There are two reasons for resorting to such time models. First, the system may not be understood, and even if it is understood it may be extremely difficult to measure the cause and effect relationship of parameters affecting the time series. Second, the main concern may be only to predict the next value and not to explicitly know why it was observed **(Box, Jenkins & Reinsel, 1976)**

Time Series analysis consists of four major components for characterizing time-series data **(Madsen, 2008)**. First, Trend component- these indicate the general direction in which a time series data is moving over a long interval of time, denoted by *T*. Second, Cyclic component- these refer to the cycles, that is, the long-term oscillations about a trend line or curve, which may or may not be periodic, denoted by *C*. Third, Seasonal component- these are systematic or calendar related, denoted by *S*. Fourth, Random component- these characterize the sporadic motion of time series due to random or chance events, denoted by *R*. Time-series modeling is also referred to as the decomposition of a time series into these four basic components. The time-series variable *Y* at the time *t* can be modeled as either the product of the four variables at time *t* (i.e., $Y_t = T_t \times C_t \times S_t \times R_t$) using multiplicative model proposed by **(Box, Jenkins &**

Reinsel, 1970) where T_t means Trend component at time t, C_t means cyclic component at time t, S_t means seasonal component at time t and R_t signifies Random component at time t. As an alternative, additive model **(Balestra & Nerlove, 1966; Bollerslev, 1987)** can also be used in which ($Y_t = T_t+C_t+S_t+R_t$) where Y_t, T_t, C_t, S_t, R_t have the same meaning as described above. Since multiplicative model is the most popular model, we will use it for the time series decomposition. Example of time series data is the airline passenger data set **(Fig. 1)** in which the main variable Y is the number of passengers (in thousands) in an airline is recorded w.r.t time, where each observation on main variable is recorded on monthly basis from January 1949 to December 1960. Clearly, the time series is affected by increasing trend, seasonal and cyclic variations.

Fig. 1: Time series Data of the Airline Passenger Data from Year 1949 to 1960 represented on monthly basis.



Review of Literature

In time series analysis there is an important notion of de-seasonalizing the time series **(Box & Pierce, 1970)**. It makes the assumption that if the time series represents a seasonal pattern of L periods, then by taking moving average M_t of L periods, we would get the mean value for the year. This would be free of seasonality and contain little randomness (owing to averaging). Thus $M_t=T_t \times C_t$ **(Box, Jenkins & Reinsel, 1976).** To determine the seasonal component, one would simply divide the original series by the moving average i.e., $Y_t/M_t= (T_t \times C_t \times S_t \times R_t)/(T_t \times C_t)= S_t \times R_t$. Taking average over months eliminates randomness and yields seasonality component S_t. De-seasonalized Y_t time series can be computed by Y_t/S_t.

The approach described in **(Box, *et al*, 1976)** for predicting the time series, uses regression to fit a curve to De-seasonalized time series using

least square method. To predict the values in time series, model projects the De-seasonalized time series into future using regression and divide it by the seasonal component. The Least Square Method is explained in **Johnson and Wichern (2002).**

Exponential Smoothing has been proposed in **(Shumway & Stoffer, 1982)** which is an extension to above method to make more accurate predictions. It suggests, making prediction for Y_t weighing the most recent observation (Y_{t-1}) by α and weighting the most recent forecast (F_{t-1}) by ($1- \alpha$). Note α lies between 0 and 1 (i.e., $0 \leq \alpha \leq 1$). Thus the forecast is given by $F_{t+1} = Y_{t-1} * \alpha + (F_{t-1}) * (1- \alpha)$. Optimal α is chosen based on the smallest MSE (Mean Square Error) value during the training.

ARIMA (Auto-Regressive Integration Moving Average Based Model) has also been proposed **(Box, et al., 1970, 1976; Hamilton, 1989)**. ARIMA model is categorized by *ARIMA(p,q,d)* where *p* denotes order of auto-regression, *q* denotes order of differentiation and *d* denotes order of moving averages. The model tries to find the value of *p*, *q*, and *d* that best fits the data. In time series forecasting using a hybrid ARIMA and neural network model has proposed a model that tries to find *p*, *q* and *d* using neural network **(Zhang, 2003)**.

Proposed Work: Graph Based Framework for Time Series Prediction

In this paper, I propose to use graph based framework for time series prediction. The motivation to use the graphs is to capture the tacit historical pattern present in the dataset. The idea behind creation of graph over time series is to utilize two facts. First, some aspect of time series pattern will continue in future and graph is a data structure that is well suited to model a pattern. Second, similarity can be calculated between graphs to know the similar patterns and their order of occurrence. Thus, graph is created with the motivation to store a pattern over time series and make prediction based on similarity of observed pattern from historical data as an alternative to Regression and curve fitting. The major shortcoming of using the regression and curve fitting is that it requires expert knowledge about curve equation and the number of parameters in it. If parameters are too many there is problem of over fitting and if parameters are too less, model suffers from problem of under fitting **(Han & Kamber, 2006).** The complete pattern in time series is not known initially and it is affected by random component which makes the regression harder, hence deciding the curve equation and number of parameters in it is a major issue.

To further explore the concept of pattern, let there be time series on monthly data of *N* years where first observation was in first month of *m* year, Data = $\{Y_{1(k)} Y_{2(k)} \ldots Y_{12(k)}, Y_{1(k+1)} Y_{2(k+1)} \ldots Y_{12(k+1)}, \ldots, Y_{1(k+N)} Y_{2(k+N)} \ldots Y_{12(k+N)}\}$ where $Y_{1(k)}$ means value of variable under study for first month of year *k*

& $Y_{12(k+N)}$ means value of variable under study for twelfth month of year *k+N*. Note $m \le k \le (m+N)$. In general let *d*, be the time interval which makes a pattern. If a pattern has to be stored yearly and data is available monthly *d=12*, data is available quarterly *d=4*, etc. Each successive observation to Y_{ij} (meaning month *i* and year *j*) on main variable ordered by time is in general given by $Y_{i'j'}$ where if Y_{ij} $1 \le i \le 12$, $k \le j \le (k+N)$, then for $Y_{i'j'}$ if $i<12$ then $i'=i+1$, $j'=j$ else $i'=1$, $j'=j+1$. A graph over each successive *d* observation is created to store the pattern. This is called *'last-pattern-observed-graph'*. To make the prediction we also store the knowledge in each graph that how the last pattern observed effect the next observation. This is called *'knowledge-graph'*. Example If we consider the data $\{Y_{1(k)}Y_{2(k)}\ldots Y_{12(k)}, Y_{1(k+1)} Y_{2(k+1)} \ldots Y_{12(k+1)}, \ldots, Y_{1(k+N)} Y_{2(k+N)}\ldots Y_{12(k+N)}\}$, last-pattern-observed-graph for Jan of year *(k+1)* will be generated using data $\{Y_{1(k)}Y_{2(k)}\ldots Y_{12(k)}\}$ and knowledge-graph of Jan for year *(k+1)* will be generated using $\{Y_{1(k)}Y_{2(k)}\ldots Y_{12(k)}, Y_{1(k+1)}\}$ data. Knowledge graph is created with intuition to capture how the variable under study changed over last *d* observations and its effect on *d+1* observation.

In time series data, the graph is created with the motivation to model each observation as vertex and represent the effect of variation in observations with respect to time in form of edges. The number of vertices in graph is equal to time interval over which a pattern has to be stored. The edges are created to take into account the effect of each observation on other. Since the past values will affect the future values, but future values would not affect the past values and hence the edges are created between vertices corresponding to it and all the subsequent observations which measure the change in angle with horizontal. The graphs generated can be represented in computer memory either by using Adjacency matrix representation or Adjacency list representation **(Cormen, 2001)**. I have used Adjacency list representation to save the memory required to store the graph as each graph will have n(n-1)/2 edges thus space required will be n(n-1)/2 using adjacency list representation as compared to n^2 space using adjacency matrix representation.

Dataset of N tuples is partitioned into two sets. First set for training data of *m* tuples and second *{N-m}* tuples for training and validation of model. During the training phase, a Knowledge-Graph is generated over training data tuples over each subsequent *d+1* observation. $Y_{i(k)}Y_{(i+1)(k)}\ldots Y_{(i+12)(k)}, Y_{(i+13)(k)}$ where *i* has bounds $1 \le i \le 12$ and if $i>12$ then $i=1$ & $k=k+1$ for all m tuples in training Dataset. Thus *m-12* Knowledge-Graphs are generated. These generated graphs are partitioned into d sets *(d=12)*, where each graph is stored in the interval over which knowledge they have captured (i.e. graph for all Jan's are stored together, all Feb's stored together, etc.). To implement this we have used an array of size *d* of linked list of graphs.

Each linked list stores all the knowledge graph corresponding to interval over which knowledge it represents. The graphs are partitioned with the motivation to ease the search since while making prediction, model will query for all patterns observed w.r.t a particular month, since the graphs are already stored in partitioned form, time taken by model to execute this query will be *O(1)*.

To predict the next value in time series, model will take the last *d* known observations previous to the month on which prediction has to be done and compute 'last-pattern-observed-Graph'. The model will search for a Knowledge graph (stored in the partitioned form corresponding to month for which prediction has to be made) that is most similar to 'last-pattern observed graph', considering only number of vertices equal to 'last-pattern observed graph' in Knowledge-Graph. To compute the similarity between two graphs, graph-edit distance technique has been used **(Brown, 2004; Bunke & Riesen, 2008)**. The key idea of Graph-edit Distance approach is to model structural variation by edit operations reflecting modifications in structure and labeling. A standard set of edit operations is given by insertions, deletions, and substitutions of both nodes and edges. While calculating graph edit distance for time-series Graph for g_1 (source graph) & g_2 (destination graph), requires only substitutions of edges (change in angle) in g_2 to make it similar to g_1 and a summation of cost incurred with each edit operation is calculated. The graph with least edit cost is most similar & selected as a graph that will form the basis, of the prediction.

To make the prediction, model takes into account the structural difference between two graphs in vertex ordered weighted average manner. To make the prediction on graph g_1 (last-pattern-observed-Graph) using graph g_2 (Knowledge Graph which is most similar to g_1), every vertex in g_1 predicts the angle between itself and the predicted value using the knowledge of g_2 and taking into account the difference of edges between itself & it's corresponding vertex in g_2 in a weighted average manner (where edge difference to vertex that are closer to be predicted are given more weight technique to apply exponential smoothing in Graph based time series prediction approach), and thus in this way each vertex predicts the angle. Every vertex makes the prediction & the predicted value is average of value predicted by each vertex. After making the prediction, once the actual observed value is known, Knowledge graph is generated to capture the pattern corresponding to the last observation and in this way model learns in an iterative manner.

Experimental Results

The code to implement Graph Based Time Series prediction approach as discussed above is written in java. The Graph Based Time Series

prediction approach was applied on the airline passenger data set, which was first used in **(Brown & Smoothing, 1962)** and then in **(Box, et al., 1976)**. It represents the number of airline passengers in thousands observed between January 1949 and December 1960 on a monthly basis. I have used 2 years of data for training i.e., 1949 & 1950 and estimated the remaining data on monthly basis implementing iterative learning as an observation is recorded.

Fig. 2 represents Actual and Predicted number of Passenger using Graph Based Framework for Time Series prediction applied on the Time Series of airline passenger data set. **Fig. 3** represents the corresponding percentage error rate observed on monthly basis. The average error recorded on time-series is 7.05.**Fig. 4** represents the Actual and Predicted Number of passenger using Graph Based Framework for Time Series prediction applied on the De-seasonalized Time Series of airline passenger data set (using concept of Moving Average). **Fig. 5** represents the corresponding percentage error rate observed on monthly basis. The average percentage error recorded on De-seasonalized Time series is 5.81.

Fig. 2: Actual and Predicted number of Passenger using Graph Based Framework for Time Series prediction applied on the Time Series of airline passenger data set (APTS).

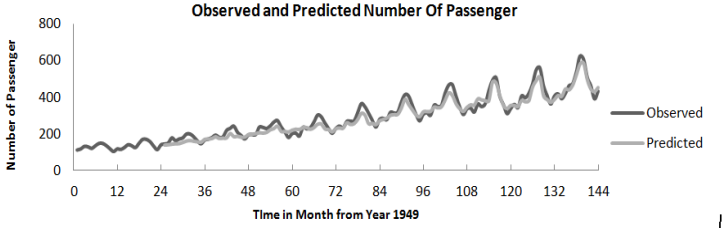


Fig. 3: Percentage Error between Actual and predicted using Graph Based Framework for Time Series prediction applied on the Time Series of airline passenger data set (APTS).

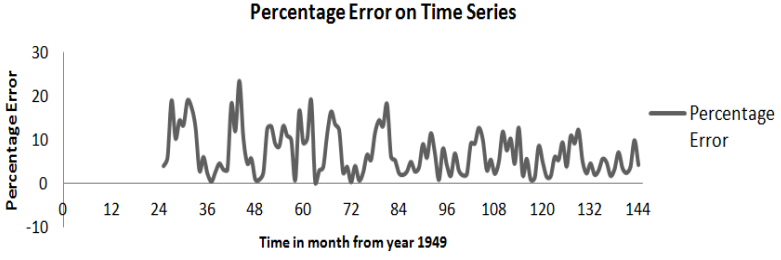


Fig. 4: Actual and Predicted number of Passenger using Graph Based Framework for Time Series prediction applied on the De-seasonalized Time Series of airline passenger data set (APTS).

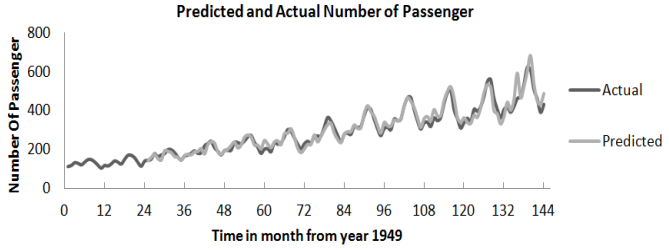
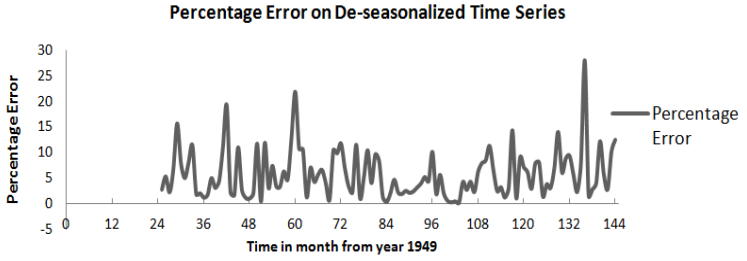


Fig. 5: Percentage Error between Actual and Predicted values using Graph Based Framework for Time Series prediction applied on the De-seasonalized Time Series of airline passenger data set (APTS).



Conclusion & Discussion

A new approach for time series prediction has been proposed & implemented which is based on graphs. The results reported show that using graph based framework for time series prediction on De-seasonalized Time Series *(Computed Using Concept of Moving Average)* on The Airline Passenger Data has 94.19 percent accuracy and on direct Time Series of The Airline Passenger Data has 92.95 percent accuracy. The accuracy on De-seasonalized time series is better since this time series has only two factors, cyclic and trend factors which leads to less error rate as compared to direct application of proposed approach on time-series which has all the four factors cyclic, trend, seasonal and randomness, which makes the prediction difficult. Thus application of Graph based framework in conjunction to Moving average offers good accuracy.

Graph based framework approach for time series prediction has incorporated the concept of exponential smoothing, moving average and graph mining to enhance its accuracy. Graph based framework approach for time series prediction is a good alternative to regression. In the proposed approach there is no need of domain expert knowledge to know the curve equation and number of parameters in it. The result validate that the new approach has good accuracy rate.

References

Balestra, P., & Nerlove, M. (1966). Pooling cross section and time series data in the estimation of a dynamic model: The demand for natural gas. *Econometrica, 34*(3), 585-612.

Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The review of economics and statistics, 69*(3), 542-547.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1970). *Time series analysis.* Oakland, CA: Holden-Day.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1976). *Time series analysis: forecasting and control* (Vol. 16): San Francisco, CA: Holden-Day.

Box, G. E. P., & Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association, 65*(332), 1509-1526.

Brown, R. G. (2004). *Smoothing, forecasting and prediction of discrete time series*. Mineola, NY: Dover Publications.

Brown, R. G., & Smoothing, F. (1962). Prediction of Discrete Time Series. *Englewood Cliffs, NJ*: Prentice Hall.

Bunke, H., & Riesen, K. (2008). Graph Classification Based on Dissimilarity Space Embedding. In N. da Vitoria Lobo, T. Kasparis, F. Roli, J. Kwok, M. Georgiopoulos, G. Anagnostopoulos & M. Loog (Eds.), *Structural, Syntactic, and Statistical Pattern Recognition (Vol. 5342, pp. 996-1007)*: Berlin / Heidelberg: Springer

Clifton, C. (2011). Data Mining. In *Encyclopaedia Britannica*. Retrieved from http://www.britannica.com/EBchecked/topic/1056150/data-mining

Cormen, T. H. (2001). *Introduction to algorithms*. Cambridge, Mass: The MIT press.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica, 57*(2), 357-384.

Han, J., & Kamber, M. (2006). *Data mining: concepts and techniques*: Morgan Kaufmann.

Johnson, R. A., & Wichern, D. W. (2002). *Applied multivariate statistical analysis* (Vol. 5): NJ: Prentice Hall Upper Saddle River.

Madsen, H. (2008). *Time series analysis*. Boca Raton: Chapman and Hall/CRC Press.

Shumway, R. H., & Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis, 3*(4), 253-264.

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159-175. doi: 10.1016/s0925-2312(01)00702-0