Pattern Matching-based Prediction using Affine Combination of Two Measures: Two are Better Than One

Abstract Time series forecasting based on pattern matching has received a lot of interest in recent years due to its simplicity and the ability to predict complex nonlinear behaviors. The popular method for this approach is *k*-nearest neighbor (*k*-NN). However, The choice of the metric to measure the similarity between two time series depends mainly on the specific features of the considered data and it can influence on forecasting results. So, adoption of only one measure can achieve good result in some cases, but opposite result in some others. In this paper, Unlike the conventional method, we propose an improved pattern matching-based prediction method using affine combination of two measures, Euclidean distance and dynamic time warping (DTW), in order to achieve a better forecasting result. These two distance measures are chosen because they are two the most commonly used metrics for pattern matching in time series. The experimental results showed that our approach can produce better results on time series forecasting work in comparison to the pattern matching-based method under Euclidean distance or DTW in terms of prediction accuracy.

Keywords: time series; pattern matching; time series prediction; dynamic time warping; *k*-nearest neighbor.

1 Introduction

A time series is a sequence of real numbers in which each number represents a value at a given point in time. Time series data arise in so many applications of various areas ranging from science, engineering, business, finance, economic, medicine to government.

An important research area in time series data mining which has received an increasing amount of attention lately is the problem of prediction in time series. A time series prediction system predicts future values of time series variables by looking at the collected variables in the past. The accuracy of time series prediction is fundamental to many decision processes and hence the research for improving the effectiveness of prediction methods has never stopped.

One thing the pattern matching-based forecasting has in common is that it needs to find the best match to a pattern from a pool of time series in the past. The choice of the metric to measure the similarity between two time series depends mainly on the specific features of the considered data and it can influence on forecasting results. So, adoption of only one measure can achieve good result in some cases, but opposite result in some others. However, it is difficult for users to recognize which measure is appropriate to which dataset. To handle this problem, in our work, we propose an improved pattern matching-based prediction method using affine combination of two measures. We investigate this approach in order to combine the benefits of these distance measures in time series prediction. The pattern matching method here is the *k*-nearest neighbor method. The *k*-nearest neighbor algorithm is selected since it is simple and it can work very fast. In this study, we investigate the affine combination of two measures, Euclidean distance and dynamic time warping (DTW). They are chosen because Euclidean distance is the most commonly used metric for pattern matching in time series (Keogh and Kasetty, 2002), while DTW is introduced as a solution to the weakness of Euclidean distance metric (Berndt and Clifford, 1994). So Euclidean distance and DTW can be complementary in time series prediction.

Pattern matching-based forecasting technique using affine combination of two measures performs as follows: first, it retrieves the pattern (subsequence) prior to the interval to be forecasted. Then this pattern is used for searching *k* nearest neighbors in history data. The search is carried out in two modules simultaneously. Each module uses one distance measure. Next, in each module, subsequences next to these found *k* nearest neighbors are retrieved. Then an estimated result in each module is

calculated by averaging the subsequences found in the immediate previous step. Finally, the estimated results in two modules are fed into the synergetic module to generate a final forecast result.

In our experiment, the proposed forecasting approach is compared to the pattern matching-based method under Euclidean distance or DTW in terms of prediction accuracy.

We experiment our proposed method on a variety of real world datasets. The experimental results show that our proposed method can produce better results on time series prediction in comparison to the conventional method under Euclidean distance or DTW.

The rest of the paper is organized as follows. In section 2 we examine related words and background. Section 3 describes our approach for forecasting in time series. Section 4 presents our experimental evaluation on real datasets. In section 5 we include some conclusions.

2 Background and related works

2.2 Related works

Various kinds of prediction methods have been developed by many researchers and business practitioners. Some of the popular methods for time series prediction such as exponential smoothing (Gelper et al., 2010), ARIMA model ((Chatfield, 2000), (Kang, 2003), (Kim, 2003)), artificial neural networks - ANNs ((Balkin and Ord, 2000), (Cadenas and Rivera, 2009), (Ghiassi et al., 2005), (Heravi et al., 2004), (Tkacz, 2001), (Zhang and Qi, 2005)) and Support Vector Machines - SVMs ((Kim, 2003), (Radhika and Shashi, 2009)) are successful in some given experimental circumstances. For example, the exponential smoothing method and ARIMA model are linear models and thus they can only capture the linear features of time series. ANN has shown its nonlinear modeling capability in time series forecasting. However, this model is not able to capture seasonal or trend variations effectively with the un-preprocessed raw data (Zhang and Qi, 2005). Besides several prediction method based on pattern matching have been introduced. Below are some typical methods.

Alicia et al. (2004) proposed a new method based on clustering technique. It is called Pattern Sequence-based Forecasting (PSF). This method uses clustering technique to generate labels and makes predictions basing only on these labels. This approach gives a new methodology to automate the obtaining of the labels which supply rules for forecasting. The approach was applied to forecast electricity market prices and electricity demand by Álvarez et al. (2011).

Sorjamaa et al. (2005) introduced a method which combines mutual information and k-nearest neighbor approximator for time series prediction. Mutual information is used as a criterion to select the best input variables from a set of possible variables and k- nearest neighbor approximator is used to improve the input selection and to provide a simple but accurate prediction method.

Nayak et al. (2007) examined the predictive power of the clustering technique on stock market data using a brute force algorithm. Their proposed method is based on the idea that a cluster formed around an event could be used as a good prediction for the future event. That cluster can be determined by using a minimum distance. Nayak et al. applied this temporal pattern matching to predict the values of Australian stock prices.

Jiang et al. (2009) proposed a new method for predicting stock time series data. This method is based on motif information. First, it discovers the most important motif in time series. After that motif is divided into two parts: prefix and postfix. If a current pattern of time series and the prefix of motif are matched, the postfix of motif can be used to predict the next time step of time series.

Arroyo and Mate (2009) proposed a time series forecasting mothod which adapts k-nearest neighbor method to forecasting histogram time series (HTS). This HTS is used to describe situations where a distribution of values is available for each instant of time. The authors showed that this method can yield promising results.

Huang et al. (2012) proposed a new method for long-term time series prediction. This method is based on multi-model integration. The multi-model integration strategy includes two step: First, a k-nearest neighbor based on least squares support vector machine (LS-SVM) approach is used for long-term time series prediction and then an autoregressive model is employed to combine the predicted values from the multiple k-NN based LS-SVM models to generate a final predicted value as the output.

Zhang et al. (2013) presented a *k*-nearest neighbor model for short-term traffic flow prediction. First, this method preprocesses the original data and then standardizes the processed data in order to avoid the magnitude difference of the sample data and improve the prediction accuracy. At last, a short-term traffic prediction based on *k*-NN nonparametric regression model is carried out.

Cai et al. (2015) proposed an improvement on the *k*-NN model for road speed forecast based on spatiotemporal correlation. This model defines the current conditions by the two-dimensional spatiotemporal state matrices, instead of the one-dimensional state vector of the time series and determines the weights by Gaussian function to adjust the matching distance of the nearest neighbors.

2.1 Background

- **Euclidean Distance**

Euclidean distance is the simplest method to measure the similarity of time series. Given two time series $Q = \{q_1,\ldots, q_n\}$ and $C = \{c_1, \ldots, c_n\}$, the Euclidean distance between Q and C is defined as:

$$D(Q,C) = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \tag{1}$$

To reduce the complexity of Euclidean distance calculation, we can use the idea of early abandoning introduced in (Mueen et al., 2009). The idea of early abandoning is performed as follows: when the Euclidean distance is calculated for a pair of time series, if the cumulative sum is greater than the current best-so-far distance at a certain point we can abandon the calculation because this pair of time series is not the best match. Figure 1 shows the intuition behind this technique. In this example the current best-so-far distance is supposed of 11. At the point the squared Euclidean distance of 121 we can stop this calculation.

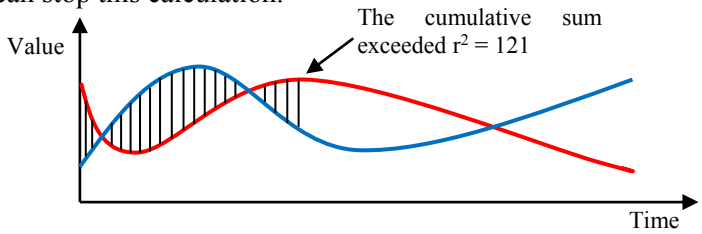

Fig. 1 An illustration of the idea of early abandoning technique

Euclidean distance is the most commonly used metric for pattern matching in time series (Keogh and Kasetty, 2002). However, its weakness is that it is sensitive to distortion in time axis. The DTW distance measure is introduced as a solution to the weakness of Euclidean distance metric (Berndt and Clifford, 1994). For example, in the case of the pattern and a candidate time series have an overall similar shape but they are not aligned in the time axis, the Euclidean distance will produce a pessimistic dissimilarity measure. In this case the DTW distance can produce a more intuitive distance measure. Figure 2 illustrates this case.

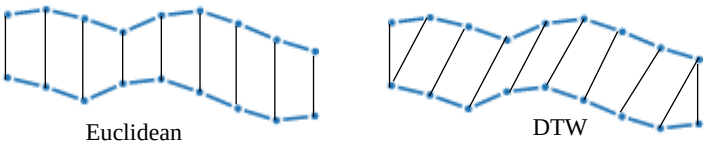

Fig. 2 An example illustrates the Euclidean distance and the DTW distance.

- **Dynamic time warping distance**

In 1994, the DTW technique is introduced to the database community by (Berndt and Clifford, 1994). This technique allows similar shapes to match even if they are out of phase in the time axis. So, it is widely used in various fields such as bioinformatics, chemical engineering, robotics, and so on. However, it does not conform to the triangular inequality.

Given two time series Q of length n, $Q = \{q_1, \ldots, q_n\}$, and C of length m, $C = \{c_1, \ldots, c_m\}$, the DTW distance between Q and C is calculated as follows.

First, an n-by-m matrix is constructed where the value of the (i^{th}, j^{th}) element of the matrix is the squared distance $d(q_i, c_j) = (q_i - c_j)^2$. To find the best distance between two sequences Q and C, a path through the matrix that minimizes the total cumulative distance between them is retrieved. A warping path, $W = w_1, w_2, \ldots, w_L$ with $\max(m, n) \le L \le m+n-1$, is an adjacent set of matrix elements that defines a mapping between Q and C. The optimal warping path is the path which has the minimum total cumulative distance. It is defined as:

$$DTW(Q,C) = \min_{W}\left\{\sum_{k=1}^{L} d_k, W = \langle w_1, w_2, \ldots, w_L \rangle\right\} \quad (2)$$

where $d_k = d(q_i, c_j)$ indicates the distance represented as $w_k = (i, j)_k$ on the path W.

To find the warping path, we can use dynamic programming which is calculated by using the following formula.

$$v(i, j) = d(q_i, c_j) + \min\{v(i-1, j-1), v(i-1, j), v(i.j-1)\} \quad (3)$$

where $d(q_i, c_j)$ is the distance found in the current cell, $V(i, j)$ is the cumulative distance of the current cell and the minimum of cumulative distances from the three adjacent cells.

Figure 3 shows an example of how to calculate the DTW distance between two time series Q and C.

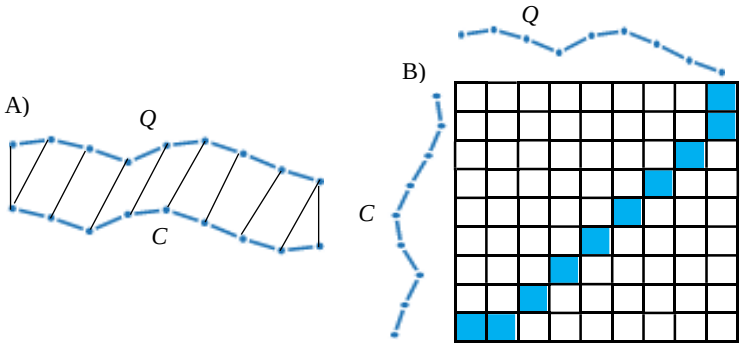

Fig. 3 An example of how to calculate the DTW distance between Q and C. (A) Two similar but out of phase time series Q and C. (B) To align two time series, a warping matrix is constructed for searching the optimal warping path.

A recent improvement of DTW that considerably speeds up the DTW calculation is a lower bounding technique based on the warping window (Keogh, Ratanamahatana, 2005). Figure 4 illustrates the Sakoe-Chiba Band (Sakoe and chiba, 1978) and the Itakura Parallelogram (Itakura, 1975) which are two most common constraints in the literature.

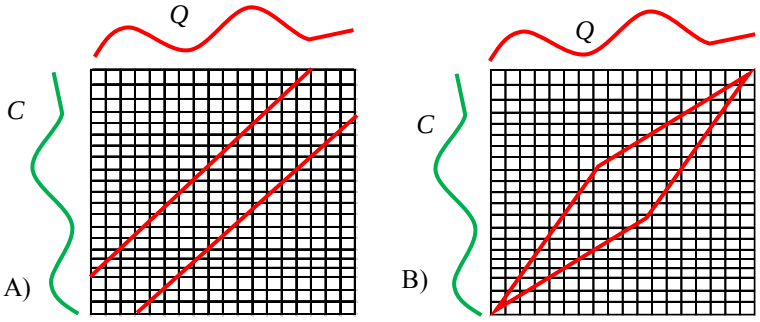

Fig. 4 An example illustrates (A) Sakoe-Chiba Band and (B) Itakura Parallelogram.

According to this technique, sequences must have the same length. If the sequences are of different lengths, one of them must be re-interpolated. In order to enhance the search performance in large databases, first a warping window is used to create an above bounding line and a below bounding line (called bounding envelope) of the query sequence. The bounding envelope of a time series Q which is formed by the upper bound (U) and lower bound (L) sequences is defined as follows (Keogh, Ratanamahatana, 2005).

$$U_i = \max(q_{i-r} : q_{i+r}) \quad (4)$$

$$L_i = \min(q_{i-r} : q_{i+r}) \quad (5)$$

where r is an allowed range of warping for a given point in a sequence. In the case of the Sakoe-Chiba band, r is independent of i. For the Itakura parallelogram, r is a function of i.

Then the lower bound distance, LB_DTW(Q, C) is calculated as the squared sum of the distances from every part of the candidate sequence not falling within the bounding envelope, to the nearest orthogonal edge of the bounding envelope.

$$LB\_DTW(Q,C) = \sqrt{\sum_{i=1}^{n} \begin{cases} (c_i - U_i)^2 & if\ c_i > U_i \\ (c_i - L_i)^2 & if\ c_i < L_i \\ 0 & otherwise \end{cases}} \tag{6}$$

For any two sequences Q and C of the same length n, for any global constraint on the warping path of the form $j - r \le i \le j + r$, the lower bound distance conforms to the lower bound condition, ie. LB_DTW(Q, C) ≤ DTW(Q, C) (Keogh, Ratanamahatana, 2005).

Figure 5 illustrates this technique in which Sakoe-Chiba Band is used to create a bounding envelope. We can do the same in the case of Itakura Parallelogram.

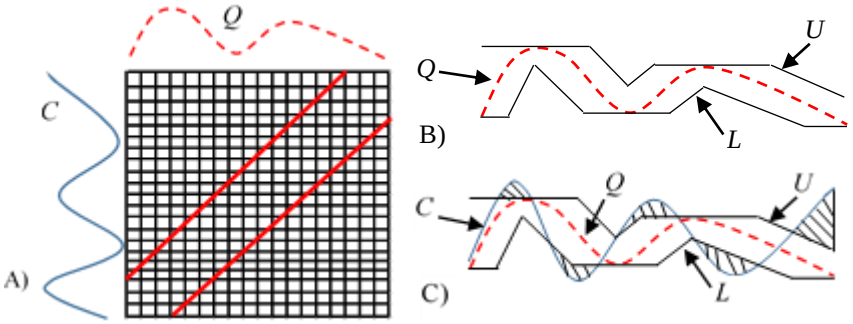

Fig. 5 (A) The Sakoe-Chiba Band is used to create a bounding envelope. (B) The bounding envelope of Q using the Sakoe-Chiba band. (C) The lower bound for DTW distance retrieved by calculating the Euclidean distance between any candidate sequence C and the closest external part of the envelope around a query sequence Q.

- **Similarity search in time series**

In similarity search, a lower bounding distance measure can help prune sequences that could not be the best match. Figure 6 illustrates an algorithm using a lower bounding distance measure to speed up the sequential scan search. In Figure 6 *lb_dist* is the lower bound distance between a candidate sequence C_i and a query Q under DTW or Euclidean distance, *true_dist* is the true distance between a candidate sequence C_i and a query Q under DTW or Euclidean distance.

Algorithm: Similarity search which uses a lower bounding distance
Input: Time series database D, a query Q
Output: Index of best match *ind*, the best so far distance *bsf_distance*

```
Create a bounding envelope (U, L) for Q
bsf_distance = infinity
ind = -1
for i = 1 to |D|
    lb_dist = lower_bound_distance(Q, Ci)  //Ci ∈ D
    if lb_dist < bsf_distance
        true_dist = true_distance(Q, Ci)
        if true_dist < bsf_distance
            bsf_distance = true_dist
            ind = i
        end if
    end if
end for
return (ind, bsf_distance)
```

Fig. 6 An algorithm using a lower bounding distance measure

Besides, a multidimensional index structure (e.g., R-tree or R*-tree) can be used to enhance the search performance in large databases. In this case, a multidimensional index structure such as R*-tree (Beckman et al., 1990) can be used for retrieving nearest neighbors of a query.

To insert a sequence into R*-tree, first we create a Minimum Bounding Rectangle (MBR) R in the N dimensional space ($N < n$) for each sequence of length n in the database. Then each sequence is inserted into R*-tree based on its MBR. To find a nearest neighbor of a query Q by searching in R*-tree, we need a distance function $D_{region}(Q, R)$ of the sequence Q from the MBR R associated with a node in the index structure such that $D_{region}(Q, R) \leq$ true_distance(Q, C), $\forall C$, any sequence which is contained in the MBR R. $D_{region}(Q, R)$ is defined as follows (Son and Anh, 2016).

$$D_{region}(Q,R) = \sqrt{\sum_{j=1}^{N} D_{region_j}(Q_j, R_j)} \tag{7}$$

Where

$$D_{region_j}(Q_j, R_j) = \sum_{i=1}^{m} d(q_{ji}, R_j)$$

$$d(q_{ji}, R_j) = \begin{cases} (L_j - q_{ji})^2 & \text{if } q_{ji} < L_j \\ (q_{ji} - H_j)^2 & \text{if } q_{ji} > H_j \\ 0 & \text{otherwise} \end{cases}$$

m is the length of segment j

Figure 7 illustrates a k-NN algorithm for similarity search problem using a multidimensional index structure which is similar to an algorithm introduced in (Keogh, Ratanamahatana, 2005). In this algorithm, a priority queue is used to contain visited nodes in the index in the increasing order of their distances from query Q. The distance defined by $D_{region}(Q, R)$ is used to search in R*-tree. If the current item is a data item, the true distance, true_distance(Q, C) which is defined by $D(Q, C)$ or DTW(Q, C), is used. A sequence C is moved from *item_list* to *kNN_result* if it is one of the k nearest neighbors.

Algorithm: Finding k nearest neighbors using R*-tree
Input: Time series database D, a query Q and k, the number of nearest neighbors
Output: k nearest neighbors

```
distance = 0
Push root node of index and distance into queue
while queue is not empty
    curr_item = Pop the top item of queue
    if curr_item is a non-leaf node
        for each child node U in curr_item
            distance = D_region(Q, R)
            Push U and distance into queue
        end for
    else if curr_item is a leaf node
        for each data item C in curr_item
            distance = D_region(Q, R)
            Push C and distance into queue
        end for
    else
        Retrieve original sequence of C from database
        distance = true_distance(Q, C)
        Insert C and distance into item_list
    end if
    for each sequence C in item_list which conforms to
                    the condition D(Q,C) ≤ curr_item.Distance
        remove C from item_list
        Add C to kNN_result
        If | kNN_result| = k   return kNN_result
    end for
end while
```

Fig. 7 The k-nearest neighbor algorithm using R*-tree

To avoid a degradation of multidimensional index structures when using with high dimension datasets, a dimensionality reduction method can be used to reduce the dimensionality of sequences. A dimensionality reduction method, which can be used under Euclidean distance or DTW, is the piecewise aggregate approximation (PAA) method. This method is described in (Keogh, Ratanamahatana, 2005) and (Keogh et al., 2000).

In the case of subsequences of a longer time series, the trivial matches are rejected in similarity search algorithm by using the relative positions of the subsequences. It means that a pair of subsequences is a candidate non-trivial match if there is a gap of at least w (w is specified by users) positions between these two subsequences.

3. Our Approach.

Our proposed approach for time series prediction is an attempt to combine Euclidean distance with DTW measure because they can be complementary in time series pattern matching. Through this integration a synergetic effect will believed to be created to improve the predication performance.

Pattern matching in the context of time series forecasting refers to the process of matching current pattern of the time series with its past states. This approach hinges on predicting samples in a time series by finding its k nearest neighbors under the Euclidean distance and the DTW measure simultaneously and then calculating the average of the subsequences of a given length subsequent to these k nearest neighbors as the estimated results. Finally, the estimated results are used to generate a final forecast result.

With this k-nearest neighbor approach, the length of prediction can be as long as required because this can be implemented with a loop in which forecasting samples can be inserted back into the dataset in order to predict further samples. That means k-nearest neighbor method can be applied in *multiple-step-ahead prediction*. Figure 8 shows our proposed model.

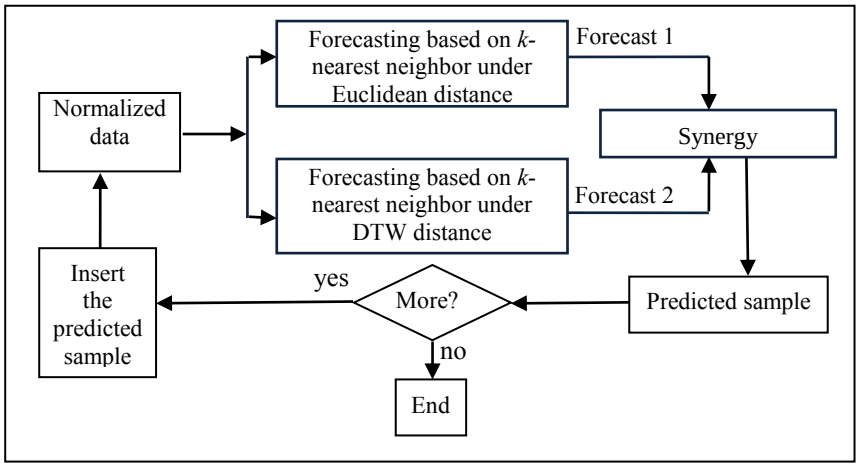

Fig. 8 The pattern matching-based predicting model using affine combination of two measures

Our approach for forecasting is described as follows:

Given the current state (pattern) of length w of time series that we have to predict a sequence of the next time step. First, the input is fed simultaneously into the k-NN under Euclidean distance-based forecasting module and the k-NN under DTW-based forecasting module. Each module searches for k nearest neighbors of the pattern. Next, in each module, subsequences next to these found k nearest neighbors are retrieved. Then the estimated sequence of each module is calculated by averaging the subsequences found in the immediate previous step. Finally, the estimated results in two modules are fed into the synergetic module to generate a final forecast result. In the case of forecasting more steps, the forecasted sequence can be inserted at the end of the data in order to predict the following pattern.

In the synergetic stage, we use affine combination of two forecast results, i.e.

$$Y_{\text{synergy}} = \omega Y_{\text{Euclid}} + (1-\omega) Y_{\text{DTW}} \tag{8}$$

where Y_{Euclid} is the forecast result obtained from k-NN under Euclidean distance-based forecasting module, Y_{DTW} is the forecast result obtained from k-NN under DTW-based forecasting module and ω is the weight parameter ($0 \le \omega \le 1$). If ω is equal to 0 or close to 0, that means the forecast generated by k-NN under DTW-based forecasting module is dominant. If ω is equal to 1 or close to 1, that means the forecast generated by k-NN under Euclidean distance-based forecasting module is dominant.

To determine the weight parameter ω we have to find the value of ω that minimizes the forecast error MSE:

$$MSE = \sum_{i=1}^{n} (Y_i - Y_{synergy,i})^2 = \sum_{i=1}^{n} (Y_i - [\omega Y_{Euclid,i} + (1-\omega) Y_{DTW,i}])^2 \tag{9}$$

where Y_i is the actual value at time point i, $Y_{Euclid,i}$ is the forecast value at time point i generated by k-NN under Euclidean distance-based forecasting module, and $Y_{DTW,i}$ is the forecast value at time point i generated by k-NN under DTW-based forecasting module. This is a quadratic function, therefor we can derive the value of ω which makes the forecast error MSE get minimum as follows:

$$\omega = \frac{\sum_{i=1}^{n} (Y_{Euclid,i} - Y_{DTW,i})(Y_i - Y_{DTW,i})}{\sum_{i=1}^{n} (Y_{Euclid,i} - Y_{DTW,i})^2} \tag{10}$$

Since ω is in the range [0,1], if the computed value of ω is negative, we can select its value as 0, and if the computed value of ω is greater than 1, we can select its value as 1.

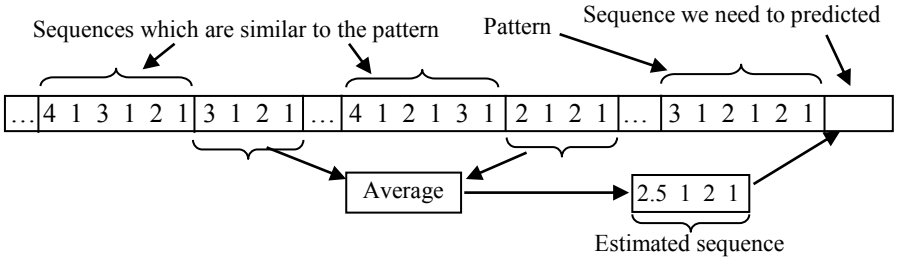

Fig. 9 Illustration of our approach for k-nearest neighbor-based forecasting.

Figure 9 illustrates the idea of our algorithm for prediction in each module introduced in (Son et al., 2013) and figure 10 shows the steps of our proposed multi-measure-based forecasting algorithm. There are three parameters that should be determined in this algorithm: k, the number of neighbors to be found; w, the length of subsequence to be matched; and m, the length of predicted sequence. The value of k has impact on the quality of forecasting. In practice, the small values of k are suitable for time series without noise. As for the pattern length (the length of window) w, the value of w is dependent on the seasonal length of the time series, for example, if the seasonal length is s, we should set the value of w to s. The value of m is dependent on the number of steps to be predicted. If the problem is one-step ahead prediction, the value of m should be set to 1.

Input: Time series D of length n_1, the current pettern of length w, the number of nearest neighbors k and the length of predicted sequence m ($m \le w << n_1$).
Output: Forecasted sequence of length m.

1. Reduce the dimensionality of subsequences of length w of D (if necessary).
2. Insert them into a multidimensional index structure (if necessary).
3. Retrieve the subsequence S of length w prior to the subsequence we have to predict in D.
4. Search for k nearest neighbors of S in two modules simultaneously (one under Euclidean distance, another under DTW measure).
5. For each k nearest neighbors found in step 3 in each module, retrieve subsequence of length m next to it in D.
6. The estimated sequence in each module is calculated by averaging subsequences found in step 5.
7. Estimated results in two modules in step 6 are fed into synergetic module to generate a final forecasted sequence.

8. Insert the final forecasted sequence in step 7 into D in order to predict the following pattern and return to step 1 (if necessary).

Fig. 10 Illustrate the algorithm of our proposed model.

In Figure 10, the first and second steps are optional. To avoid a degradation of multidimensional index structures when using with high dimension datasets, a dimensionality reduction method can be used to reduce the dimensionality of subsequences (step 1).

To enhance the search performance in large databases, a multidimensional index structure can be used for searching k nearest neighbors of a query (step 2). In a multidimensional index structure (e.g., R-tree or R*-tree), each non-leaf node is associated with a minimum bounding rectangle (MBR). A MBR at a non-leaf node is the minimum bounding box of the MBRs of its child nodes. The MBR associated with a leaf node would be a smallest rectangle that spatially contains a subsequence. To insert a subsequence of length n into a multidimensional index structure, we need to create a minimum bounding rectangle R in the N ($N < n$) dimensional space for it. Then it is inserted into the multidimensional index structure based on its MBR R, $R = (L, H)$, where $L = \{l_1, \ldots, l_N\}$ and $H = \{h_1, \ldots, h_N\}$ are the lower and higher endpoints of the major diagonal of R. The insert algorithm is introduced in (Guttman, 1984). The remaining steps are straightforward.

Note that, in the case of a multidimensional index structure is used, if $m < w$ we can use a variable to accumulate the forecasted sequences until m is equal to w. At that time we can insert the accumulated sequence into the used index structure without need to rebuild the whole index structure in step 1.

4. Experimental evaluation

• Datasets and experimental environment

We test our proposed method on five real datasets: Carbon Dioxide (CD), Fraser River (FR), Milk Production (MP), Gold prices (GP) and Stock index (SI). The datasets for experiment are described in table 1. Figure 11 shows the plots of the above datasets.

Table 1. Description of five datasets for experiment.

1	Carbon Dioxide dataset, from 1/1959 to 12/2008 (https://datamarket.com/data/set/1c8z/atmospheric-carbon-dioxide-record-from-mauna-loa#!ds=1c8z&display=line).
2	Fraser river dataset, from 1/1913 to 12/1990 (http://datamarket.com/data/set/22nm/#!display=line&ds=22nm).
3	Milk Production, from 1/1962 to 12/1975 (https://datamarket.com/data/set/22ox/monthly-milk-production-pounds-per-cow-jan-62-dec-75#!ds=22ox&display=line).
4	Gold Prices (USD/oz), from 02/01/2009 to 28/05/2013 (http://www.forexpros.com/commodities/gold-historical-data).
5	Stock index S&P 500, from 03/01/2007 to 31/12/2012 (http://www.forexpros.com/indices/usspx-500-historical-data).

The proposed method is experimented with some different pattern lengths, predicted sequences of length 1 and for each experimental dataset we test with some k values for k-nearest neighbor then choose the best one which minimizes the prediction error. The length of predicted sequences is 1 since only one-step prediction is considered in this study.

The performance of these prediction methods is compared on all segments of the dataset and the arithmetic means of errors in the predictive duration is calculated. The methods are implemented with Microsoft Visual C# and experiments are conducted on a Core i3, Ram 2GB.

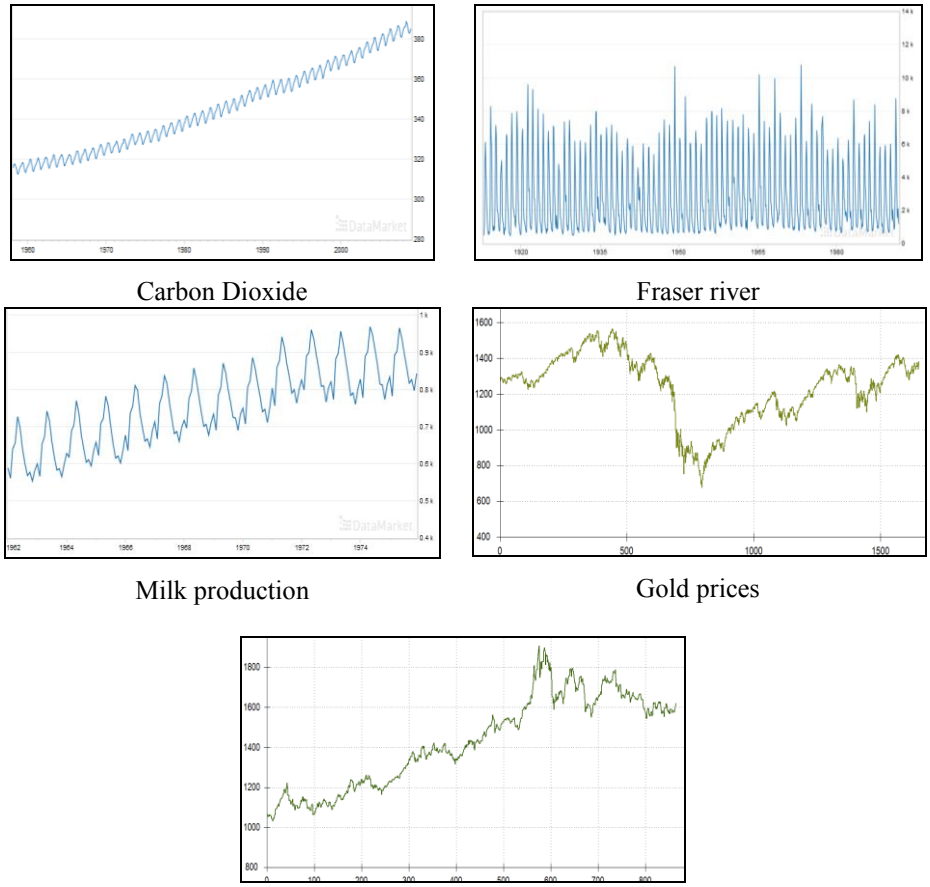

Fig. 11 Five different datasets

- **Evaluation criteria**

In this study we use the mean absolute error (MAE), the root-mean-square error (RMSE) and the coefficient of variation of the RMSE, called CV(RMSE) to measure the prediction accuracy. They are defined as follows.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_{obs,i} - Y_{\mathrm{model},i}\right| \tag{11}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Y_{obs,i} - Y_{\mathrm{model},i})^2}{n}} \tag{12}$$

$$CV(RMSE) = \frac{RMSE}{\overline{Y_{obs}}} \tag{13}$$

where Y_{obs} is observed values and $Y_{\mathrm{model},i}$ is modeled value at time i.

- **Experimental results**

In practice, the suitable value of k is usually small for time series without noise, since only a small number of different values for k must to be considered to find the optimal value (Alicia et al., 2003). Therefore to examine the impact of k on the predictive accuracy, for each dataset we test with some k values (from 1 to 10), then average the predictive errors, and finally choose the best one which minimizes the prediction error. For brevity, we only show the predictive mean absolute error (MAE) of the experiment on one dataset with k from 1 to 10 in table 2.

Table 2. The predictive errors of the experiment on Carbon Dioxide dataset with *k* from 1 to 10.

k	MAE	*k*	MAE
1	0.05165	6	0.04298
2	0.04219	7	0.04392
3	0.04179	8	0.05137
4	0.04384	9	0.04358
5	0.03941	10	0.05337

The experimental result shows that the predictive errors vary with different values of *k*. In this experiment we see that the predictive error is minimum if the chosen *k* is 5.

As for the pattern length *w*, the value of *w* is dependent on the seasonal length of the time series, for example, if the seasonal length is *s*, we should set the value of *w* to *s*. To examine the impact of the pattern length on the predictive accuracy when to forecast on more general-purpose time series data which have trend and seasonal variations, we test with some different pattern lengths. Table 3 shows the predictive errors of the experiment on Carbon Dioxide dataset with different pattern lengths. The experimental result shows that the predictive accuracy is varied when to experiment on the same dataset with different pattern lengths. The experimental results show that the predictive accuracy of our proposed method is always better than those of the pattern matching-based method under Euclidean distance or DTW.

Table 3. Experimental result from the Carbon Dioxide dataset with the best *k* and different pattern lengths.

Length of pattern	MAE			RMSE			CV(RMSE)		
	Proposed method	DTW-based method	Euclid-based method	Proposed method	DTW-based method	Euclid-based method	Proposed method	DTW-based method	Euclid-based method
3	0.03941	0.05054	0.04966	0.04959	0.06065	0.05941	0.24550	0.28876	0.30012
6	0.03941	0.05054	0.04966	0.05434	0.06610	0.06451	0.26411	0.30667	0.30583
9	0.03937	0.05014	0.04947	0.05678	0.06871	0.06607	0.24933	0.29703	0.29020
12	0.03941	0.05054	0.04966	0.05932	0.07181	0.06843	0.24402	0.29658	0.28127

Table 4 shows the forecasting result in the last five years on the Carbon Dioxide dataset with the best *k* and pattern of length 12. At the end of the table is the arithmetic mean of errors in years forecasted. The experimental results show that the predictive accuracy of our proposed method is better than those of the pattern matching-based method under Euclidean distance or DTW.

Table 4. Experimental result in the last five years on the Carbon Dioxide dataset.

Year	MAE			RMSE			CV(RMSE)		
	Proposed method	DTW-based method	Euclid-based method	Proposed method	DTW-based method	Euclid-based method	Proposed method	DTW-based method	Euclid-based method
1	0.05976	0.06609	0.06610	0.09693	0.10326	0.09847	0.40797	0.43460	0.41444
2	0.02056	0.04377	0.03172	0.03377	0.05916	0.05398	0.14502	0.25408	0.23183
3	0.03880	0.04247	0.05732	0.06323	0.06386	0.07774	0.23550	0.23786	0.28956
4	0.03325	0.04731	0.04237	0.04673	0.06459	0.05124	0.18601	0.25712	0.20397
5	0.04469	0.05307	0.05078	0.05594	0.06816	0.06071	0.24560	0.29923	0.26655
Mean	0.03941	0.05054	0.04966	0.05932	0.07181	0.06843	0.24402	0.29658	0.28127

For brevity, in table 5 we only show the summary of results obtained from the experiments on the five datasets with the best *k* and pattern of length 12. The values in this table are the arithmetic means of errors in years forecasted. The experimental results on the above five real datasets show that the

prediction errors of our proposed method are lower than those of the pattern matching-based method under Euclidean distance or DTW in all these five datasets. It means that the prediction method based on pattern matching using afffine combination of two measures, Euclidean distance and DTW, could produce a prediction result better than that of the pattern matching-based prediction method under Euclidean distance or DTW in terms of accuracy.

Table 5. The summary of results obtained from the experiment on five datasets.

Dataset	MAE			RMSE			CV(RMSE)		
	Proposed method	DTW-based method	Euclid-based method	Proposed method	DTW-based method	Euclid-based method	Proposed method	DTW-based method	Euclid-based method
CD	0.03941	0.05054	0.04966	0.05932	0.07181	0.06843	0.24402	0.29658	0.28127
FR	0.03458	0.04586	0.04587	0.05000	0.06052	0.06019	0.24226	0.29290	0.29474
MP	0.02547	0.03562	0.03392	0.03262	0.04094	0.03901	0.04485	0.05666	0.05350
GP	0.02605	0.02891	0.02864	0.04265	0.04436	0.04441	0.06321	0.06576	0.06582
SI	0.01435	0.01681	0.01778	0.01908	0.02106	0.02225	0.02398	0.02646	0.02795

5. Conclusions

In this paper, we propose an improved pattern matching-based prediction method using affine combination of two measures, Euclidean distance and DTW, for forecasting more general-purpose time series data which have trend and seasonal variations.

Our experiments on the five above datasets show that with the appropriate *k* value, our proposed method performs better than the pattern matching-based prediction method under Euclidean distance or DTW in terms of prediction accuracy. This result also implies that Euclidean distance and DTW can be complementary in time series prediction.

The contribution of this paper is summarized as follows: First, an improved pattern matching-based prediction method using affine combination of two measures is introduced. Then several experiments on different datasets are conducted to confirm the efffectiveness of our proposed method.

As for future work, we plan to experiment our proposed forecasting method on some other time series datasets and study how to determine the best *k* automatically in the module that performs the *k*-nearest neighbor search. Besides, we intend to expand this framework by using more distance measures to combine the benefits of different distance measures.

References

Alicia, T. L., José Cristóbal, R. S., José Luis, M. R., Jesús Manuel, R. S., Antonio, G. E., (2003), 'Influence of kNN-Based Load Forecasting Errors on Optimal Energy Production', *Progress in Artificial Intelligence, Lecture Notes in Computer Science*, Volume 2902, pp 189-203.

Alicia, T. L., Jesús Manuel, R. S., Riquelme, J. C., Antonio, G. E. and José, L. M. R., (2004), 'Time series prediction: Application to the short term electric energy demand', *Lecture Notes in Artificial Intelligence*, pp. 577-586.

Arroyo, J. and Mate, C., (2009), 'Forecasting histogram time series with k-nearest neighbours methods'. International Journal of Forecasting 25, pp. 192–207.

Álvarez, F. M., Alicia, T. L., Riquelme, J. C. and Ruiz, J. S. A. (2011), 'Energy Time Series Forecasting Based on Pattern Sequence Similarity', *International Journal of Knowledge and Data Engineering*, IEEE Transaction, vol. 23, Issue 8, pp. 1230 – 1243.

Berndt, D. and Clifford, J., (1994), 'Using dynamic time warping to find patterns in time series' in *AAAI Workshop on Knowledge Discovery in Databases*, pp. 229-248.

Balkin, S. D. and Ord, J. K., (2000), 'Automatic neural network modeling for univariate time series', *International Journal of Forecasting*, vol.16, pp. 509–515.

Beckman, N., Kriegel, H.P., Schneider, R. and Seeger, B. (1990), 'The R*-tree: An efficient and robust access method for points and rectangles', In: *Proc. of 1990 ACM-SIGMOD Conf.*, Atlantic City, NJ, May 1990, pp. 322-331.

Chatfield, C. (2000), Time-series forecasting, New York, NY: Chapman and Hall, Inc.

Cadenas, E. and Rivera, W. (2009), 'Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks', *Renewable Energy*, vol. 34, no. 1, pp. 274–278.

Cai, P., Wang, Y., Lu, G. and Chen, P., (2015), 'An Improved k-Nearest Neighbor Model for Road Speed Forecast Based on Spatiotemporal Correlation', *CICTP 2015*, pp, 342-351.

Gelper, S., Fried, R. and Croux, C., (2010), 'Robust forecasting with exponential and Holt-Winters smoothing', *International Journal of Forecasting*, vol. 29, pp. 285-300.

Ghiassi, M., Saidane, H. and Zimbra, D. K., (2005), 'A dynamic artificial neural network model for forecasting series events', *International Journal of Forecasting*, vol.21, pp. 341–362.

Guttman, A., (1984), 'R-trees: A dynamic index structure for spatial searching', In: *Proceedings ACM SIGMOD conference*, pp 47–57

Heravi, S., Osborn, D. R. and Birchenhall, C. R., (2004), 'Linear versus neural network forecasting for European industrial production series', *International Journal of Forecasting*, vol.20, pp. 435–446.

Huang, Z., Shyu, M. L., Tien, J. M., (2012), 'Multi-Model Integration for Long-Term Time Series Prediction'. In Information Reuse and Integration (IRI) proceeding from IEEE 13th International Conference, pp. 116-123

Itakura, F. (1975), 'Minimum prediction residual principle applied to speech recognition', *IEEE Trans. Acoustics, Speech, and Signal Proc.*, Vol. ASSP-23, pp. 52-72.

Jiang, Y., Li, C., Han, J., (2009), 'Stock temporal prediction based on time series motifs', In: *Proc. of 8th Int. Conf. on Machine Learning and Cybernetics*.

Keogh, E. and Kasetty, S., (2002), 'On the Need for Time series Data Mining Benchmarks: A Survey and Empirical Demonstration', In *the 8th ACM SIGKDD*, pp. 102-111.

Keogh, E., Ratanamahatana, C. A., (2005), 'Exact indexing of dynamic time warping', *Journal of Knowledge and Information Systems*. Vol. 7 Issue 3, pp. 358 – 386.

Kang, I.-B., (2003), *Multi-period forecasting using different models for different horizons- An application to U.S. economic time series data*. International Journal of Forecasting, vol.19, pp. 387–400.

Kim, J. H., (2003), 'Forecasting autoregressive time series with bias-corrected parameter estimators', *International Journal of Forecasting*, vol.19, pp. 493–502.

Kim, K. J., (2003), 'Financial time series forecasting using support vector machines', *Neuro-computing*, vol. 55, pp. 307-319.

Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S., (2000), 'Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases', In *Proceedings of Conference on Knowledge and Information Systems*, pp. 263-286.

Mueen, A., Keogh, E., Zhu, Q., Cash, S. and Westover, B., (2009), 'Exact Discovery of Time Series Motifs', In: *Proc. of SIAM Int. Conf. on Data Mining*, pp. 473-484.

Nayak, Richi and te Braak, Paul (2007), 'Temporal Pattern Matching for the Prediction of Stock Prices', In *Ong, K.-L. and Li, W. and Gao, J., Eds. Proceedings 2nd International Workshop on Integrating Artificial Intelligence and Data Mining* (AIDM 2007) CRPIT, 84, pages pp. 99-107, Gold Coast.

Radhika, Y. and Shashi, M., (2009), 'Atmospheric Temperature Prediction using Support Vector Machines', *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, 2009, pp. 55-58.

Sakoe, H. and chiba, S. (1978), 'Dynamic programming algorithm optimization for spoken word recognition', *IEEE Trans. Acoustics, Speech, and Signal Proc.*, Vol. ASSP-26. pp. 43-49.

Sorjamaa, A., Hao, J., and Lendasse, A., (2005), 'Mutual Information and k-Nearest Neighbors Approximator for Time Series Prediction'. W. Duch et al. (Eds.): ICANN 2005, LNCS 3697, pp. 553 – 558. Springer-Verlag Berlin Heidelberg.

Son, T. N., Le, H. N. and Anh, T. D., (2013), 'Time Series Prediction Using Pattern Matching'. In *Proc. of International Conference on Computating, Management and TeleCommunications (ComManTel 2013)*, Ho Chi Minh city, Vietnam, January 21-24, 2013, pp. 401-406.

Son, T. N. and Anh, T. D., (2016), 'Discovery of time series k-motifs based on multidimensional index'. *International Journal of Knowledge and Information Systems*, Springer, 46(1), pp. 59-86, 5 Jan 2016.

Tkacz, G., (2001), 'Neural network forecasting of Canadian GDP growth', *International Journal of Forecasting*, vol.17, pp. 57–69.

Zhang, G. P. and Qi, M., (2005), 'Neural Network Forecasting for Seasonal and Trend Time Series', *European Journal of Operational Research*, vol. 160, pp. 501-514.

Zhang, L., Liu, Q., Yang, W., Wei, N., Dong, D., (2013), 'An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction', In *Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings* from the 13th COTA International Conference of Transportation Professionals (CICTP2013), Vol. 96, 2013, pp. 653–662.