PATTERN-SIMILARITY-BASED MODEL FOR TIME SERIES PREDICTION

SAURABH BHARDWAJ,1 SMRITI SRIVASTAVA,2 AND J. R. P. GUPTA2
1ECE Department, Galgotias University, Greater Noida, India
2ICE Department, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, India

This research proposes a pattern/shape-similarity-based clustering approach for time series prediction. This article uses single hidden Markov model (HMM) for clustering and combines it with soft computing techniques (fuzzy inference system/artificial neural network) for the prediction of time series. Instead of using distance function as an index of similarity, here shape/pattern of the sequence is used as the similarity index for clustering, which overcomes few of the shortcomings associated with distance-based clustering approaches. Underlying hidden properties of time series are captured with the help of HMM. The prediction method used here exploits the pattern identification prowess of the HMM for cluster selection and the generalization and nonlinear modeling capabilities of soft computing methods to predict the output of the system. To see the validity of the proposed method in the real-life scenario, it is tested on four different time series. The first is a benchmark Mackey–Glass time series, which is tested for delay parameters $D_{17}$ and $D_{30}$. The remaining time series are monthly sunspot data time series, Laser data time series and the last is Lorenz attractor time series. Simulation results show that the proposed method provide a better prediction performance in comparison with the existing methods.

Received 14 August 2012; Revised 20 May 2013; Accepted 13 June 2013

Key words: hidden Markov models, fuzzy inference systems, artificial neural networks, shape-based clustering, time series prediction.

1. INTRODUCTION

Time series forecasting or prediction is a successor of the process of time series analysis and has now become a topic of widespread research in various fields such as econometrics, business, biology, physics, and meteorology. A detailed discussion on applications of time series can be seen in Schreiber (1999). Time series generally is a result of systems whose model and parameters are not very clearly known and the only source of information is the past data. Fundamentally, the goal of time series prediction is to estimate some future values based on current and past data samples. The main challenges for predicting the time series are that they are nonstationary (often in the form of switching between regimes) and have different lengths; thus, the traditional methods based on dimension space cannot be directly applied to the time series modeling and forecasting. In previous years, model-based analysis of time series has attracted more research interest. In model-based forecasting of the time series, previous observations are collected, and on the basis of these observations, models are made to capture the underlying data generation process. Then, the model is used to predict the future. The model-based approaches can be basically divided into two categories: statistical methods and soft computing methods. Traditionally, sophisticated statistical techniques were applied over available data points in an effort to predict the future. The frequently used statistical techniques are autoregressive, moving average and auto regressive integrated moving average (ARIMA) methods (Harvey 1989; Montgomery et al. 1990; West and Harrison 1997; Box et al. 1994). There were many problems associated with these techniques such as ARIMA was limited by the requirement of stationarity of

Address correspondence to Saurabh Bhardwaj, ECE Department, Galgotias University, Greater Noida, Gautam Buddha Nagar, India 201308; e-mail: bsaurabh2078@gmail.com

© 2013 Wiley Periodicals, Inc.
time series. The second problem was that all these techniques assumed that the time series is generated by a linear process while almost all the practical process are nonlinear, and hence, linear modeling methods were not adequate. The nonlinear statistical predictors such as threshold, exponential, polynomial and bilinear predictors were proposed to add more precision to prediction (Voitcu and Wong 2004). However, the selection of the suitable nonlinear model, as well as the computation of its parameters, is a difficult task for a practical problem without a priori knowledge about the time series under consideration. Unlike these traditional forecasting approaches, soft computing approaches are able to adapt nonlinearity and approximate complex relationships without extensive data or knowledge. Furthermore, in soft computing methods, there is no need to specify the structure of a model a priori, which is needed in the classical regression analysis methods. Neural networks can approximate complex mapping functions between their input and output spaces without a prespecified model. Over the last decade, artificial neural networks (ANNs) have been used among various time series prediction approaches and have demonstrated a good performance in many cases (Yao and Liu 1997; Atiya et al. 1999; Coyle et al. 2005). Neural networks can uniformly approximate any real continuous nonlinear function to an arbitrary degree of accuracy (Cybenko 1989; Homik 1993; Blum and Li 1991; Chung Tsoi and Scarselli 1998; Feuring and Lippe 1999; Homik et al. 1989). This result is proved in Homik et al. (1989) with the help of the Stone–Weirstrass theorem. Similarly on the aforementioned guidelines, many authors have proved that fuzzy logic systems can also be used as a universal approximator (Mendel 1995; Kosko 1994; Castro and Delgado 1996; Wang 1992). A brief discussion on the fuzzy logic systems as universal approximators can be seen in Wang (1992). Because of this property of fuzzy logic systems, they have been used for the prediction of time series (Castillo and Melin 2002).

It is a fundamental observation in science that a complex problem can be solved by partitioning the problem space into smaller components and analyzing these components individually. Solutions to these individual components are easier to find, and these solutions can be combined to provide a better solution or provide (in this case) a more precise estimate than can be obtained by a single model. Conventionally, single models were used for the prediction of time series, and although theoretically a single model can emulate any function, it is often very difficult to extract such an accurate single model from the data.

In this research, powerful pattern identification properties of hidden Markov model (HMM) are used for the clustering of data. In the proposed algorithm for time series prediction using shape-based clusters, distance functions are not used for the clustering of data because of the following: First, there is no natural distance function between time series, and required qualitative behavior of time series is not easy to obtain with a distance function (Knab et al. 2003). Second, as shown in Haixun and Jian (2008), the distance functions are not always adequate for capturing correlations among the objects. It is also shown that strong correlations may still exist among a set of objects even if they are far apart. Earlier approaches in which clustering is used as a first step for the forecasting of time series are shown in Martínez-Álvarez et al. (2007) and Martínez-Álvarez et al. (2011).

Hidden Markov models are the dominant models for the sequential data. Although HMMs have been extensively used in speech recognition, pattern recognition and time series prediction problems, they have not been widely used for the clustering problems, and only few articles can be found in the literature. Many researchers have used single sequences to train the HMMs and proposed different distance measures based on a likelihood matrix obtained from these trained HMMs. Clustering of sequences using HMM were introduced in Smyth (1997). In this, a log-likelihood (LL)-based scheme for automatically determining the number of clusters in the data is proposed. A similarity-based clustering of sequences using HMMs is presented in Bicego et al. (2004). In this, a new representation space is built
in which each object is described by the vector of its similarities with respect to a predetermined set of other objects. These similarities are determined using LL values of HMMs. An HMM-based clustering method was also proposed in Hassan and Nath (2005), which utilized LL values as the similarity measures between data points. The method was useful for finding the number of clusters in the dataset with the help of LL values, but it is tough to actually obtain the data elements for the clusters as the threshold for the clusters was estimated by simply inspecting the graph of LL.

This research is an application of our previous work for time series prediction in which an HMM-based unsupervised clustering algorithm was proposed, which uses shape similarity as a measure to capture the correlation among the objects (Srivastava et al. 2013). Here, the hidden state information of HMM is utilized as a tool to obtain the similar patterns among the objects. Further, these shape-based clusters are processed by soft computing techniques for time series prediction.

The distinguished characteristics of the proposed approach are as follows:

(1) Here, a novel clustering algorithm is used that divides the data into clusters depending upon the shape of the input sequence; the clusters are further processed by the function approximation model to predict the output. The novel clustering paradigm has the following features:

1.1 This is an unsupervised clustering algorithm in which the prior knowledge about the number of clusters is not required. Also, there is no restriction on the size of the clusters. The size of the clusters is automatically adjusted by the algorithm.

1.2 Instead of using distance functions as the similarity index, the shape similarity is used as a measure to capture the correlation among the objects, which overcomes few of the problems associated with distance-based clustering approaches.

1.3 The shortcoming of previous HMM-based clustering approaches in which the number of HMMs required were equal to that of the number of sequences/classes (Smyth 1997; Bicego et al. 2004) is removed by utilizing only single HMM for clustering and hence reducing the computational time and complexity considerably.

(2) If a single module is used to train an entire dataset, the computing power required is very high. An increase in data elements would increase the nonlinearity expected by the ANN/fuzzy inference system (FIS), hence increasing the computational complexity of the process. To overcome this problem, the training process is divided into small clusters.

(3) Here, the property of HMM to deal with time sequential structures combine with the generalization capability of soft computing method is used for the prediction of time series.

During the last few decades, statistical and soft computing techniques have been widely used alone for the prediction of time series (Chen et al. 2004; Chen et al. 2006; Mao et al. 2005; Park et al. 1996; Milanese and Novara 2005). Because individual methods are not perfect for attaining the goal, two or more methodologies are combined in a way that their weaknesses are compensated by each other, and advantages of both methodologies are used to achieve the solution. Because of the similarities in both approaches (statistical and soft computing), hybrid algorithms were developed, which used the concepts from both paradigms (Chandra and Omlin 2006; Firoiu and Cohen 2002; Rynkiewicz 1999). In this research, both the methods are distinct, and only intermediary results are exchanged. Earlier approaches to such models are summarized in Willinski et al. (1998). The authors have used ANNs and FISs as the soft computing methods for the prediction of time series. A short comparison between fuzzy logic and ANN concepts can be seen in Mendel (1995).
The article is organized as follows. Section 2 briefly describes the HMM. In Section 3, the shape-based clustering approach is explained. Section 4 shows how the shape-based clustering model is used to predict the time series. Experimental results are provided to illustrate the effectiveness of the proposed algorithms in Section 5. Finally, conclusions are drawn in Section 6.

2. HIDDEN MARKOV MODEL

Hidden Markov model (Rabiner 1989; Blimes 1998) springs forth from Markov processes or Markov chains. It is a canonical probabilistic model for the sequential or temporal data. It depends upon the fundamental fact of real world, “Future is independent of the past but driven by the present.” The HMM is a doubly embedded stochastic process, where the final output of the system at a particular time depends upon the state of the system and the output generated by that state.

There are two types of HMMs: discrete HMMs (DHMMs) and continuous density HMMs (CDHMMs). These are distinguished by the type of data that they operate upon. DHMMs operate on quantized data or symbols; on the other hand, the CDHMMs operate on continuous data, and their emission matrices are the distribution functions. The basic notations of HMM are as shown in Table 1.

There are three major design problems associated with an HMM outlined here:

1. Given the observation sequence \( \{O_1, O_2, O_3, \ldots, O_T\} \) and model \( \lambda(A, B, \pi) \), the first problem is the computation of probability of the observation sequence \( P(O|\lambda) \).
2. The second problem finds the most probable state sequence \( Q\{q_1, q_2, \ldots, q_T\} \).
3. The third problem is related to the choice of the model parameters \( \lambda(A, B, \pi) \), such that the probability of the observation sequence, \( P(O|\lambda) \), is the maximum.

The solution to the aforementioned problems emerges from three algorithms: forward, Viterbi and Baum–Welch (Rabiner 1989).
2.1. Continuous density hidden Markov model

Let \( O = \{ O_1, O_2, \ldots, O_T \} \) be the observation sequence and \( Q = \{ q_1, q_2, \ldots, q_T \} \) be the hidden state sequence. Now, we briefly define the expectation maximization (EM) algorithm for finding the maximum likelihood estimate of the parameters of an HMM given a set of observed feature vectors. The \( Q \) function is generally defined as

\[
Q(\lambda, \lambda^{(i-1)}) = \sum_{q \in \ell} \log P(O, Q | \lambda) P(O, Q | \lambda^{(i-1)})
\]

1

The E-step and the M-step of EM algorithm are as follows:

E-step:

\[
Q(\lambda, \lambda^{(i-1)}) = \sum_{q \in \ell} \sum_{m \in M} \log P(O, Q, m | \lambda) P(O, m | \lambda^{(i-1)})
\]

M-step:

\[
\lambda^{(i)} = \arg \max_{\lambda} \{ Q(\lambda, \lambda^{(i-1)}) \} \text{ + constainment}
\]

Final result: when \( i \to \infty, \lambda^{(i-1)} \to \lambda_{ML}. \)

The optimized equations for the parameters of the mixture density are (Blimes 1998)

\[
\mu_{il} = \frac{\sum_{t=1}^{T} O_t P(q_t = i, m_{q,t} = 1 | O, \lambda^{(i-1)})}{\sum_{t=1}^{T} P(q_t = i, m_{q,t} = 1 | O, \lambda^{(i-1)})}
\]

\[
\sum_{il} = \frac{\sum_{t=1}^{T} (O_t - \mu_{il})(O_t - \mu_{il})^T P(q_t = i, m_{q,t} = 1 | O, \lambda^{(i-1)})}{\sum_{t=1}^{T} P(q_t = i, m_{q,t} = 1 | O, \lambda^{(i-1)})}
\]

\[
c_{il} = \frac{\sum_{t=1}^{T} P(q_t = i, m_{q,t} = 1 | O, \lambda^{(i-1)})}{\sum_{t=1}^{T} \sum_{l=1}^{M} P(q_t = i, m_{q,t} = 1 | O, \lambda^{(i-1)})}
\]

3. SHAPE-BASED CLUSTERING

Cluster analysis is a method of creating group of objects or clusters in such a way that the objects in one cluster are very similar to each other while the objects in different clusters are quite different. Data clustering algorithms could be generally classified into the following categories (Gan et al. 2007): hierarchical clustering, fuzzy clustering, center-based clustering, search-based clustering, graph-based clustering, grid-based clustering, density-based clustering, subspace clustering and model-based clustering algorithms. Every clustering algorithm is based on the index of similarity or dissimilarity between data points. Many authors have used distances as the index of similarity. Generally, different distance functions such as Euclidean distance, Manhattan distance, Minkowski distance and Mahalanobis distance (Xu and Wunsch 2005) are employed for clustering of data. Although the distance functions have been widely used for clustering of data, these distance functions are not always effective to capture the correlations among the objects. In fact, strong correlations may still exist among a set of objects even if their distances are far apart as measured by the distance functions. Figure 1 shows four objects with five attributes among a set of 300 objects that were allotted in different clusters when the segmental \( k \)-means was applied to partition them in six clusters. It is clear from Figure 1 that these objects physically have the same pattern of shape and also have the strong correlation among each other, which is shown with the help of correlation matrix between the four data elements in
Figure 1. Clustering results with the segmental $k$-means.

Table 2. Correlation Coefficient Matrix.

<table>
<thead>
<tr>
<th></th>
<th>Data-1</th>
<th>Data-2</th>
<th>Data-3</th>
<th>Data-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-1</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data-2</td>
<td>0.988</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data-3</td>
<td>0.955</td>
<td>0.989</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>Data-4</td>
<td>0.905</td>
<td>0.959</td>
<td>0.990</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Table 2. Taking motivation from here, we have extended the basic concept of shape-based batching model as introduced in Srivastava et al. (2010) and Bhardwaj et al. (2011). Earlier, it was shown that by carefully observing the datasets and their corresponding LL, it is possible to find the shape of the input variation for certain value of LL, but further, it is found that to detect the shape by simply observing is not always easy. Moreover, in some datasets, it is very difficult to determine the threshold for batch allocation. Although the states are hidden for many practical applications, there is often some physical significance attached to the states of the model (Rabiner 1989). In this article, it is found that the pattern of objects corresponding to any particular state of HMM is highly correlated and have a different pattern or uncorrelated with the objects corresponding to any other state. The concept of shape-based batching is modified here, and now shape is a function of the state and not of LL.

Here, unsupervised clustering algorithm is proposed in which it is important to note that the number of clusters is not fixed and the algorithm automatically decides the number of clusters. The whole procedure is as shown in Figure 2. As a first step, the number of clusters in the dataset is obtained. The steps for obtaining the number of clusters are as follows: Estimate the CDHMM model parameters $\lambda(T, B, \pi)$ for the entire input dataset using the Baum–Welch/EM algorithm, for the appropriate values of the state $Z$ and mixture components $m$. Once the HMM has been trained, the forward algorithm is used to compute the value of $P(O|\lambda)$, which can then be used to calculate the LL of each row of the dataset. LL values give an indication of how the two vectors are similar to each other. To obtain the number of clusters, LL values are arranged in ascending order by using a standard sorting algorithm. The next step is to detect the sharp changes in the value of LL, which is obtained by calculating the difference between two consecutive LL values. A cluster change is
indicated when the difference in consecutive LL values exceeds a desired threshold. All such discontinuities in the LL graph are used to acquire the number of clusters.

Now, after obtaining the information about the number of clusters, initialize the value of the parameters of the HMM. This includes initialization of transition matrix $T$, initialization matrix $\pi$ and the mixture component $m$ for each state. Take the number of states equal to the number of clusters. The CDHMM is trained using the Baum–Welch/EM algorithm for the entire input dataset. After freezing the HMM parameters, the next step is to find the optimal state sequence, with the help of “Viterbi algorithm” by taking the entire input dataset as $D$-dimensional observation vector sequence, where $D$ is the number of attributes in the dataset. After doing this, one important thing is observed: the data vectors that are associated with the same state have identical shape (have strong correlation among each other), while the data vectors with different states have no similarity (or very less correlation) in their shapes. Thus, once the optimal value of hidden state sequence is deduced, the next step is to divide the data into clusters according to their states. After this step, each cluster have almost identical shape, but by simply observing the clusters, it is difficult to find the required threshold value for shape-based similarity; thus, an attempt is made to obtain the appropriate values of $Z$ and $m$ for the required shape-based clusters by calculating the value of correlation coefficient among the data vectors into the clusters. Here, the Pearson $R$ model (Shardanand and Maes 1995) comes handy for finding the coherence (correlation) among a set of objects. The correlation between the two objects $x_1$ and $x_2$ is defined as follows:

$$
\frac{\sum (x_1 - \bar{x}_1)(x_2 - \bar{x}_2)}{\sqrt{\sum (x_1 - \bar{x}_1)^2}\sqrt{\sum (x_2 - \bar{x}_2)^2}}
$$

where $\bar{x}_1$ and $\bar{x}_2$ are the mean of all attribute values in $x_1$ and $x_2$, respectively. It may be noted that Pearson $R$ correlation measures the correlation between two objects with respect to all the attribute values. A large positive value indicates a strong positive correlation, while a large negative value indicates a strong negative correlation. Now, the correlation coefficients can be used as a threshold value of the similarity between the data vectors in the clusters, and by using this value as a threshold, the appropriate value of $Z$ and $m$ can be determined for the shape-based clusters. Using these basic criteria, an algorithm was developed, which arranged the data into clusters.
3.1. Algorithm

The following iterative algorithm is used to obtain the shape-based clusters.

Step 1: Estimate HMM parameters for the entire input dataset.
Step 2: Calculate LL for each row of the dataset.
Step 3: Sort LL values.
Step 4: Set threshold for LL.
Step 5: Find discontinuities and obtain the number of clusters.
Step 6: Take the number of states equal to the number of clusters and estimate the HMM parameters $\lambda(T, B, \pi)$ for the entire input dataset by taking the appropriate value of the mixture components $m$.
Step 7: Calculate the optimal value of hidden state sequence with the help of Viterbi algorithm by taking the input as a $D$-dimensional observation vector.
Step 8: Rearrange the complete dataset according to their state values.
Step 9: Calculate correlation matrix by using the Pearson $R$ model as in (11).
Step 10: Change the value of $m$ and repeat steps 6–9 until the required tolerance of correlation is achieved.

To demonstrate the effectiveness of the proposed paradigm, it is applied on the benchmark clustering problem of Iris Plants Database. The dataset contains three classes of 50 instances each, where each class refers to a type of Iris plant. Figure 3 shows the patterns of Iris data before clustering. After executing steps 1–3 of the proposed algorithm, sorted LL values are obtained. Figure 4 shows the graph of sorted LL values in ascending order. After selecting the suitable threshold (20 in this case), discontinuities in the sorted LL values are found to obtain the number of clusters.

After the execution of steps 6–10 of the proposed algorithm, Table 3 is obtained. The description of this table is as follows: Rows 1–4 show the four attribute values of the IRIS data. Row 5 shows the number of data vector. Row 6 shows the LL values corresponding to data vectors. Row 7 shows the optimized state values associated with that particular data vector.

Because of the limitation of page width, it is not possible to show the complete table. So only five values of each class are shown. The values of LL in the table are only displayed to show the effectiveness of the proposed method over LL-based clustering method. As it is clear from the table that the LL values between the data element “54” is almost equal to the LL value of data element “105,” these two elements belong to two different clusters.

![Figure 3. Iris plant data patterns.](image-url)
Therefore, if data are partitioned with the help of LL, then it would not be classified accurately. On the basis of the aforementioned discussion, it can be said that the clustering based upon the LL values is not always adequate, while it is clear from Table 3 that the “states” as shown in row 7 of the table partitions the data accurately, and in this particular dataset (Iris dataset), the misclassification is zero, which means 100% accuracy. The plot of the three clusters obtained after the application of the proposed algorithm is as shown in Figure 5, and the actual parameters of the model are shown in the Table 4.

4. SHAPE-BASED CLUSTERING USED FOR TIME SERIES PREDICTION

For prediction, preprocessing of time series is required. The time series such as Mackey–Glass time series (MGTS) shows stochastic behavior in time and frequency domain and deterministic behavior in phase space structures. As a first step, the time series is converted into phase space. The phase space of time series is generated by using time delay embedding parameter $\tau$ and embedding dimension $d$. The conversion of time series into phase space and time delay embedding is based upon “Takens’ embedding theorem” (Takens 1981). This theorem allows the phase space of the attractor to be reconstructed. The choice of $\tau$ determines the accuracy of the reconstructed attractor. If this value is small, then it will plot the attractor along a line, and if it is large, then it will not reveal the structure of the attractor.

The input test data is a one-dimensional continuous series

$$X = \{x_t, t = 1, 2 \ldots, N\}$$ (8)
Figure 5. Iris data cluster.

Table 4. Actual Parameters of the Model.

<table>
<thead>
<tr>
<th>Sigma(:,:,1)</th>
<th>Sigma(:,:,2)</th>
<th>Sigma(:,:,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2706 0.0833 0.1788 0.0545</td>
<td>0.1318 0.0972 0.0160</td>
<td>0.4061 0.0921 0.2972</td>
</tr>
<tr>
<td>Mean</td>
<td>Initial matrix</td>
<td>Transition matrix</td>
</tr>
<tr>
<td>5.936 5.006 6.589</td>
<td>0.0101 0.000 1.000</td>
<td>0.0479 0.000 0.000</td>
</tr>
<tr>
<td>5.936 5.006 6.589</td>
<td>0.0101 0.000 1.000</td>
<td>0.0479 0.000 0.000</td>
</tr>
<tr>
<td>0.0833 0.1064 0.0812 0.0405</td>
<td>0.1508 0.0115 0.0115</td>
<td>0.0921 0.0701 0.3087</td>
</tr>
<tr>
<td>Transition matrix</td>
<td>States =3</td>
<td>0.0479 0.000 0.000</td>
</tr>
<tr>
<td>0.0833 0.1064 0.0812 0.0405</td>
<td>0.1508 0.0115 0.0115</td>
<td>0.0921 0.0701 0.3087</td>
</tr>
<tr>
<td>0.1788 0.2273 0.0723 0.0487</td>
<td>0.2972 0.000 0.000</td>
<td></td>
</tr>
<tr>
<td>0.1788 0.2273 0.0723 0.0487</td>
<td>0.2972 0.000 0.000</td>
<td></td>
</tr>
<tr>
<td>0.0545 0.0405 0.0723 0.0487</td>
<td>0.2972 0.000 0.000</td>
<td></td>
</tr>
<tr>
<td>0.0545 0.0405 0.0723 0.0487</td>
<td>0.2972 0.000 0.000</td>
<td></td>
</tr>
</tbody>
</table>

where $t$ is the time index and $N$ the total number of observations. If the value of time delay factor is $\tau$ and the embedding dimension is $d$, then (12) in the phase space can be represented as shown in (13).

$$X_t = \left(x(t-(d-1)\tau), x(t-(d-2)\tau), \ldots, x_t\right) \in R^b$$

where $X$ is a matrix whose row vector is a point in the phase space given by (13). This transformation preserves the nonlinear dynamics of the original time series. Generally, the value of the time lag $\tau$ is determined using first minimum of mutual information technique.
or autocorrelation function, and the value of embedding dimension \( d \) can be estimated using false nearest neighbor method (Kennel et al. 1992). If for any particular time series the estimated values of \( \tau \) and \( d \) are 4 and 4, respectively, then the time series in phase space can be represented as

\[
X_t = \begin{pmatrix} x(t-12) \, x(t-8) \, x(t-4) \, x(t) \end{pmatrix}
\]

(10)

4.1. Estimation of delay time

The delay time is chosen in such a way that all the time-delayed elements are independent. If the value of \( \tau \) is too small, then the successive elements of the delay vector will be highly correlated. If the value of \( \tau \) is too large, then the adjacent elements will be highly uncorrelated. Generally, the autocorrelation or mutual information methods are used for estimating the value of \( \tau \). The mutual information method is better than the autocorrelation function because mutual information method measures general dependence of two variables while autocorrelation only measures linear dependence. To calculate the value of mutual information, a histogram for probability distribution of the data is created. The mutual information for the time delay \( \tau \) can be calculated as

\[
M(X_t, X_{(t+\tau)}) = \sum_{i,j} p_{ij}(\tau) \ln p_{ij}(\tau) - 2 \sum_i p_i(\tau) \ln p_i
\]

(11)

where \( p_i \) denote the probability that a data value is inside the \( i \)th bin of the histogram and \( p_{ij}(\tau) \) is the probability that \( X_t \) is in bin \( i \) and \( X_{t-\tau} \) is in bin \( j \). It is advised that one should look for the first minimum of time-delayed mutual information.

4.2. Estimation of the embedding dimension

One way to estimate the embedding dimension \( d \) is using the false nearest neighbor method (Kennel et al. 1992). This method can be described as follows: For each vector \( X_i^d \), its nearest neighbor can be calculated as follows:

\[
X_i^d = \arg\min_j \| X_i^d - X_j^d \|
\]

where \( \| X_i^d - X_j^d \| \) is the Euclidean distance between the two vectors. For a change of embedding dimension from \( d \) to \( d + k \) the distance between the two vectors can be calculated from (16):

\[
R_i = \sqrt{ \frac{\| X_i^{d+k} - X_j^{d+k} \|^2 - \| X_i^d - X_j^d \|^2}{\| X_i^d - X_j^d \|^2} }
\]

(12)

Every \( R_i \) is compared with a given threshold \( R_t \). If the value of \( R_i > R_t \), then \( X_i^d \) is marked as having a false nearest neighbor. Once the value of \( R_i \) is calculated for every vector, the next step is to count the number of vectors whose \( R_i \) value is greater than \( R_t \). The data vectors for which \( R_i > R_t \) is zero, and then the embedding dimension is determined to be sufficiently large. Now, after selecting the appropriate value of time delay \( \tau \) with the help of mutual information and the value of embedding dimension \( d \) using the nearest neighbor
method, the data are converted into phase space form. For prediction, the data in phase space form are divided into input–output structure as shown in Table 5.

In Table 5, it is important to note that each row of the table is a single point in the phase space. In the example shown earlier, each row vector consists of five elements: four input elements known as predictor variables and one output element known as dependent variable are represented in (13) and (14).

\[
\text{Input} \quad \mathbf{x}_{(t-12), (t-8), (t-4), (t)} \quad \text{(13)} \\
\text{Output} \quad x_{(t+4)} \quad \text{(14)}
\]

Figure 6 shows the data of three time series in phase space form. It can be easily observed from the figures that the structure of the data is complex and its training is tedious. Hence, the obvious solution to handle such a huge data is to divide it into clusters to make the training easier.

After converting the dataset into input/output form, the first step is to initialize the value of the parameters of CDHMM. The important point to note is that the training dataset here includes only the input/predictor variables and not the output/dependent variables, and hence, HMM is trained only for the input dataset. The output/dependent variable will be used later for the training of FISs or ANNs. After the application of shape-based clustering algorithm, the MGTS data are divided into clusters. Figure 7 shows the clustered data of MGTS into shape-based clusters based on the “states” and the “correlation coefficient” values. Here, the MGTS data are divided into six different clusters. It is observed from Figure 7 that the data in the similar clusters have similar shapes. The data vectors assigned to a particular cluster have the strong correlation with the other data vectors in that cluster. Here, the threshold value for correlation coefficient is taken as 0.7. Table 6 shows the correlation matrix among the data vectors of one particular cluster (cluster 6). Because of the limitation of page width, only 8 (1, 2, 3, 4, 47, 48, 49, and 50) of 50 rows have been shown. One important thing to observe here is that the size and number of clusters are not fixed, and
the proposed algorithm automatically decides the size and number of clusters. Shape-based clustering results of various other time series are shown in Appendix A. Once the data have been arranged into clusters on the basis of their shape, the next step involves the training of FISs or ANNs to predict the results. For an FIS/ANN, it is now easier to analyze the data after the shape-based clustering.

4.3. Prediction

After clustering, the output data are added to each cluster from the initial dataset so that the individual clusters are now arranged in the input/output form. These clusters are
trained with the help of FISs/ANNs described henceforth as well-trained clusters. To allocate the data vector in a proper cluster, a method that uses the LL ratio test is developed. For a given vector \( X \), the LL from a model \( \lambda \) is computed as \( \log P(X|\lambda) \). Given a set of \( N \) models \( \{\lambda_1, \lambda_2, \ldots, \lambda_N\} \) corresponding to \( N \) clusters, the cluster to which a given test sequence belongs can be calculated as

\[
P(X|\lambda_{\text{required}}) = F[P(X|\lambda_1) \ldots P(X|\lambda_N)]
\]

where \( F[] \) is maximum function of the likelihood values from the models \( \{\lambda_1, \lambda_2, \ldots, \lambda_N\} \).

To predict the output, clusters are trained in any one of the following way:

(a) Prediction with fuzzy inference system (SBCF)

(b) Prediction with artificial neural network (SBCN)

Once the test sequence is assigned a cluster, it is easy for the well-trained clusters to predict the output for the given test sequence. This output is calculated by both the training models (FISs/ANNs), and the results are compared with their desired values.

5. CASE STUDIES AND EXPERIMENTAL RESULTS

The shape-based clustering and prediction approach is applied on the following four time series datasets: benchmark data of MGTS, sunspot number time series, laser data time series and Lorenz time series. For evaluating the performance of the proposed paradigm and to compare the results with the previous methods, the error criteria used are mean square error (MSE), root mean square error (RMSE) and normalized mean squared error (NMSE). These are calculated with the help of (16)–(18).

\[
\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (x_1 - \hat{x}_1)^2
\]

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_1 - \hat{x}_1)^2}
\]

\[
\text{NMSE} = \frac{\sum_{i=1}^{N} (x_1 - \hat{x}_1)^2}{\sum_{i=1}^{N} (x_1 - \bar{x}_1)^2}
\]

where \( N \) is the number of data points in the test data. \( x_1, \hat{x}_1 \) denote the \( i^{th} \) observed value; \( \hat{x}_i \) predicted value computed by the proposed models and \( \bar{x}_1 \) denote the average of the observed data. The results of the algorithm are compared with the other results in the literature, and it can be conferred that the given approach provides a better prediction as compared with the previous methods.
5.1. Mackey–Glass time series

The MGTS models the chaotic oscillations in physiological process (Glass and Mackey 1987). This series is a nonlinear time delay differential equation given by (19). Fourth-order Runge–Kutta method was used to find the numerical solution to the aforementioned Mackey–Glass equation. Here, it is assumed that time step is $0.1; x.0/ = 0.1:2$.

\[
\frac{dx(t)}{dt} = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t) \tag{19}
\]

Using (19), 2000 data vectors have been produced by taking values of delay parameter $\tau = 17$ and 30. Of the 2000 data vectors, the first 1000 data vectors, i.e., 50%, of the total data are considered as the training dataset, and the rest 1000 data vectors are used for the validation. The initial data are divided into six clusters with the help of the proposed clustering algorithm, and the test data are fed into the SBCF and SBCN for the prediction by FIS and ANN, respectively. The six state HMM parameters are estimated using the Baum–Welch algorithm.

For $\tau = 17$: Figure 8(a) and (b) shows the comparison of original MGTS with the predicted one with the help of SBCF and SBCN, respectively. The figures also show the error curve in the case of training and test data. Table 7 shows the actual estimated parameters for the HMM and initial parameters for FISs and ANNs. RMSE comparison method of various previous approaches is shown in Table 8.

For $\tau = 30$: Figure 9(a) and (b) shows the comparison of original MGTS with the predicted one with the help of SBCF and SBCN, respectively. The figures also show the error curve in the case of training and test data. RMSE comparison method of various previous approaches is shown in Table 9. The data in the first three rows of the table are taken from Almaraashi et al. (2010).

5.2. Sunspot data time series

The sunspot number is the oldest solar index measuring the solar activity. For a long time, it was the only index representative of the solar cycle, and many studies on
The cyclical behavior of the sun led using the sunspot number. Sunspots can be used to study the climate change because of their effect upon global temperature variations. Sunspots are dark blotches on the sun, and yearly average of sunspots has been recorded since 1700. Of the 1500 data, the first 1000 data vectors, i.e., 67%, of the total data are considered to be the training dataset, and the rest 500 data vectors are used for the validation.

The initial data is divided into eight clusters with the help of the proposed algorithm, and the test data are fed into SBCF and SBCN for the prediction by FIS and ANN, respectively. Figure 10(a) and (b) shows the comparison of original sunspot number with the predicted

<table>
<thead>
<tr>
<th>Mean</th>
<th>Covariance matrix</th>
<th>Transition matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1012</td>
<td>0.7538</td>
<td>0.8506</td>
</tr>
<tr>
<td>6.8142</td>
<td>0.5071</td>
<td>0.1259</td>
</tr>
<tr>
<td>5.8285</td>
<td>0.150</td>
<td>0.0000</td>
</tr>
<tr>
<td>8.1644</td>
<td>1.0778</td>
<td>0.0000</td>
</tr>
<tr>
<td>10.6149</td>
<td>0.3751</td>
<td>0.0000</td>
</tr>
<tr>
<td>11.6903</td>
<td>0.7192</td>
<td>0.1494</td>
</tr>
<tr>
<td>9.6279</td>
<td>0.4500</td>
<td>0.416</td>
</tr>
<tr>
<td>7.3013</td>
<td>0.0776</td>
<td>0.0557</td>
</tr>
<tr>
<td>5.8999</td>
<td>0.6404</td>
<td>0.0926</td>
</tr>
<tr>
<td>7.5227</td>
<td>1.1352</td>
<td>0.0000</td>
</tr>
<tr>
<td>6.9226</td>
<td>1.1400</td>
<td>0.4500</td>
</tr>
<tr>
<td>8.6070</td>
<td>0.6134</td>
<td>0.7615</td>
</tr>
<tr>
<td>10.8485</td>
<td>0.3126</td>
<td>0.7615</td>
</tr>
<tr>
<td>11.6009</td>
<td>0.7666</td>
<td>0.2516</td>
</tr>
<tr>
<td>11.5023</td>
<td>0.3679</td>
<td>0.2516</td>
</tr>
<tr>
<td>11.4524</td>
<td>0.2718</td>
<td>0.4510</td>
</tr>
<tr>
<td>11.8889</td>
<td>0.1969</td>
<td>1.5041</td>
</tr>
<tr>
<td>10.6789</td>
<td>0.1751</td>
<td>1.1831</td>
</tr>
<tr>
<td>8.2030</td>
<td>0.4150</td>
<td>1.2939</td>
</tr>
<tr>
<td>6.2362</td>
<td>0.4482</td>
<td>1.1070</td>
</tr>
<tr>
<td>6.1442</td>
<td>1.1955</td>
<td>1.2338</td>
</tr>
<tr>
<td>6.2247</td>
<td>0.1909</td>
<td>1.3473</td>
</tr>
<tr>
<td>8.8824</td>
<td>0.7101</td>
<td>2.1953</td>
</tr>
<tr>
<td>10.5438</td>
<td>0.4548</td>
<td>0.6719</td>
</tr>
<tr>
<td>11.1247</td>
<td>0.4329</td>
<td>1.0243</td>
</tr>
<tr>
<td>9.8726</td>
<td>0.3779</td>
<td>0.4548</td>
</tr>
<tr>
<td>10.8073</td>
<td>0.1764</td>
<td>0.3010</td>
</tr>
<tr>
<td>11.4794</td>
<td>0.2733</td>
<td>0.4204</td>
</tr>
<tr>
<td>11.3294</td>
<td>0.3010</td>
<td>0.1094</td>
</tr>
<tr>
<td>9.5432</td>
<td>0.5224</td>
<td>0.1494</td>
</tr>
</tbody>
</table>

Number of states = 6
Covariance type = full
FIS parameters
Number of membership functions = 2
Membership function = Gaussian
Tuning = gradient descent method
ANN Parameters
Feed forward back propagation network
Number of input layers = 2
Activation function = tangent sigmoid
TABLE 8. Comparison of Previous Methods (MGTS) ($\tau = 17$).

<table>
<thead>
<tr>
<th>Learning algorithm</th>
<th>Results (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liner predictive model (Kim and Kim 1991)</td>
<td>0.55</td>
</tr>
<tr>
<td>Autoregressive model (Kim and Kim 1991)</td>
<td>0.19</td>
</tr>
<tr>
<td>Wang and Mendel (1992)</td>
<td>0.091</td>
</tr>
<tr>
<td>Cascade correlation $N$ (Kim and Kim 1991)</td>
<td>0.06</td>
</tr>
<tr>
<td>Sixth-order polynomial (Kim and Kim 1991)</td>
<td>0.04</td>
</tr>
<tr>
<td>Kim and Kim (1991)</td>
<td>0.026</td>
</tr>
<tr>
<td>EPNet (Yao and Liu 1997)</td>
<td>0.02</td>
</tr>
<tr>
<td>DENGIS ( offline ) (Song and Kasbov 2002)</td>
<td>0.016</td>
</tr>
<tr>
<td>Data-driven linguistic modeling using relational fuzzy rules (Zurada and Gaweda 2003)</td>
<td>0.009</td>
</tr>
<tr>
<td>ANFIS (Jang 1993)</td>
<td>0.007</td>
</tr>
<tr>
<td>GEFREX (Russo 2003)</td>
<td>0.0061</td>
</tr>
<tr>
<td>Hassan et al. (2006)</td>
<td>0.0055</td>
</tr>
<tr>
<td>SBCN (1000 values)</td>
<td>0.00092</td>
</tr>
<tr>
<td>SBCF (1000 values)</td>
<td>0.00071</td>
</tr>
</tbody>
</table>

Comparison with previous approaches is shown in Table 10. The data of the first nine rows in the table are taken from Ardalani-Farsa and Zolfaghari (2010).

5.3. Laser data time series

Laser data time series prediction problem is widely used to test the accuracy of various prediction models. Here, the dataset of 900 data vectors is taken. Of these, the first 500 data vectors, i.e., 56%, of the total data are considered to be the training dataset, and the rest 400 data vectors are used for the validation. The values of embedding dimension and time delay are taken as 4 and 2, respectively. The initial data are divided into six clusters with the help of the proposed algorithm, and the test data are fed into SBCF and SBCN for the prediction.

![Figure 9](image_url) (a) SBCF and (b) SBCN prediction of MGTS ($\tau = 30$).
TABLE 9. Comparison of Previous Methods ($\tau = 30$).

<table>
<thead>
<tr>
<th>Learning algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back propagation with fuzzy system (Mendel and John 2002)</td>
</tr>
<tr>
<td>One pass method with fuzzy system (Mendel and John 2002)</td>
</tr>
<tr>
<td>Gradient descent with fuzzy system (Musikasuwan et al. 2004)</td>
</tr>
<tr>
<td>Fuzzy system tuned with simulated annealing (Almarashi et al. 2010)</td>
</tr>
<tr>
<td>SBCN</td>
</tr>
<tr>
<td>SBCF</td>
</tr>
</tbody>
</table>

FIGURE 10. (a) SBCF and (b) SBCN prediction of Sunspot Data (SSD) Time series.

TABLE 10. Comparison of Previous Methods.

<table>
<thead>
<tr>
<th>Learning algorithm</th>
<th>Prediction error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
</tr>
<tr>
<td>WP-MLP (Teo et al. 2001)</td>
<td>1.25E–01</td>
</tr>
<tr>
<td>McNISH-Linc. (McNish and Lincoln 1949)</td>
<td>8.00E–02</td>
</tr>
<tr>
<td>Nonlinear method (Sello 2001)</td>
<td>3.40E–01</td>
</tr>
<tr>
<td>Sello (2001)</td>
<td>5.60E–01</td>
</tr>
<tr>
<td>Denkmayr (1997)</td>
<td>1.85</td>
</tr>
<tr>
<td>RBF-OLS (Gholipour et al. 2006)</td>
<td>4.60E–02</td>
</tr>
<tr>
<td>LLNF-LoLiMot (Gholipour et al. 2006)</td>
<td>3.20E–02</td>
</tr>
<tr>
<td>ERNN (Ma et al. 2007)</td>
<td>1.29E–02</td>
</tr>
<tr>
<td>MLP (Koskela et al. 1996)</td>
<td>9.79E–02</td>
</tr>
<tr>
<td>Elman–NARX (Ardalani-Farsa and Zolfaghari 2010)</td>
<td>1.40E–04</td>
</tr>
<tr>
<td>SBCF</td>
<td>1.4450E–04</td>
</tr>
<tr>
<td>SBCN</td>
<td>1.3970E–04</td>
</tr>
</tbody>
</table>
5.4. Lorenz data time series

The Lorenz attractor time series is generated with the help of following equations:

\[
\frac{dx}{dt} = -\sigma x + \sigma y \\
\frac{dy}{dt} = Rx - y - xz \\
\frac{dz}{dt} = -bz + xy
\]

Time series data have been generated by using a standard Runga–Kutta method with a step size of 0.01. The series is generated by taking $\sigma = 10$, $R = 28$, and $b = 8/3$. The data of length 1500 are produced with the help of the aforementioned equations. Of the 1500, we have taken a dataset of 1000 data vectors, i.e., 66%, of the total data for training and the rest 500 for the validation. The values of embedding dimension and time delay are taken as 3 and 16, respectively. The initial data are divided into six clusters with the help of the proposed algorithm, and the test data are fed into SBCF and SBCN for the prediction by FIS and ANN, respectively. Figure 12(a) and (b) shows the comparison of original time series with the predicted one. RMSE comparison with previous approaches is shown in Table 12.
6. CONCLUSION

In this article, we have proposed a novel time series prediction model that is based upon shape-based clusters. Here, an unsupervised shape-based clustering algorithm is proposed, which involves CDHMM with Pearson model to perform clustering of input data on the
basis of their similarities in shape. These shape-based clusters are further processed by FISs/ANNs, resulting in well-trained clusters.

As a first step, the time series is converted into phase space. The conversion of time series into phase space is based upon Takens’ embedding theorem. After converting the time series into phase space, the data are divided into clusters according to their pattern similarity with the help of shape-based clustering algorithm. Once the data are arranged into clusters based on their shape, the next step involves the training of FISs or ANNs to predict the results. For an FIS/ANN, it is now easier to analyze the data after the shape-based clustering. When the test sequence arrives, the first step is to analyze that to which cluster this sequence belongs. To allocate the sequence to a proper cluster, LL ratio test is used. Once the test sequence is assigned a cluster, the output is predicted with the help of well-trained clusters. The output has been predicted with a high degree of accuracy, and it is clearly visible from the results that both the proposed methods outperform other past approaches for the prediction of the mentioned time series.

There is a scope for future work as the process of shape-based clustering lays a firm foundation for other methods to work upon. In the case of biological sequences that do not have continuous values, but rather have symbols, DHMMs can be used for both clustering and prediction.

REFERENCES


**APPENDIX A: SHAPE-BASED CLUSTERS OF VARIOUS TIME SERIES**

![Figure A1. Sunspot data time series.](image-url)
**Figure A2.** Lorenz time series.

**Figure A3.** Henon time series.
FIGURE A4. Laser time series.