Decision Support in Time Series Modeling by Pattern Recognition

Kun Chang LEE and Sung Joo PARK

Department of Management Science, Korea Advanced Institute of Science and Technology, Cheongryang, Seoul, Korea

This research is aimed at presenting a new, pattern recognition-based DSS scheme for the time series model identification. The scheme is based on two principles: pattern matching and inductive learning. Pattern matching is used to classify a pattern of the time series into one of the autoregressive moving-average models. The pattern is obtained from the extended sample autocorrelations of the time series. Inductive learning is used to enhance the capability of recognizing input patterns, and linear discriminants are used to discriminate one pattern from the others. To implement the idea, a decision support system named DSSTSM was designed and a prototype was developed on the microcomputer. Experimental results show that the combination of the pattern recognition principles with a DSS can yield a promising solution to the time series modeling.

Keywords: Decision Support System (DSS), Time Series Modeling, Pattern Recognition, Features, Linear Discriminant, Pattern Matching, Inductive Learning, Classification, Autoregressive Moving-Average (ARMA) Model, Extended Sample Autocorrelation Function (ESACF).

Kun Chang Lee holds a M.S. from Korea Advanced Institute of Science and Technology (KAIST) and is currently a doctoral candidate in management information systems at KAIST. His current research interests include DSS design and development for managerial forecasting system, expert system-aided decision making, applications of artificial intelligence to strategic decision makings and information systems management.

Sung Joo Park is an Associate Professor of Management Science at Korea Advanced Institute of Science & Technology. He received a Ph.D. in System Science at the Michigan State University. His current research interests include computer-aided software engineering methodologies and tools, simulation and modeling, knowledge based simulation, decision support systems, and integration of software engineering and management science techniques. He is editor of the Korean Management Science Review.

1. Introduction

In the field of management the need for planning is great because the lead time for decision making ranges from several years (for the case of capital investments) to a few days or even a few hours (for transportation or production schedules). Usually, a time series is used for the management planning. In this sense, determining a proper model for the time series is very important for effective and efficient planning.

The time series model that has been widely used is an autoregressive moving-average (ARMA) model, which has been proposed by Box and Jenkins [3] and generally advocated by both scholars and practitioners [20]. An ARMA \((p, q)\) process \(Z_t\) is generated by the following relation:

\[
Z_t = \phi_1 Z_{t-1} + \cdots + \phi_p Z_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q},
\]

where \(\varepsilon_t\) represents normally distributed, mutually independent, random variables with 0 mean and variance \(\sigma^2\), and \(\phi\) and \(\theta\) are parameters to be estimated. In view of the ARMA model, the time series modeling means the determination of both autoregressive (AR) order \(p\) and moving average (MA) order \(q\). For this purpose, many approaches have been developed in the field of statistics. To review the literature on this issue, see [1,2,3,4,5,6,11,12,17,18]. Among the approaches, the Box–Jenkins method has emerged as one of the most popular approaches because of its theoretical elaborateness and accuracy for short-term forecasting. However, the approach is not satisfactory in many cases for the following reasons:

1. It requires a great amount of time to build a satisfactory model.
2. The process of finding an appropriate model is very difficult for the general user to fully understand.

To overcome these difficulties, this study combines two pattern recognition principles with a
decision support system (DSS): pattern matching and inductive learning. The pattern matching implies that once a prototype pattern is specified, the proper model for a time series is an ARMA model whose prototype pattern most closely matches that of the time series. In this study, the pattern of a time series is specifically obtained from the extended sample autocorrelation function (ESACF) [17]. The inductive learning is used to recognize the patterns and is accomplished by both pattern base and linear discriminants. For a full review of the pattern recognition, please refer to [15]. In this research, we focus on the following problems:

1. To avoid a complicated approach to the time series modeling, the pattern matching concept is used.
2. To enhance the capability of recognizing various types of the patterns, the inductive learning technique is used in connection with a pattern base.
3. To build a framework for resolving the time series modeling by the pattern recognition approach, a DSS is designed and developed.

The prototype system is named DSSTSM – Decision Support System for Time Series Modeling. Throughout the paper, the system means the DSSTSM. The DSS is well described in [13] and medical application of pattern recognition to monitory waveforms is discussed in [19].

The structure of the remaining sections is as follows: The pattern recognition approach to the time series modeling is discussed in section 2. The learning method and its verification are described and some issues related to the role of the pattern base are shown in section 3. In section 4, the solution process of the DSSTSM is presented and the experimental results are illustrated.

2. Pattern Recognition Approach

2.1. Fundamentals

Pattern recognition can be treated as the categorization of input data into identifiable classes via the extraction of significant features of the data. A pattern class is a category determined by some common attributes. A pattern is the description of any member of a category representing a pattern class. Mathematically, a pattern is a vector and is composed of a set of features values or features [15]. The features of a pattern class are the characterizing attributes common to all the patterns belonging to that class. The important problem in the pattern recognition is to obtain features characterizing the different patterns so that they can be classified correctly. In this sense, the pattern recognition approach consists of two stages: feature extraction and classification. The feature extraction is to extract features specific to the input data and the classification is to assign an unknown pattern to one of the pattern classes.

2.2. Pattern by ESACF

The ESACF approach [17] begins with the assumption that the data $Z_t$ are generated by an ARMA $(p, q)$ process, where $p$ and $q$ are unknown. For $p = k$, the value of the $k$th ESACF at lag $j$ is defined as the sample autocorrelation of an estimate of the moving average portion of the ARMA $(k, q)$ process under the assumption $q = j$. For each $j$ (i.e., each assumed value of $q$), the $k$ autoregressive parameters are estimated by an algorithm that uses ordinary least squares (OLS) estimates from autoregressive regressions of orders $k$ through $k + j$ on $Z_t$. The algorithm yields consistent estimates of the true autoregressive parameters for $j > q$ and $p - k$. It follows that the $k$th ESACF of an ARMA$(k, q)$ process has the same cutting-off property as the sample autocorrelation function (SACF) of a pure MA$(q)$ process. Each element of the $k$th ESACF, however, is derived from a different set of estimates of the autoregressive parameters. In practice, the ESACF approach can be characterized by arranging the set of extended sample autocorrelations corresponding to various values of $p$ and $q$ in a two-dimensional table. The extended sample autocorrelations in each row correspond to a fixed value of $p$, and those in each column correspond to a fixed value of $q$. It is worth to note that for $k = p$, the ESACF cuts off for $j > q$. Thus if the rows of the table of the ESACF are numbered 0, 1, 2, ..., to specify the order of AR process and the columns are similarly numbered to specify the order of MA process, then the ESACF within two standard deviations will form a triangle with boundaries given by the lines $k = p$ and $j - k = q$. The row and column coordinates of the vertex of the trian-
Table 1
Triangular Pattern by ESACF Approach.

<table>
<thead>
<tr>
<th>AR</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

gle correspond to tentative estimates of the AR order p and MA order q, respectively. One advantage of the ESACF relative to the popular Box–Jenkins approach is its ability to identify mixed ARMA processes while the Box–Jenkins method is especially helpful in identifying pure AR or pure MA model. Furthermore, the ESACF approach does not require that the series be stationary. It, therefore, eliminates the need to specify the order of differencing and precludes problems associated with overdifferencing.

In this paper, a pattern class represents one of the ARMA models. A pattern is a vector whose elements are composed of binary numbers obtained by the ESACF values of the time series. The binary numbers are assigned to 1 when the ESACF value lies between two standard deviations and otherwise assigned to 0. A triangular pattern by the ESACF approach is shown in table 1, where the prototype pattern of ARMA (2, 2) model is shown. The underlined numbers represent the boundary of the triangular pattern. It is noted that in table 1 there exists only one triangular pattern. However, in practice, such an ideal case rarely occurs. Specifically, a time series with seasonality often yields many triangular patterns. In this case, each of the triangular patterns might represent true values of both AR and MA orders, causing confusion in the correct identification of a time series model. This difficulty can be overcome by the use of the pattern matching concept.

2.3. Formulation by Pattern Matching

Levine [7] has shown that it is impossible to have a general theory for feature extraction. It is also shown that the design of a feature extractor is often heuristic, employing many ad hoc strategies. In this sense, this paper adopts the ESACF approach as a method of extracting features because it can yield particular values about a time series of interest. The time series modeling can be newly posed as three stages of the pattern recognition:

1. Preparation of the prototype patterns for a number of ARMA models.
2. Extraction of a pattern specific to the time series.
3. Classification into an ARMA model showing a pattern which best matches that of the time series.

Stage 1 dictates that various patterns should be learned before the classification. Stages 2 and 3 imply feature extraction and classification, respectively. Fig. 1 shows a schematic diagram of the pattern recognition approach to the time series modeling.
3. Learning

3.1. Fundamentals

Learning is a many-faceted phenomenon. The learning process includes the acquisition of new declarative knowledge, the development of cognitive skills through instruction or practice, the organization and generalization of new knowledge, and the discovery of new facts and theories through observation and experimentation. The learning of this paper belongs to an inductive learning in the sense that new knowledges about the patterns are discovered through observations [8]. Specifically, it is done through adaptation of parameters. One of the similar approaches is a perceptron [9], in which learning is realized by adjusting coefficients in the case of misclassification.

This research uses a linear discriminant [14] to discriminate one pattern from the others. Once a function type has been specified, the remaining problem is how to determine the coefficients. Using the learning algorithm presented, the coefficients are obtained from the training samples (or patterns). The role of the training samples is to provide knowledges about the patterns. In order for the algorithm to converge to the solution coefficients, a linear separability between the pattern classes should be guaranteed. The linear separability implies that each of the pattern classes can be separated by a linear discriminant [15]. In this research, each of the patterns denotes one of the ARMA models, so that a linear separability between the patterns, instead of the pattern classes, should be assured so as to use the algorithm. In view of the fact that the patterns being considered are different from each other, the linear separability is certainly retained. It follows that the algorithm can result in the solution coefficients of the linear discriminants separating the training patterns.

3.2. Derivation of Learning Algorithm

The algorithm can be easily derived through the well-known gradient concept [14]. Basically, the gradient scheme provides a tool for finding the minimum (or maximum) of a function. Assume that functions have a unique minimum value. Let the criterion function be

$$J(w, x) = \frac{1}{2}( |wx| - wx ),$$

where $|wx|$ is the absolute value of $wx$. $w$ and $x$ are a weight vector and a pattern vector, respectively. The partial derivative of $J$ with respect to $w$ is given by

$$\frac{\partial J}{\partial w} = \frac{1}{2}[x \text{sgn}(wx) - x],$$

where, by definition,

$$\text{sgn}(wx) = 1 \text{ if } wx \geq 0$$

$$= -1 \text{ if } wx < 0.$$  \hspace{1cm} (3)

It is clear that the minimum of function (1) is 0 and that this minimum results when $wx \geq 0$. If we let $w^{(k)}$ represent the value of $w$ at $k$th step, an iterative descent algorithm for minimizing function (1) may be written as

$$w^{(k+1)} = w^{(k)} - c \left\{ \frac{\partial J(w^k, x^k)}{\partial w} \right\}_{w = w^{(k)}},$$

where $w^{(k+1)}$ represents the new value of $w$ at $(k + 1)$th step, and a constant $c > 0$ dictates the magnitude of the correlation, that is, a correction increment. It is noted that no corrections are made on $w$ when $(\partial J/\partial w) = 0$, which is the condition for a minimum. Substituting eq. (2) into eq. (4) yields

$$w^{(k+1)} = w^{(k)} + c \left\{ x^k - w^{(k)} \text{sgn}[w^{(k)}x^k] \right\},$$

where $x^k$ represents the training pattern being considered at the $k$th iterative step. Substituting eq. (3) into eq. (5) results in the algorithm

$$w^{(k+1)} = w^{(k)}$$

$$= w^{(k)} + cx^k$$

$$= w^{(k)} + cx^k \text{ if } w^{(k)}x^k \geq 0,$$

where $c > 0$ and $w^{(1)}$ is arbitrary. Expression (6) indicates that the iterative descent scheme for $w$ can be derived so that the criterion function (1) is minimized eventually. Simply stated, the algorithm makes a change in $w$ if and only if the pattern being considered at the $k$th training step is misclassified. In the case of correct classification, no corrections are made on $w$.

3.3. Decision Rule

For convenience, suppose that there are two ARMA classes $C_1$ and $C_2$ such that $x_1, \ldots, x_m$ are training patterns from $C_1$ and $C_2$. A pattern $x_i$ is supposed to be composed of $n$ features. We want to obtain a linear discriminant separating $C_1$ and
C. For a pattern \( x_i \) of unknown classification, the decision rule is

if \( w x_i + w_{n+1} \geq 0 \), then \( x_i \in C_1 \), and

if \( w x_i + w_{n+1} < 0 \), then \( x_i \in C_2 \).

where \( w = (w_1, w_2, \ldots, w_n) \) and \( x_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \). Consider now augmented samples of \((n + 1)\) vectors. Let \( W = (w, w_{n+1}) \) and \( X_i = (x_i, 1) \).

Using this augmented notation the decision rule becomes

if \( WX_i \geq 0 \), then \( X_i \in C_1 \),

if \( WX_i < 0 \), then \( X_i \in C_2 \).

Arbitrarily multiplying the patterns of \( C_2 \) by \(-1\), we may express the algorithm in a form equivalent to expression (6):

\[
W^{(k+1)} = W^{(k)} \text{ if } W^{(k)} X_i^{(k)} \geq 0
\]

\[
= W^{(k)} + c X_i^{(k)} \text{ if } W^{(k)} X_i^{(k)} < 0. \tag{7}
\]

The algorithm (7) can be justified heuristically; multiplying both sides by \( X_i^{(k)} \) results in

\[
W^{(k+1)} X_i^{(k)} = W^{(k)} X_i^{(k)} + c X_i^{(k)} X_i^{(k)} \text{ if } W^{(k)} X_i^{(k)} < 0.
\]

The term \( c X_i^{(k)} X_i^{(k)} \) is clearly positive, resulting in \( W^{(k+1)} X_i^{(k)} > W^{(k)} X_i^{(k)} \). This implies that the algorithm always improves toward a solution through iterations.

3.4. Convergence

Convergence means that under the condition of linear separability the algorithm (7) will terminate after a finite number of iterations yielding the solution coefficients (or weights). We present a convergence proof for the case of two pattern classes, which is easily extended to the \( N \)-class problem [10]. Let \( x_1, \ldots, x_m \) represent \( m \) training patterns belonging to two classes, where the patterns of class \( C_2 \) have been multiplied by \(-1\) so as to simplify the classification procedure. It is stipulated that the algorithm of eq. (6) yields a solution weight vector \( w^* \) with the property

\[
w^* x_i \geq 0, \quad i = 1, 2, \ldots, m. \tag{8}
\]

For simplicity, it is assumed that \( c = 1 \). Consider the \( k \) th step in which a correction occurs. Then

\[
w^{(k+1)} = w^{(k)} + x_i^{(k)}, \quad \text{and} \tag{9}
\]

\[
w^{(k)} x_i^{(k)} < 0. \tag{10}
\]

From eq. (9),

\[
w^{(k+1)} = w^{(k)} + x_i^{(k)} + x_i^{(2)} + \ldots + x_i^{(k)} \tag{11}
\]

Taking the inner product of \( w^* \) with both sides of eq. (11) yields

\[
w^* w^{(k+1)} = w^* w^{(k)} + w^* x_i^{(1)} + \ldots + w^* x_i^{(k)} \tag{12}
\]

Using eq. (8),

\[
w^* w^{(k+1)} \geq w^* w^{(k)} + k a, \tag{13}
\]

where \( a = \min_i [w^* x_i^{(j)}], \quad j = 1, \ldots, k \). Since \( w^* \) is a solution vector, \( a \) is greater than or equal to 0 by expression (8). Using the Cauchy–Schwartz inequality,

\[
[w^* w^{(k+1)}]^2 \leq \| w^* \|^2 \| w^{(k+1)} \|^2. \tag{14}
\]

Expression (14) may be written equivalently in the form

\[
\| w^{(k+1)} \|^2 \geq \frac{[w^* w^{(k+1)}]^2}{\| w^* \|^2}. \tag{15}
\]

Substituting expression (13) into (15) yields

\[
\| w^{(k+1)} \|^2 \geq \frac{[w^* w^{(k+1)} + k a]^2}{\| w^* \|^2}. \tag{16}
\]

Inequality (16) states that the squared length of the weight vector must grow quadratically at least with the number of iterations \( k \). On the other hand, from eq. (9),

\[
\| w^{(j+1)} \|^2 = \| w^{(j)} \|^2 + 2 w^{(j)} x_i^{(j)} + \| x_i^{(j)} \|^2. \tag{17}
\]

Equivalently,

\[
\| w^{(j+1)} \|^2 - \| w^{(j)} \|^2 = 2 w^{(j)} x_i^{(j)} + \| x_i^{(j)} \|^2. \tag{17}
\]

Using expression (10) and letting \( Q = \max \| x_i^{(j)} \|^2 \) results in

\[
\| w^{(j+1)} \|^2 - \| w^{(j)} \|^2 \leq \| x_i^{(j)} \|^2 \leq Q. \tag{18}
\]

Adding these inequalities for \( j = 1, 2, \ldots, k \) yields the inequality

\[
\| w^{(k+1)} \|^2 \leq \| w^{(1)} \|^2 + Q k. \tag{19}
\]

Expression (19) states that the squared length of the weight vector can grow no faster than linearly with the number of steps \( k \). Clearly, the bounds given by inequalities (16) and (19) conflicts with each other such that for sufficiently large values of \( k \) there exists an iterative step \( k_m \) at which the
lower bound in (16) exceeds the upper bound in (19). This is clearly a contradiction. Hence, \( k \) can be no larger than \( k_m \), which is a solution to the equation

\[
\frac{\left[ w^* w^{(1)} + k_m a \right]^2}{\| w^* \|^2} = \| w^{(1)} \|^2 + Q k_m. \tag{20}
\]

This fact indicates that the algorithm must terminate after at most \( k_m \) steps if a solution vector \( w^* \) exists. Thus the proof is completed.

3.5. Correction Increment

Three rules for choosing a correction increment \( c \) are:

(1) Fixed-increment rule: \( c \) is any fixed positive number.

(2) Absolute correction rule: \( c \) is taken to be the smallest integer that will make the value of \( wx \) cross the threshold of zero. That is, \( c = \) smallest integer greater than \( |wx| / xx \).

(3) Fractional correction rule: \( c \) is chosen such that \( c = h(\|wx\|/xx) \), where \( h \) is determined such that \( \|w^{(i+1)} x^{(i)} - w^{(i)} x^{(i)}\| = h \|w^{(i)} x^{(i)}\|, \quad 0 < h \leq 2 \).

The convergence for these three error-correction rules can be proved. For a detailed proof, refer to [10]. For simplicity we choose the first rule as a correction procedure. Therefore, the assumption that \( c = 1 \) loses no generality.

3.6. Pattern Base

The principal role of the pattern base is to store information about the learned patterns: linear discriminants and classified models. During the classification, the unknown pattern is either recognized or not. In the case of misclassification, the pattern is to undergo the learning process and corresponding results are stored in the pattern base. In this way, knowledges about the misclassified patterns are accumulated. As a result, it is more likely that a correct model for the unknown pattern can be derived inductively. In view of the relational data base, the pattern base consists of \( n + 1 \) attributes representing the weights as well as one attribute denoting the classified model. Also it contains a number of tuples indicating the number of the learned patterns. Table 2 shows the structure of the pattern base.

<table>
<thead>
<tr>
<th>( w_1 )</th>
<th>( w_2 )</th>
<th>( w_3 )</th>
<th>\cdots</th>
<th>( w_n )</th>
<th>( w_{n+1} )</th>
<th>classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>\cdots</td>
<td>1</td>
<td>1</td>
<td>ARMA(( p_1, q_1 ))</td>
</tr>
</tbody>
</table>

4. Decision Support System

This section shows the architecture and the solution process of the DSSTSM. Also presented is the strategy to classify the patterns from each other. Finally, it is illustrated that the performance of the DSSTSM is well enhanced through a number of learning.

4.1. System Architecture

The major component of the DSSTSM is a menu-driven solving system, which consists of four subsystems: feature extractor, learning, classification and why. The role of the feature extractor is to extract features from the time series and generate the corresponding pattern. The learning subsystem performs the function of recognizing the prototype patterns and the misclassified patterns. In the classification subsystem, the input pattern is classified into one of the ARMA models. The why subsystem shows the brief process of deriving a model for the unknown pattern by using the information in the pattern base. The data base contains a set of time series under consideration. The prototype of the DSSTSM was implemented on the IBM-PC/AT using Pascal. Fig. 2 depicts the architecture of the DSSTSM.

4.2. Classification Strategy

Let \( d_i \) denote a decision value when a pattern \( x \) belongs to class \( i \). Given an input pattern, the decision value is defined as a numeric value yielded by a linear discriminant in the pattern base regarding for the input pattern. In the classification stage, the decision value is used as a criterion for selecting the model. If there are two patterns in the pattern base, only one linear discriminant is
needed to separate the patterns, but for the case of multiple patterns it is necessary to find the linear discriminants that classify one pattern from another. There are three strategies of classifying the multiple patterns [16]:

1. Consider one pattern as class 1 and all the other patterns as class 2. Then find a linear discriminant that separates these two classes. For \( N \) classes, \( N \) discriminants are needed to classify. In this strategy, for a pattern of class 1, \( d_1 > 0 \) and all the remaining patterns have a negative decision value.

2. Each pattern is separable from every other pattern by a linear discriminant. In other words, the patterns are pairwise separable. In this case, if a pattern belongs to class \( i \), \( d_{ij} > 0 \) for all \( j \neq i \).

3. Combine two strategies above to eliminate cases in which class assignment is impossible. If a pattern belongs to class \( i \), then \( d_i > d_j \) for all \( j \neq i \).

The third strategy is most flexible and general because it has no unclassifiable regions. Therefore, we adopt the strategy as the rule of finding the linear discriminants that classify the patterns.

4.3. Solution Process

The solution process of the DSSTSM as depicted in fig. 3 can be described as follows:

(1) First, extract a pattern from the input series by computing the binary numbers which are obtained from the extended sample autocorrelations of the series.

(2) The system is trained for the training patterns, that is, the patterns of a number of ARMA models. The training process terminates when all the patterns are correctly classified.

(3) After the training, the linear discriminants are obtained and stored in the pattern base. Consequently, the system has the capability to recognize a variety of the input patterns.

(4) For the input pattern, the decision values are computed for each of the linear discriminants stored in the pattern base.

(5) The best model is an ARMA model associated with the linear discriminant showing the maximum decision value.

(6) If misclassified, the pattern undergoes the learning process. Then, a new knowledge about unknown pattern is added to the pattern base. Repeat steps 3 through 5 until the best model is found. If found, stop.

4.4. Illustration

The user is supposed to know the correct model of the input series until the system gains a satisfactory performance via learning. This fact indicates that after the learning, the system is expected to have reached the state of identifying a wide variety of patterns satisfactorily.

4.4.1. Illustrative Data

For illustration, consider the caffeine data analyzed in [5]. The data consist of 178 observations of caffeine levels in instant coffee taken every weekday. Thus a seasonality is expected. The reason why a seasonal series is chosen as the illustrative data is that we want to show how the system resolves the problem of seasonality, which frequently causes some distorted patterns making it difficult to correctly recognize. This illustrative data have been known to follow ARMA(2, 5) model [5].

4.4.2. Simulation Steps

For clarity, the simulation steps are summarized as three steps: training, extraction of features and pattern matching.

Step 1. Training. The DSSTSM was trained to classify various patterns. The training patterns consist of 90 patterns. Maximum order of AR and MA part is specified to 4 and 5, respectively. After the training, all the linear discriminants obtained
are stored in pattern base. Now, it is ready to classify the unknown patterns.

**Step 2. Extraction of Features.** The features and the pattern of the illustrative data are shown in table 3.

Table 3: Features and Pattern of the Illustrative Data.

<table>
<thead>
<tr>
<th>AR</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 2 3 4 5 6 7 8 9</td>
</tr>
<tr>
<td>(a) Features</td>
<td>0.89 0.73 0.58 0.48 0.42 0.46 0.49 0.48 0.42 0.34</td>
</tr>
<tr>
<td>0</td>
<td>0.29 -0.14 -0.14 -0.43 0.07 0.25 0.25 0.12 0.01</td>
</tr>
<tr>
<td>1</td>
<td>0.13 0.08 -0.11 0.03 -0.42 0.15 0.12 0.13 0.07 0.01</td>
</tr>
<tr>
<td>2</td>
<td>-0.27 0.25 -0.07 -0.01 -0.48 0.33 -0.08 0.08 -0.02 0.08</td>
</tr>
<tr>
<td>3</td>
<td>-0.37 -0.20 -0.02 0.04 -0.51 0.30 0.31 0.01 -0.05 0.08</td>
</tr>
<tr>
<td>4</td>
<td>-0.10 -0.19 -0.02 0.40 -0.45 0.33 0.06 -0.05 -0.18 -0.02</td>
</tr>
<tr>
<td>5</td>
<td>0.42 -0.33 -0.23 -0.29 -0.21 -0.08 0.00 -0.08 -0.17 -0.01</td>
</tr>
<tr>
<td>6</td>
<td>-0.35 -0.06 -0.15 -0.21 -0.32 -0.10 -0.03 -0.01 -0.19 -0.24</td>
</tr>
<tr>
<td>7</td>
<td>-0.14 -0.09 0.00 0.24 -0.16 -0.09 0.00 -0.01 -0.14 -0.13</td>
</tr>
<tr>
<td>8</td>
<td>-0.13 -0.11 0.00 0.28 -0.39 0.01 -0.07 0.02 -0.12 -0.17</td>
</tr>
<tr>
<td>(b) Pattern</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>1 1 1 1 0 1 0 0 1 1</td>
</tr>
<tr>
<td>1</td>
<td>1 1 1 1 0 1 1 1 1 1</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1 1 0 0 1 1 1 1</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1 1 0 0 0 1 1 1</td>
</tr>
<tr>
<td>1</td>
<td>0 0 1 1 0 0 1 1 0 1</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 1 1 1 0 1</td>
</tr>
<tr>
<td>0</td>
<td>1 1 1 0 0 1 1 1 1 1</td>
</tr>
<tr>
<td>1</td>
<td>1 1 1 0 0 1 1 1 1 1</td>
</tr>
<tr>
<td>1</td>
<td>1 1 1 0 0 1 1 1 1 0</td>
</tr>
</tbody>
</table>
Table 4.
Results of Simulation Experiments.

<table>
<thead>
<tr>
<th>Series</th>
<th>Classified Model</th>
<th># of Learning</th>
<th>Decision Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical Process</td>
<td>ARMA(1, 1)</td>
<td>2</td>
<td>-3324</td>
</tr>
<tr>
<td>Chemical Process</td>
<td>ARMA(2, 0)</td>
<td>2</td>
<td>-3320</td>
</tr>
<tr>
<td>Simulated with</td>
<td>ARMA(4, 1)</td>
<td>3</td>
<td>-3290</td>
</tr>
<tr>
<td>U.S. GNP*</td>
<td>ARMA(2, 0)</td>
<td>2</td>
<td>-3727</td>
</tr>
<tr>
<td>IBM Stock Prices*</td>
<td>ARMA(1, 1)</td>
<td>3</td>
<td>-3810</td>
</tr>
<tr>
<td>University of Wisconsin Hospital In-Patient Census*</td>
<td>ARMA(2, 3)</td>
<td>1</td>
<td>-2681</td>
</tr>
<tr>
<td>U.S. Consumer Price Index*</td>
<td>ARMA(4, 3)</td>
<td>2</td>
<td>-3345</td>
</tr>
<tr>
<td>U.S. Expenditures on Producers' Durables*</td>
<td>AR(1)</td>
<td>2</td>
<td>-3123</td>
</tr>
<tr>
<td>Champagne Sales*</td>
<td>MA(1)</td>
<td>1</td>
<td>-3021</td>
</tr>
<tr>
<td>U.S. Investment by Large Commercial Banks in New York*</td>
<td>ARMA(4, 3)</td>
<td>3</td>
<td>-3481</td>
</tr>
</tbody>
</table>


However, experiments with another set of data which are mainly from socio-economic sources showed that after a number of learning the system was able to correctly recognize the input patterns, which are summarized in table 4.

5. Concluding Remarks

In this paper, two pattern recognition principles were used to devise a new approach to time series modeling. Pattern matching was used to classify the input patterns easily and the inductive learning was applied to recognize the patterns gradually. Although the learning procedure is simple and heuristic, it has a great potential for developing the learning DSS in view of its convergent property and its ability to handle multi-classes. The major contributions of this paper are: (1) demonstration of the applicability of the pattern matching concept to the time series modeling, (2) verification of the inductive learning procedure useful for the development of the learning DSS, and (3) reduction of the complexities involved in the time series modeling via a DSS approach. The experimental results assured that the pattern recognition principles presented are useful for both developing the trainable DSS and reducing the difficulties of the time series modeling.

References