Association for Information Systems
AIS Electronic Library (AISeL)

ICEB 2002 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-10-2002

Dynamic Pattern Matching Using Temporal Data Mining for Demand Forecasting

Wen-Bin Yu

James H. Graham

Hokey Min

Dynamic Pattern Matching Using Temporal Data Mining for Demand Forecasting

Wen-Bin Yu
Dept. of Computer Engineering
and Computer Science
Speed Scientific School
University of Louisville
Kentucky, USA
weny@louisville.edu

James H. Graham
Dept. of Computer Engineering
and Computer Science
Speed Scientific School
University of Louisville
Kentucky, USA
jhgrah01@louisville.edu

Hokey Min
UPS Center for World Wide
Supply Chain Management
University of Louisville
Kentucky, USA
h0min001@louisville.edu

Abstract

Traditional time series methods are designed to analyze historical data and develop models to explain the observed behaviors and then predict future value(s) through the extrapolation from the models. The underlying premise is that the future values should follow the path of the historical data analyzed by the time series methods, and as such, these methods necessitate a significant amount of historical data to validate the model. However, this assumption may not make sense for applications, such as demand forecasting, where the characteristics of the time series may alter frequently because of the changes of consumers' behavior and/or cooperate strategies such as promotions. As the product life cycle gets shorter as it tends to be in today's e-business, it becomes increasingly difficult to make a forecast using traditional time series methods. In response to this challenge, this paper proposes a novel pattern matching procedure to decide whether one or combination of several patterns actually represents the development of the time series and then to use the patterns in forecasting. Several pattern transformation algorithms are also proposed to facilitate a flexible match. Rematching through dynamic reevaluation of the new data may be needed until the true development of the time series is discovered. Initial evaluation indicates superior performance in predicting the demand of a new product.

1. Introduction

Demand forecasting has attracted increasing attention because an accurate prediction of future demand dictates the success of supply chain planning. Various forms of demand forecasting techniques have been proposed in the literature including statistical approaches, decomposition methods and approaches borrowed from artificial intelligences. Statistical techniques, such as moving average (MA), autoregressive (AR) or integration of both methods i.e., auto regressive intergraded moving average (ARIMA) [2] regard times series as stochastic processes. Decomposition methods, such as Fourier transformation, convert the original time series into components in both time and frequency domains, and then predictions are applied to the individual components. In practice, simple methods such as moving average and exponential smoothing are frequently used due to their simplicity. Ironically, these simple methods might occasionally lead to more accurate forecasting results than sophisticated or complex methods. Recently, techniques in the area of artificial intelligence, such as neural networks and genetic algorithms [4, 5, 7], became popular in time series forecasting. However, all the traditional time series forecasting methods are established under the assumption that certain "behavior" exists in the time series that can be modeled systematically with past history. Even though researchers claim neural network as a "model-free" approach, the technique still relies on selected time series data to predict future events. Such methods would not be able to predict the "turning point", or detect sudden behavioral changes.

Almost all the methods require a significant amount of past historical data to build or "train" the model. The validity of the model would be doubtful when only very short history of time series is available. A good example is that, the life cycle of consumer products has become much shorter than ever before due to changes in business paradigm from mass production to semi-customization. As a result, a lot of products did not have long historical records to make a demand forecast with traditional forecasting methods.

The paper presents a pattern-matching algorithm for time series forecasting, especially in demand forecasting when very short history of time series is available. The objective is to take advantage of new methodologies in the field of temporal data mining and apply them to forecasting. Section two presents the new pattern-matching algorithm. Section three presents an experimental evaluation in demand forecasting of a new product using the pattern-matching algorithm. Section four presents the conclusions and future research.

2. Pattern Matching in Time Series Forecasting

A vast amount of literature exists on detecting meaningful patterns in data mining and other domains. Both technical analysis and programming generated methods of detecting patterns have been proposed. However, the purpose of pattern matching in those studies is to find a good match of specific patterns "inside" a time series. Examples of researches in pattern matching include pattern recognition [8] and the similarity measurement [1, 3, 6]. While the basic concept of the proposed pattern-matching algorithm for forecasting is that the meaningful patterns would repeat itself in the future of the time series. Therefore we only consider if a certain pattern is developing at the end of the time series. In other words, we treat the patterns not in the context of discovering higher-level relations of what they mean but use the patterns as possible representations of future values for the time series. The main purpose our pattern matching is to decide whether a pattern would actually represent the development of an on-going time series.

The idea of the proposed forecasting method originates from the concept that humans use available patterns to predict the future. Hence, we consider the followings steps as possible procedures for pattern matching in time series forecasting:

(1) Match patterns at the end of the time series only. The purpose is to predict future value. The way the patterns exist in the (past) times series is not important;
(2) Start by guessing and matching "partial" patterns. Only the front part of the patterns, which represents the portion of the pattern that has already occurred, is matched into the time series. The remainder or the "unmatched" portion of the pattern is used as forecasts of future values.
(3) Consider several candidates before a good match is completed. There might be several candidates of patterns at the beginning of the forecasting process. The true, or closest pattern that represents the on-going time series may not be revealed at the beginning but will be identified as the process is continued. In some cases, a combination of several patterns may be the best way to characterize the on-going time series. Hence, a weighting scheme would be necessary to make an overall forecast from different patterns. Re-evaluation is needed as the new data of the time series arrives.
(4) Adjust patterns to fit the time series. Transformations of pattern may be needed to increase the precision of the pattern matching as well as the forecasting accuracy.

Note that it is not obligatory to obtain the patterns directly from the historical data of the same time series. For example, in the case of demand forecasting of a new product or the life cycle of the product is too short to extract meaningful patterns and assuming that the company had produced similar items in the past, the demand patterns of the old products might be very good indicators for the new product. This suggests a way to forecast a short time series in such a special case (i.e., use patterns found in other time series).

An important assumption of the proposed matching algorithm is that the current time series is developing into a trend that is equal or similar to an interesting pattern and we have several possible candidates of the pattern at hand. The problem is then reduced to find which pattern or combination of patterns will be best to represent the immediate future of the time series.

Based on the above analysis, we propose the following algorithm:

Pre-condition: time series x, candidate patterns y.

Post condition: forecasted future values x¢.

1. Apply the matching algorithm to each pattern. (Matching algorithm is stated in section 2.1)
2. Decide the candidate set of patterns and the weights for each pattern (Weighting scheme is stated in section 2.2). The overall forecast is the weighted sum of forecasts from the patterns.
3. Update mean square error (MSE) for each pattern when new data arrives. If the updated MSE's indicated a better match or one or more of the candidate patterns were exhausted, repeat step (1) and iterate Step (2). Otherwise, advance the time index, report the new forecasts with the same set of patterns and their weights.

2.1 Matching Algorithm

From the discussion in the previous section, we propose the matching algorithm in this section. For us to find the best matching for the pattern, we need to define how well the patterns will fit at the end of the time series. The similarity measure used in this study is mean square error (MSE). MSE was chosen in this study because patterns may not match to the same depth for each pattern. If measures such as total sum of square were used to estimate the forecasting accuracy, the result would be biased by the length that individual pattern matched to the time series. The objective of the proposed matching algorithm is to find the best match with the minimum MSE. The pattern-matching algorithm is described as follows.

```
Given: Time series x={x(0), x(-1), x(-2)  x(-3),….}
          /* x(t) indicate the past history at time t. */
          /* t=0 indicates current time */
         Pattern: {y(0), y(1) y(2),…y(n)}
          /* length of the pattern = n*/

Output: optm, number of period used in the pattern
         that generate the best MSE; opt = the best
         MSE, and forecasts {x'(1), x'(2), x'(3), …
         x'(n-optm) }
begin
    optm= 3;       /*starting with a 3-point match to
                    avoid over-fitting*/
    opt = 10000;         /* initializing with a large
                          number*/

    for m= 3 to n-1 do
                    /* m indicates the matching depth */
       begin
          mse=10000;  /* initializing with a large
                              number*/
          for k = 1 to m do
               mse=(x(-k+1)-y(m-k))^2;
           mse=mse/m;

           if mse <= opt;
           begin
                optm=m;
```

```
            opt=mse;
        end
    end

    for j =1 to (n-optm) do
        x'(j) = y(optm+j);
              /* reporting the remainders of the */
              /* pattern as forecasts*/
     output  optm, opt, x';
end
```

An example of matching depths (m) 4 and 5 of a pattern into a time series is illustrated in Figure 1. The matching depth of 4 indicated the first 4 data points in the pattern (y_0, y_1, y_2 and y_3) which are compared with the last 4 data points in the time series (x_{-3}, x_{-2}, x_{-1} and x_0). The MSE of matching depth of 4 is calculated as the average of the squared difference of the 4 paired data points. The construction of the MSE for matching depth of 5 would follow a similar definition As shown in Figure 1, m = 4 has a smaller MSE than m = 5. If m = 4 has the smallest MSE, y_4 y_5, … will be reported as forecasts of the future values. In the next iteration, i.e. when new data arrives, the MSE will be updated. If the updated MSE is within the threshold, y_5, y_6… will be reported as forecasts. Otherwise, a "rematch" will be performed for this pattern by using the proposed matching algorithm.

obtain an overall forecast. In the literature on combining forecasts, researchers proposed various ways of assigning weights to different forecasting methods. However, most of the methods employ mathematical procedures such as quadratic programming that requires all the methods to be examined over the same period of time. In the proposed pattern matching forecasting, the matching depths of patterns may not be the same; hence the quadratic programming is not applicable. Instead, we proposed a simple weight-assigning procedure in order to decide how well the pattern fit into the time series.

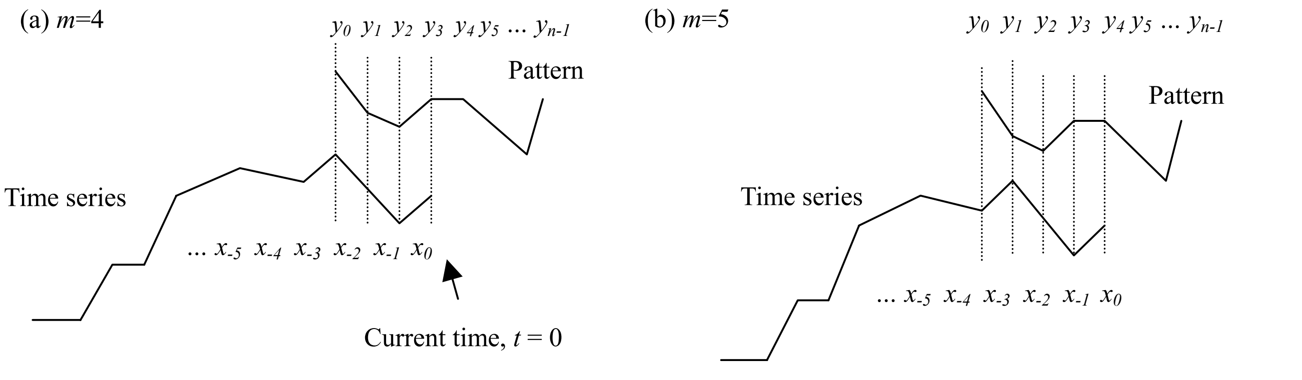
The MSE for each pattern is used to evaluate the performance of the pattern, and thus, will be used to decide the weight assigned to the pattern. Not all of the patterns are included in the overall forecast. A threshold value m* is first determined to select the candidate patterns. A candidate pattern is the one whose minimum MSE is less than m*. The weight assignment for each pattern is then proposed as follows.

$$w_i = \frac{m^* - mse_i}{\sum_{i \in C} (m^* - mse_i)} \quad \text{for all } i \in C;$$

$$w_i = 0 \qquad \text{otherwise.} \qquad (1)$$

where w_i is the weight for candidate pattern i, mse_i is the MSE for candidate pattern i and C is the index set for the candidate patterns.

For example, assume that we choose the threshold, m^*



$$mse = \frac{(x_0 - y_3)^2 + (x_{-1} - y_2)^2 + (x_{-2} - y_1)^2 + (x_{-3} - y_0)^2}{4} \qquad mse = \frac{(x_0 - y_4)^2 + (x_{-1} - y_3)^2 + (x_{-2} - y_2)^2 + (x_{-3} - y_1)^2 + (x_{-4} - y_0)^2}{5}$$

Where x_t = time series data at time t with current time t equal to 0
y_i = pattern data in a relative time index i = 0, 1, 2, ... n.
n = length of the pattern

Figure 1. Illustrated calculation of *mse* with pattern matching depth m= 4 and 5.
(Patterns were lifted for better demonstration).

2.2 Weighting Scheme

In the case of using several candidate patterns to perform the forecasting, a weighting scheme would be necessary to combine the forecasts of the patterns and

= 5, the MSE's are 6, 4, 2.5, 7, and 1 for patterns i = 1, 2, …, 5, respectively. Then patterns 2, 3 and 5 will be the candidate patterns since their MSE's are lower than 5. Their weights can be calculated as follows: w_2 =

1/(1+2.5+4)=0.1333, w_3 = 2.5/(1+2.5+4)=0.3333, w_5 = 4/(1+2.5+4)=0.5333. Weights w_1 and w_4 will both be zero because the MSE's for patterns 1 and 4 are larger than *m**. Note that pattern 5 has the largest weight for having the smallest MSE among the patterns. If the reported forecasts are 45, 36, 47, 52, and 67 from the five patterns, the overall (weighted) forecast would be 45 × 0 + 36 × 13.33% + 47 × 33.33% + 52 × 0 + 67 × 53.33% = 56.2.

The weights are adjusted dynamically to improve forecasting accuracy. Since the MSE for each pattern will likely be different when the new data arrives, the candidate set and the assigned weights need to be re-evaluated to reflect the best combination of the overall forecast. In the above example, we will still keep patterns 1 and 4 and their predictions because there might come a time when they would make a better forecast. In other words, once their MSE's become smaller than m*, they can be included in the candidate set. This dynamic re-evaluation of weights makes the proposed pattern-matching algorithm possible to facilitate potential changes in characteristics of the time series.

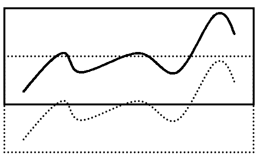
2.3 Transformation of Patterns

Transformation of a pattern are needed to lower the MSE for the reason that the time series may be developing similarly to but not exactly the same as the pattern. We would like the pattern-matching algorithm to be flexible and able to deal with scaling and time axis distortion problems. Techniques such as dynamic time wrapping (DTW) have been used in regular pattern matching inside a time series [1, 6]. However, they are not applicable to the forecasting algorithm proposed in this study. The reason is that only a partial pattern is used in our matching algorithm, while DTW requires two full patterns. Hence, the following pattern transformations are proposed.

(1) Vertical Shift --

$$y'_i = y_i + s, \qquad (2)$$

where s is the displacement in the vertical direction. 's' could be either positive or negative to represent a shift up and/or down respectively. Horizontal shift is not required because of the way pattern fits into the time series covers the horizontal shift.

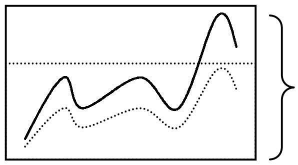


}s

(2) Vertical extension/compression --

$$y'_i = \min_i y_i + v\left(\frac{y_i - \min_i y_i}{\max_i y_i - \min_i y_i}\right), \qquad (3)$$

where v is the ratio the whole pattern extends vertically. (See appendix A for detailed derivation) If v<1, this indicates a compression whereas if v>1, indicates an extension.



v

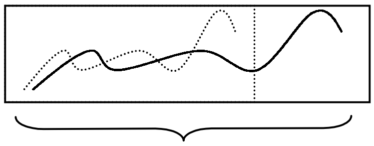
(3) Horizontal extension/compression – the way to calculate a horizontal extension is more complex than the calculation for a vertical one. First, we need to reconstruct a new time index i' = 1, 2, 3, … [nh] where h is the ratio of the whole pattern extending horizontally, n is the length of the pattern, and [nh] is the largest integer less than or equal to nh.

Then $y'_{i'}$ = Interpolation of y_i and y_{i+1}

where $h\, i <= i' < h(i+1)$

$$= y_i + (i' - hi)/h \times (y_{i+1} - y_i) \qquad (4)$$

(See appendix B for detailed derivation)



h

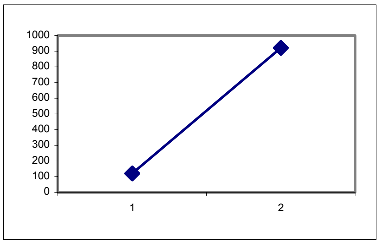
(4) Combination of the above transformations

Transformation(s) should be applied to the pattern prior to the matching algorithm. The optimal transformations (*s, v* and/or *h*) may be decided by using mathematical programming. In demand forecasting, the transformation in horizontal direction may not be much realistic because of the seasonality issue of product sale. Therefore, the combination of shift and vertical extension/compression will be the two main transformations in the application on of demand forecasting. A heuristic procedure to find s and v is (1) shifting the pattern so that the mean value of the selected data points in pattern is equal to the mean value of the matched data points in time series; then (2) finding v that generates the minimum MSE.

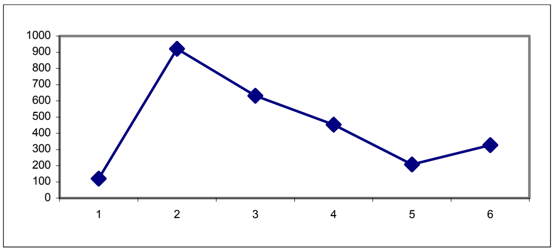
3. Experimental Evaluation

As mentioned in the previous section, the transformation of today's business focus from mass production to semi-customization shortens a product's life cycle and makes demand forecasting difficult. This is especially true when a new product enters the market, because there is no historical data available to make a forecast. However, the proposed pattern matching procedure still generates reasonable predictions since patterns were extracted from the past sales of similar products and/or predictions from human experts.

To demonstrate the usefulness of the proposed pattern matching procedure, we use a real world data set. All data are obtained from the same company where the demand forecasting of a new product X is desired. In particular, the sales data of the first two months of product X is collected (Figure2 a). The past sales data of four similar products (products O, Q, R and U) are used as patterns (Figure3) to predict the future demand of product X.



(a)



(b)

Figure 2. The actual demand of product X during (a) first two months (b) the first 6 months.

As shown in Figure 3, most of the sale patterns consist of an immediate surge of sales shortly after the products were introduced to the market due to their promotion, then the demand subsides soon after the new product's introduction. Those patterns look similar, but have their own characteristics. What differentiates the patterns are the customers' reaction to the promotion of the new products and how the products were accepted by the customers. The purpose for this case study is to examine if the development of product X is similar to those patterns, in fact, we use the combination of those patterns to predict the future demand of product X.

In our example, month 2 will be the current time, which is noted as t_0. We attempt to predict month 3 (t_1), 4 (t_2), 5(t_3)… etc. Forecasted and actual values will be denoted as x_i' and x_i, respectively, with an appropriate time index.

Notice that we only have two data points of product X (120 and 921) at the beginning of the experiment, almost all the existing forecasting methods is not applicable to such a short time series. One can only "guess" that the value for the third month will either follow the linear trend from the first two months, i.e. $x'_1 = 921+(921-120)=1722$, or average the sales of the first two months i.e. $x'_1 = (120+921)/2=520.5$. Apparently, neither would be a reasonable forecast because judging from the sale patterns of products O, Q, R and U, there might be a turning point in month 2 (t_0) or 3 (t_1). But, we will apply the matching algorithm and let the procedure decide what the turning point should be, by finding the best matches using the matching algorithm.

The result of the matching is shown in Table 1. As

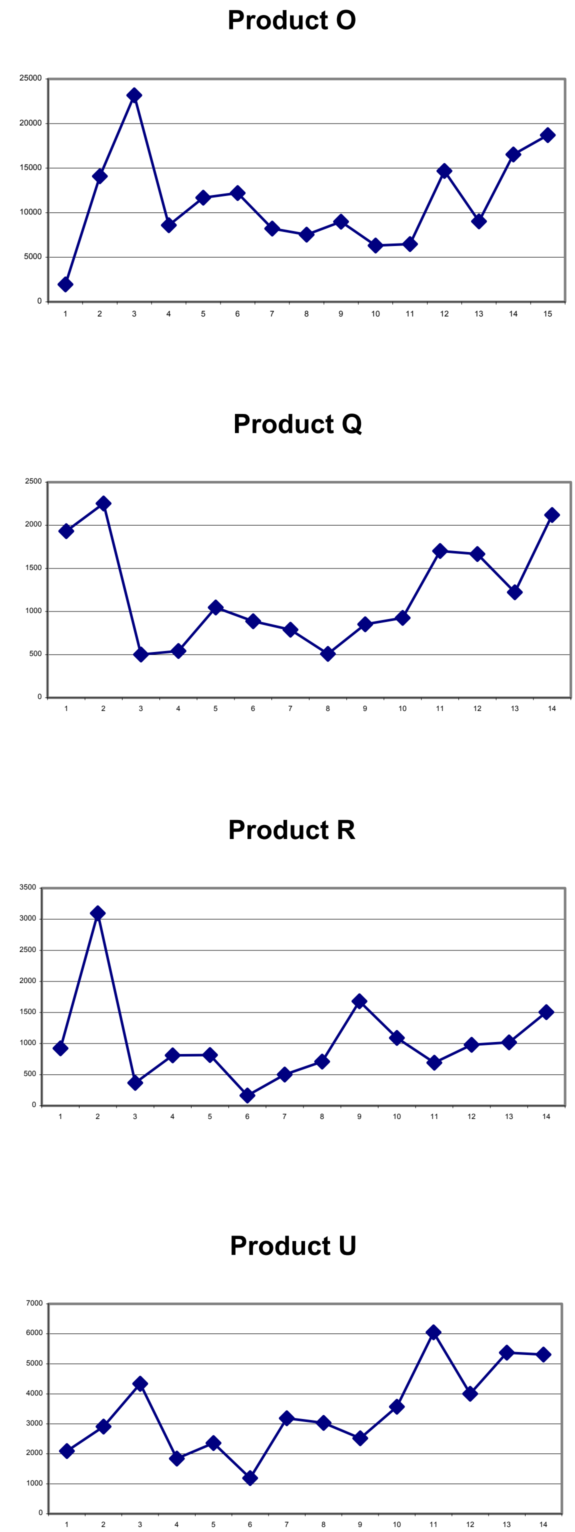


Figure 3. Sale patterns of products O, Q, R and U that were used to predict the future demand of product X

shown in Table 1, the weights were assigned equally at the first month, because there were no historical performance evaluations for the patterns. Starting at month 4 and continuing to month 8, the weights for patterns of products O and R are heavier than the other two products. If we examine the patterns and the overall development of product X (Figure 2(b)), we can easily tell that product X's on-going sale pattern is more comparable to products O and R than those of products Q and U. In other words, the patterns of two products that show more closer resemblances to product X receive heavier weights.

Table 1. Forecasts with patterns of past sales products, corresponding weights and over all forecast for product X

	Month					
	3	4	5	6	7	8
Forecast						
product O	1320.4	339.69	395.49	374.80	245.23	222.23
product Q	0	451.41	428.61	344.43	313.20	277.36
product R	0	260.63	319.63	161.00	209.35	222.30
product U	2121.4	677.57	366.71	107.26	450.10	382.78
Weights						
product O	25%	59.94%	33.04%	29.70%	28.11%	28.95%
product Q	25%	4.24%	14.05%	14.90%	17.23%	18.56%
product R	25%	21.46%	29.94%	31.11%	29.95%	30.97%
product U	25%	14.36%	22.97%	24.28%	24.72%	21.52%
Overall Forecast	860.44	375.97	370.82	238.79	296.83	267.03

The comparison of predictions and the actual demand for up to month 8 are listed in Table 2. In Table 2, we also compare our result with the predictions made by the exponential smoothing method. The exponential smoothing method was selected because all the other methods, such as ARIMA and neural networks, would require much more data to create or train a model. Exponential smoothing and simple moving averaging are the only two choices available for such a short past history. Exponential smoothing is preferred over the simple moving averaging method in terms of better forecasting accuracy. The values shown for months 3-7 for the exponential methods are fitted value; the only forecast value is the one for month 8. The RMSE (root mean squared errors) of the forecast for both methods are listed. The graphic representation of how the two systems performed in predicting product X is depicted in Figure 4. Apparently, the proposed pattern matching procedure has outperformed exponential smoothing, which is the only traditional forecasting method available to predict a time series with such a short history.

Table 2. Comparison of forecasts with pattern matching and exponential smoothing.

	Month					
	3	4	5	6	7	8
Actual	632	453	208	327	220	374
Forecast: Pattern Matching	860.4	376.0	370.8	238.8	296.8	267.0
Exponential Smoothing	395.5	425.1	428.6	401.0	391.8	370.3
Note	RMSE for Pattern Matching is 140 RMSE for Exponential smoothing is 281					

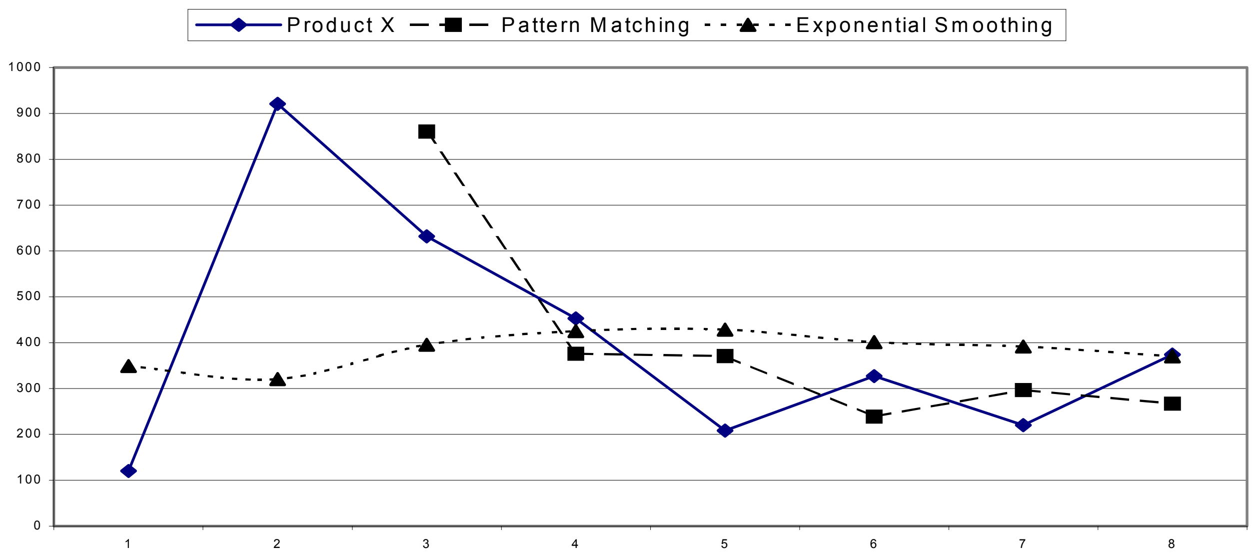


Figure 4. Comparison of pattern matching and exponential smoothing.

4. Conclusion

This paper has presented a novel approach to demand forecasting that is designed for pattern matching. Patterns could be defined by human experts or extracted from various data mining techniques. In our example, instead of creating models from the immediate past history to predict future values, the past sales of similar products were used as predictors of the sale of the product examined. Initial evaluation indicates superior performance in predicting demand of a new product. With little demand data available for the new product, most existing methods are not even applicable to create valid models for demand forecasting.

Although it is not the focus of this study, the presence of "quality" patterns to forecast is essential for this approach. Nevertheless, the approach provides an alternative way of forecasting and beefs up some weakness of the traditional methods. It may be used in conjunction with other forecasting methods to improve overall forecasting results and to apply the pattern matching procedure to a wide range of forecasting problems. In fact, the proposed dynamic pattern-matching algorithm is part of the architecture of an agent-based forecasting system [10]. The purpose is to capture the uncertainty that exists in temporal data sets and apply advanced data mining technique to improve forecasting accuracy.

Appendix - Derivation of Transformation Equations

A. Vertical extension/compression

let y_i denote the original pattern at time i, y_i' denote the transformed pattern at time i

let v be the ratio of the whole pattern extends vertically from the bottom of the pattern

$\Rightarrow \min\limits_i y_i$ remain the same, i.e. $\min\limits_i y_i' = \min\limits_i y_i$,

while $\max\limits_i y_i$ extend up to a ratio of v i.e. $\max\limits_i y_i' = \min\limits_i y_i' + (\max\limits_i y_i - \min\limits_i y_i) \times v$

$\Rightarrow$ for any y_i, y_i extend up to a ratio of $(y_i - \min\limits_i y_i) / (\max\limits_i y_i - \min\limits_i y_i) \times v$

i.e $y_i' - \min\limits_i y_i' = (y_i - \min\limits_i y_i) / (\max\limits_i y_i - \min\limits_i y_i) \times v$

$\Rightarrow y_i' = \min\limits_i y_i + (y_i - \min\limits_i y_i) / (\max\limits_i y_i - \min\limits_i y_i) \times v$

B. Horizontal extension/compression

let y_i denote the original pattern at time i,
y_i' denote the transformed pattern at time i
n denote the length of the pattern.

let h be the ratio of the whole pattern extends horizontally from the beginning of the pattern

$\Rightarrow y_i$ will map to the new domain set of $\{h, 2h, 3h, ...nh\}$
Since the element of the new domain set may not necessary be integer, which is not eligible to be a time index set, a new time index set I' will be defined as $\{1, 2, 3, ...[nh]\}$. Where $[nh]$ is the largest integer that less than or equal to nh.

$\Rightarrow y_i'$ the will become the new range set that the time index set I' maps into.

$\Rightarrow y'_{i'}$ = Interpolation of y_i and y_{i+1}
where $h\,i <= i' < h(\,i+1)$

$\Rightarrow (y'_{i'} - y_i) / (y_{i+1} - y_i) = (i' - ih) / ((i+1)h - ih)$

$\Rightarrow y'_{i'} = y_i + (y_{i+1} - y_i) \times (i' - ih)/h$

References

[1] Berndt D. J. and J. Clifford, "Finding Patterns in Time Series: A Dynamic Programming Approach," in U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, Menlo Park, California, pp229-248, 1996

[2] Chatfield C., "Time-Series Forecasting," Boca Raton: Chapman and Hall, 2001

[3] Gunopulous, D. and G. Das, "Time Series Similarity Measures," Tutorial Notes for ACM SIGKDD 2000, pp 245-307, 2000

[4] Hill, T., M. O'Connor and W. Remus, "Neural Network Models for Time Series Forecasts" Management Science, 42:7, pp 1082-92, 1996

[5] Kao, R. J., "A Sale Forecasting System Based on Fuzzy Neural Network with Initial Weights Generated by Genetic Algorithm," European Journal of Operational Research, 129, pp 496-517, 2001

[6] Keogh, E. and Pazzani, M., "Scaling up Dynamic Time Warping for Datamining Applications," Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 285-89, 2000

[7] Lee, K. C. and S. B. Oh, "An Intelligent Approach to Time Series Identification by a Neural Network-driven Decision Tree Classifier," Decision Support System, 17 pp 183-97, 1996

[8] Morrill, J. P., "Distributed Recognitions of Patterns in Time Series Data" Communications of the ACM, 41:5, pp 45-51, 1998

[9] Antunes, C. M. and A. L. Oliveria, "Temporal Data Mining: an overview," Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop on Temporal Data Mining, August 26, 2001

[10] Yu, W. and J. H. Graham, "A MultipleAgent Architecture for Demand Forecasting in Electronic Commerce Supply Chain System," Proceedings of the 17th International Conference on Computers and Their Applications (CATA-2002), pp. 462-66, April 2002