Identifying Distinctive Subsequences in Multivariate Time Series by Clustering

Tim Oates
Computer Science Department, LGRC
University of Massachusetts, Box 34610
Amherst, MA 01003-4610
oates@cs.umass.edu
(413) 577-0669

Abstract

Most time series comparison algorithms attempt to discover what the members of a set of time series have in common. We investigate a different problem, determining what distinguishes time series in that set from other time series obtained from the same source. In both cases the goal is to identify shared patterns, though in the latter case those patterns must be distinctive as well. An efficient incremental algorithm for identifying distinctive subsequences in multivariate, real-valued time series is described and evaluated with data from two very different sources: the response of a set of bandpass filters to human speech and the sensors of a mobile robot.

Reference Number: 357

1 Introduction

Given two or more sequences of discrete tokens, a dynamic programming algorithm exists for finding the longest common subsequence they share (Cormen, Leiserson, & Rivest 1990). This basic algorithm has been adapted in various ways to find patterns shared by real-valued time series as well (Kruskall & Sankoff 1983). Unfortunately, the time and space complexity of these algorithms is exponential in the number of sequences. This paper demonstrates that an answer to a slightly different question concerning sequences can be obtained in time and space that are approximately linear in the total length of the sequences. Although the discussion focuses on multivariate, real-valued time series, the approach generalizes trivially to categorical sequences.[1]

Rather than determining what makes a set of sequences similar, we are interested in determining what makes them different from other sequences obtained from the same source. The following example makes this distinction clear. Consider a set of time series, each of which contains the values recorded by the black box of a different airplane that crashed. A longest common subsequence algorithm is likely to find that there are large portions of the sequences that are strikingly similar, including takeoff, the ascent to cruising altitude, and some amount of time spent flying at that altitude. What one really wants to know is whether there is some pattern shared by these time series that does not occur during successful flights. Such patterns can support both prognostic and diagnostic functions, allowing prediction of trouble when they match data obtained during a flight, and focusing analysis to determine probable causes of failures.

We call this process finding *distinctive* subsequences because patterns identified in this manner serve to distinguish the time series under consideration from other time series generated by the same source. To be somewhat more precise, let $\mathcal{R}$ be a data source that produces a vector of q real numbers on each time step. The value of q might be the number of sensors on an earth observing satellite, or the number of stocks whose closing prices are recorded to create a financial time series. Let $\mathcal{P}$ be a pattern, a specification of how the values pro-

[1] The remainder of the paper will use the terms sequence, series and time series interchangeably. In each case, the term means multivariate time series of real-valued data. Exceptions to this convention will be clearly identified as such.

duced by $\mathcal{R}$ are expected to change over some interval of time. Regardless of whether one is interested in distinctive or common subsequences, time series must be gathered for analysis. Suppose the series are obtained individually in response to the occurrence or non-occurrence of an event $\mathcal{E}$, such as a plane crash. Let $p(\mathcal{P}|\mathcal{R})$ be the probability of the pattern occurring in a sequence obtained from the source. A pattern is said to be distinctive if $p(\mathcal{P}|\mathcal{R} \wedge \mathcal{E})$ is significantly different from $p(\mathcal{P}|\mathcal{R} \wedge \overline{\mathcal{E}})$. $\mathcal{P}$ occurs in some number of the time series gathered in response to $\mathcal{E}$, although we do not know where, or even what $\mathcal{P}$ looks like. The goal is to identify $\mathcal{P}$. Details required to operationalize this definition, such as how patterns are represented and what it means for a pattern to occur in a sequence, will be provided in subsequent sections.

There are many domains of application in which identifying distinctive subsequences is potentially more interesting than discovering common subsequences. Examples include financial time series gathered prior to significant declines or advances in the stock market, time series produced by the monitors in an intensive care unit for patients who die, and traces of the behavior of unauthorized users of computer systems.

One particularly interesting area of application is the extraction of information from multimedia databases, which typically contain tremendous amounts of image, video, audio and textual information. For example, there is currently interest in automated methods for associating names with faces that appear in televised news reports (Houghton 1998; Satoh & Kanade 1997). Given a representation of faces that allows them to be detected in the video stream (e.g. wavelet decompositions (Meza & Favela 1998)), one can identify the associated names in the audio stream as follows. Each time a particular face appears, the audio signal can be recorded over a window centered on the occurrence of the face. The portion of the audio signal referring to the face (i.e. the utterance of the name by the newscaster) may occur before, after or at the same time the face appears, and the relative timing of the two may change from one occurrence to another. In the notation developed above, $\mathcal{R}$ is the audio signal, $\mathcal{E}$ is the occurrence of the face in the video stream, and $\mathcal{P}$ is a pattern in the audio signal that corresponds to the name associated with the face. Clearly, there is a total lack of knowledge concerning the location and the nature of $\mathcal{P}$ in the individual audio time series. Under the reasonable assumption that names are uttered more frequently when their referent is present than when it is absent, it will be the case that $p(\mathcal{P}|\mathcal{R} \wedge \mathcal{E})$ is significantly different from $p(\mathcal{P}|\mathcal{R} \wedge \overline{\mathcal{E}})$. That is, associating names with faces can be cast in terms of finding distinctive, rather than common, subsequences in multimedia databases.

The method for finding distinctive subsequences, which is described in detail in the sections that follow, is a three step process. First, a set of representative or prototypical patterns for the source is obtained by randomly sampling subsequences of length L from $\mathcal{R}$. We call these fixed-length subsequences L-sequences. The L-sequences are then clustered using an appropriate distance metric and cluster prototypes are extracted. Second, the probability of occurrence of each prototype is estimated by sampling additional L-sequences from $\mathcal{R}$ both when the event of interest occurs and when it doesn't. When these probabilities are significantly different for any given prototype, that prototype is a distinctive L-sequence. Finally, distinctive L-sequences are used to identify longer, variable-length distinctive subsequences in the time series. If the events that trigger the collection of time series are not provided by an external "teacher", then the method is completely unsupervised.

For this approach to be effective with large time series databases, it must have modest time and space requirements, and it must be incremental. Our ability to produce large volumes of time series data often outstrips the resources available to store it. The method outlined above is ideal in this sense; it requires time and space that are approximately linear in the total length of the sequences. Although the presentation below is based on a batch implementation of the method, it is trivial to construct an incremental implementation.

The remainder of the paper is organized as follows. Sections 2 – 4 describe the three stages of the approach outlined above. Section 5 describes experiments involving the discovery of distinctive subsequences in data from two very different sources: the sensors of a Pioneer1 mobile robot and the response of a set of bandpass filters to human speech. Finally, section 6 concludes and points to future work.

2 Finding Prototypical L-Sequences

The first step toward the discovery of variable-length distinctive subsequences is the identification of a set of fixed-length subsequences that capture patterns occurring in the data generated by $\mathcal{R}$. This is accomplished by randomly sampling sequences of length L, called L-sequences, from the source. Given n L-sequences and a measure of similarity between multivariate, real-valued time series, we construct an n-by-n similarity matrix. The matrix is used to cluster the L-sequences and to select a prototype from each cluster by finding the sequence that minimizes the average distance to all other sequences in the cluster. For reasons that will become apparent in the next section, we use the k-means clustering algorithm (Mirkin 1996). This is essentially the high level approach outlined in (Das *et al.* 1998), though we explore a measure of similarity that is more appropriate for complex, multivariate time series, and the clusters are put to a very different use.

The cluster prototypes effectively partition the space of all possible L-sequences, with the region of that space assigned to a given prototype being all L-sequences that are more similar to it than any other prototype. The character of that partition is highly dependent on k, the number of clusters, as well as n and L. When k is large, the partition is fine grained, capturing subtle differences between L-sequences. If k is too large then the clusters may capture meaningless differences between L-sequences that are due to noise. When k is small, the opposite situation holds, with each prototype standing for a large class of potentially very different L-sequences. The quality of the clustering is dependent on n, as large samples ensure that the clusters fit structure in the data rather than noise. Finally, L determines the temporal extent of the prototypes. Qualitatively different patterns may emerge at different time scales, with the appropriate time scales being highly domain dependent.

In general, finding a measure of similarity for time series suitable for clustering is not easy because time series that are qualitatively the same may be quantitatively different in at least two ways. First, they may be of different lengths (although this is not the case with L-sequences), making it difficult or impossible to embed the time series in a metric space and use, for example, Euclidean distance to determine similarity. Second, within a single time series, the rate at which progress is made can vary non-linearly. The same pattern may evolve slowly at first and then speed up, or it may begin quickly and then slow down. Such differences in rate make similarity measures such as cross-correlation unusable.

The measure of similarity that we use is Dynamic Time Warping (DTW) (Sankoff & Kruskall 1983). DTW is a generalization of classical algorithms for comparing discrete sequences (e.g. minimum string edit distance (Cormen, Leiserson, & Rivest 1990)) to sequences of continuous values. It was used extensively in speech recognition, a domain in which the time series are notoriously complex and noisy, until the advent of Hidden Markov Models, which offered a unified probabilistic framework for the entire recognition process (Jelinek 1997). Despite DTW's useful properties as a measure of similarity between time series, it has received little attention in the KDD community (Berndt & Clifford 1994).

Given two time series, S_1 and S_2, DTW finds the warping of the time dimension in S_1 that minimizes the difference between them. Consider the two univariate time series shown in the first column of Figure 1. Imagine that the time axis of S_1 is an elastic string and that you can grab that string at any point corresponding to a time at which a value was recorded for the time series. Warping of the time dimension consists of grabbing one of those points and moving it to a new location. As the point moves, the elastic string (the time dimension) compresses in the direction of the motion and expands in the other direction. Consider the middle column in Figure 1. Moving the point at the third time step from its original location to the seventh time step causes all of the points to its right to compress into the remaining available space, and all of the points to its left to fill the newly created space. Of course, much more complicated warpings of the time dimension are possible, as with the third column in Figure 1 in which four points are moved.

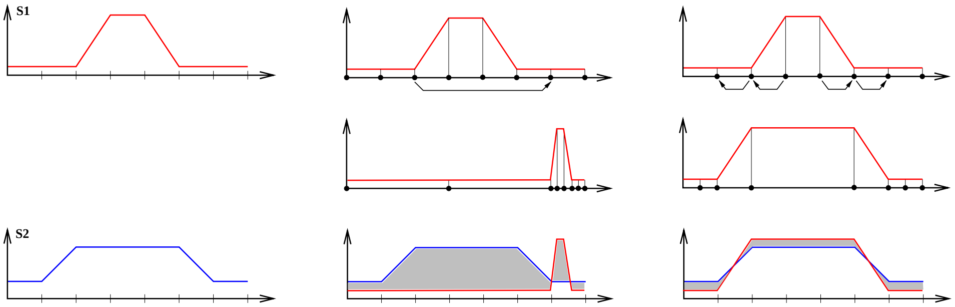

Figure 1: Two time series, S_1 and S_2, (the leftmost column) and two possible warpings of S_1 into S_2 (the middle and rightmost columns).

Given that a warping of the time dimension in S_1 yields a new time series that we will denote S_1', one can compare the similarity of S_1' and S_2 by determining the area between the two curves. That area is shown in gray in the bottom row of Figure 1. Note that the first warping of S_1 in which a single point was moved results in a poor match, that is, one with a large area between the curves. However, the fit given by the second, more complex warping is quite good. DTW returns the optimal warping of S_1, the one that minimizes the area between S_1' and S_2, and the area associated with that warping. This area is used as a measure of similarity between the two time series. In general, the area between S_1' and S_2 may not be the same as the area between S_2' into S_1. We use a symmetrized version of DTW that essentially computes the average of those two areas based on a single warping (Kruskall & Liberman 1983).

This stage of the three step process, which yields a set of prototypical L-sequences, requires $O(n^2)$ space to store the similarity matrix and $O(n^2)$ time to fill the matrix and construct clusters. Even though there are exponentially many ways to warp the time dimension of a sequence, DTW uses dynamic programming to find the optimal warping in time that is a low order polynomial of the lengths of S_1 and S_2, i.e. $O(|S_1||S_2|)$. In the case of L-sequences, the time is $O(L^2) = O(1)$. Although a straightforward implementation of DTW is more expensive than computing Euclidean distance or cross-correlation, there are numerous speedups that both improve the properties of DTW as a distance metric and greatly reduce the constant factor hidden inside the $O(1)$ notation.

3 Identifying Distinctive L-Sequences

The previous section described how prototypical L-sequences are obtained by sampling from $\mathcal{R}$ without regard to whether the event $\mathcal{E}$ occurs. Because distinctive patterns, by definition, occur more or less frequently in the presence of $\mathcal{E}$ than in its absence, sampling in this manner ensures that clustering has access to the full range of patterns that can occur within a window of width L. Given k prototypical L-sequences, $\mathcal{P}_1$ through $\mathcal{P}_k$, we now want to determine which of them are distinctive. That is, we want to identify those prototypes for which $p(\mathcal{P}_i|\mathcal{E})$ is significantly different from $p(\mathcal{P}_i|\overline{\mathcal{E}})$.

Estimation of $p(\mathcal{P}_i|\mathcal{E})$ and $p(\mathcal{P}_i|\overline{\mathcal{E}})$ requires a set of sequences obtained from $\mathcal{R}$. This set must contain some sequences that co-occurred with $\mathcal{E}$ and some that did not. A window of width L is passed over each sequence, and DTW is used to determine which of the k prototypes is most similar to each of the resulting L-sequences. The L-sequences obtained in this manner are drawn from larger sequences that either did or did not co-occur with $\mathcal{E}$. If an L-sequence is most similar to prototype i and the former case holds, the counter $n_{i,\mathcal{E}}$ is incremented. If the latter case holds the counter $n_{i,\overline{\mathcal{E}}}$ is incremented. It is then a simple matter to estimate the probabilities of interest:

$$p(\mathcal{P}_i|\mathcal{E}) = \frac{n_{i,\mathcal{E}}}{\sum_{j=1}^{k} n_{j,\mathcal{E}}} \qquad\qquad p(\mathcal{P}_i|\overline{\mathcal{E}}) = \frac{n_{i,\overline{\mathcal{E}}}}{\sum_{j=1}^{k} n_{j,\overline{\mathcal{E}}}}$$

To determine whether these probabilities are significantly different we use a two-tailed t-test as follows. Consider a random variable whose value is either 1 or 0 depending on whether an L-sequence matches or does not match the i^{th} prototype. If that random variable is associated with only those L-sequences that co-occurred with $\mathcal{E}$, then it has the following mean and variance:

$$\mu_{i,\mathcal{E}} = \frac{n_{i,\mathcal{E}}}{\sum_{j=1}^{k} n_{j,\mathcal{E}}} = p(\mathcal{P}_i|\mathcal{E})$$

$$\sigma_{i,\mathcal{E}}^2 = n_{i,\mathcal{E}}(1 - p(\mathcal{P}_i|\mathcal{E}))^2 + (\sum_{j=1}^{k} n_{j,\mathcal{E}} - n_{i,\mathcal{E}})p(\mathcal{P}_i|\mathcal{E})^2$$

The mean and variance of the random variable associated with L-sequences that did not co-occur with $\mathcal{E}$ is analogous. It is then straightforward to apply a standard t-test to determine the probability of making an error in rejecting the null hypothesis that $\mu_{i,\mathcal{E}} = \mu_{i,\overline{\mathcal{E}}}$, i.e. that prototype i is not a distinctive L-sequence. If that probability is below a given significance level, then the L-sequence is said to be distinctive.

Aside from the precise technical definition of distinctiveness given above, what does it mean for a prototypical L-sequence to be distinctive? Simply put, it means that occurrences of the event $\mathcal{E}$ and a particular pattern of fixed length that occurs in the time series are not independent. Dependence need not be the result of a causal relationship, though non-independence is frequently cited as one of the necessary conditions for causality (Suppes 1970). For applications in mining multimedia databases, discovering dependence relationships, rather than causality, is vitally important. While there is clearly dependence between, for example, the words recorded on the audio track and images recorded on the video track, it is not the case that one causes the other.

The space requirements of this stage of the process are modest. Only the k prototypical L-sequences and their associated counters need to be retained permanently. It is possible to process sequences obtained from the source incrementally, constructing a new L-sequence each time a new vector is produced and determining which counters to update. The time needed to determine the prototype that is most similar to a given L-sequence is $O(kL^2) = O(1)$. The number of L-sequences to be processed is linear in the total length of the larger sequences obtained from the source. Assuming that total is greater than n^2, where n is the number of samples used to create the clusters, this stage is the most expensive, requiring computation that is linear in the size of the inputs.

4 Variable Length Distinctive Subsequences

The previous two sections described how to obtain prototypical L-sequences and to determine which of the prototypes are distinctive. This section explains how the fixed-length prototypes are used to identify variable-length distinctive subsequences that span more than L time steps.

Note that prototype i can be distinctive for one of two reasons. Either $p(\mathcal{P}_i|\mathcal{E}) \gg p(\mathcal{P}_i|\overline{\mathcal{E}})$ or $p(\mathcal{P}_i|\mathcal{E}) \ll p(\mathcal{P}_i|\overline{\mathcal{E}})$. If the former condition holds we say that all L-sequences that are more similar to $\mathcal{P}_i$ than any other prototype are *frequent* L-sequences. Such L-sequences occur more frequently in the presence of $\mathcal{E}$ than in its absence. If the latter condition holds we say that all of the L-sequences that are more similar to $\mathcal{P}_i$ than any other prototype are *infrequent* L-sequences. Finally, L-sequences matching prototypes that are not distinctive are said to be *neutral*.

A subsequence of length greater than L is frequent if all of the L-sequences that it contains are either frequent or neutral. The subsequence is infrequent if those L-sequences are either infrequent or neutral. In both cases, the subsequence is distinctive. It is possible to locate all of the frequent and infrequent variable-length subsequences in a larger time series in time that is linear in the length of the time series. In practice, we amend this definition in two ways. First, frequent subsequences must start and end with frequent L-sequences (likewise for infrequent subsequences). Second, there can be no more than *max-neutral* consecutive neutral L-sequences in the subsequence, thereby ensuring that occurrences of frequent (or infrequent) L-sequences in the subsequence are temporally proximal. Without this restriction, a subsequence that started and ended with frequent L-sequences and that contained millions of intervening neutral L-sequences would be deemed a frequent subsequence.

5 Empirical Results

This section presents the results of applying the method just described for discovering distinctive subsequences to two very different sources of data: the sensors of a Pioneer1 mobile robot and the response of a set of bandpass filters to human speech. The data are meant to be similar in quality, not quantity, to that which might be found in a multimedia database. The audio portion of the data is human commentary on the activities of the robot, and the sensor data reflects the robots ability to perceive what the human is describing. The idea is that one can use discrete events in either the audio or sensor data to identify correlated patterns in the other data stream. In the former case, the goal of the experiment is to identify patterns in the robot's sensors that correspond to the referents of words (assuming that words are presented as discrete tokens). In the latter case, the goal is to identify words, patterns in the raw waveform of continuous speech, that denote salient features in the environment (assuming that salient features are presented as discrete tokens).

5.1 Identifying the Referents of Words

The first application of the method for identifying distinctive subsequences involves identifying the referents of words in the sensors of a Pioneer1 mobile robot. The Pioneer has a pair of drive wheels that allow translational and rotational motion, and a gripper that can be used to pick up small objects. Its sensors include an array of seven sonars, a bumper on the end of the gripper that indicates when the gripper is touching something, and an infrared break beam between the gripper paddles that indicates when an object is inside the gripper.

The values of these and a variety of other sensors are recorded ten times each second.

We made a videotape of 41 different scenes in which the Pioneer engaged in simple activities in a lab environment that included trash cans, partitions, a toy car, cups, a large box and mats of different colors on the floor. For example, in one scene the robot picked up a cup that was sitting on a blue mat and carried it to a red mat. The video was then shown to several human subjects who were instructed to generate one sentence for each scene that described what the robot was doing. No restrictions were placed on the vocabulary or the grammar the subjects could use.

For each of the 41 scenes, a time series was created by recording the values of the break beam, the gripper bumper and the state of the gripper. The gripper can be in one of three states: down and open, up and closed, or moving between these two positions. The break beam can either be on (object present) or off (no object present), and the bumper can either be on (touching an object) or off (not touching an object). Seven prototypical patterns in these time series were obtained by clustering 200 samples that each covered one second of real time. The resulting prototypes are shown in Table 1. They cover a variety of physically realizable situations. For example, $\mathcal{P}_3$ corresponds to a situation in which the gripper is down and touching an object, but nothing is between the gripper paddles. This might occur when the robot has run into a large obstacle such as a wall or a trash can.

Prototype	Gripper State	Break Beam	Gripper Bumper
$\mathcal{P}_1$	down	off	off
$\mathcal{P}_2$	up	off	off
$\mathcal{P}_3$	down	off	on
$\mathcal{P}_4$	down	on	off
$\mathcal{P}_5$	up	on	off
$\mathcal{P}_6$	moving	on	off
$\mathcal{P}_7$	up	off	on

Table 1: How prototypes obtained from the gripper time series relate to the state of the gripper.

Next, from all of the words used by the human subjects to describe the robot's activities, three were chosen that are particularly relevant to the gripper. They are "pushed", "picked" (as in "The robot picked up the red cup") and "raised". Each of these words was used to divide the time series into two sets based on whether the word co-occurred with the associated scene. For example, all of the time series that at least one subject described as involving pushing were placed in one set, and all of the time series that were never described as involving pushing were placed in another set. Distinctive prototypes for each word were then identified by drawing 200 additional L-sequences from the sets. The value of α used was 0.05.

Word	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$
`pushed`	-		+	+	-	-	+
`picked`		-	-	+	+	+	-
`raised`		+	-	-	-	-	-

Table 2: Frequent (+) and infrequent (-) prototypical L-sequences for four of the words used to describe scenes in the video. Empty cells in the table indicate that the prototype was not distinctive for the word.

Table 2 shows for each combination of word and prototype whether the prototype was frequent (+), infrequent (-) or neutral (blank) for the word. Consider the frequent distinctive prototypes associated with the word "pushed". They capture aspects of the three ways that the robot can push objects: by getting small objects between its gripper paddles and moving ($\mathcal{P}_4$), by putting its gripper against large objects (that won't fit between the paddles) and moving ($\mathcal{P}_3$), and by pushing against objects of any size with the gripper raised and closed ($\mathcal{P}_7$). Visual inspection of the variable-length subsequences identified with the information in Table 2 indicates that three qualitatively different types of subsequences were identified, corresponding to the three kinds of pushes. In addition, the method successfully located the portions of the time series involving picking up objects and raising the gripper while empty.

5.2 Discovering Words in Continuous Speech

This subsection presents the results of an experiment in which presence of specific visual features, such as salient objects or people, serves as the event, $\mathcal{E}$, that triggers collection of time series. The source of time series, $\mathcal{R}$, is the raw audio waveform. The patterns of interest,

$\mathcal{P}$, are subsequences that correspond to the word that denotes $\mathcal{E}$. Under the assumption that words are uttered more frequently in the presence of their referents than in their absence, it will be the case that $p(\mathcal{P}|\mathcal{R} \wedge \mathcal{E}) > p(\mathcal{P}|\mathcal{R} \wedge \overline{\mathcal{E}})$, and $\mathcal{P}$ will be a distinctive subsequence.

Consider the following simple scenario. Suppose a video contains scenes in which objects of various sizes, shapes and colors stand in certain spatial relationships, and that each scene is accompanied by a descriptive utterance. Visual features of the scenes can be used to partition the utterances into two sets, one containing utterances that co-occurred with a particular feature and one containing utterances that did not. For example, every time a blue object appears in the scene, the accompanying utterance is placed in one set, and all other utterances are placed in a different set. The procedure outlined in sections 2 – 4 can then be used to identify occurrences in the speech waveform of the word that denotes blue.

The experiment in previous subsection assumed that individual words in the speech stream were presented as tokens, and that these tokens were used to drive the search for patterns in the robot's sensors. The experiment in this section assumes that the presence of particular visual features are presented as tokens and that they are used to drive the search for patterns in the speech stream.

The above scenario was simulated by randomly generating 100 sentences according to the grammar shown in Figure 2. Each sentence was read aloud and digitized by sampling at a rate of 8000Hz. The resulting signal was passed through a bank of eight digital bandpass filters that covered frequencies from 150Hz to 3900Hz (Picone 1993). Every 10ms the average response of each filter over the preceding 32ms was recorded. This preprocessing phase, which is standard practice in the speech recognition community, was used to convert each digitized sentence into a multivariate time series containing eight component series.

```
S             ⇒  the OBJECT is RELATIONSHIP the OBJECT
OBJECT        ⇒  SIZE COLOR SHAPE
SIZE          ⇒  tiny | small | medium | large
COLOR         ⇒  red | purple | green | blue
SHAPE         ⇒  circle | square | triangle | rectangle
RELATIONSHIP  ⇒  on | under | over | touching
```

Figure 2: The grammar used to generate sentences for word identification experiments.

Fifteen prototypical L-sequences were obtained by clustering 400 samples drawn from the 100 time series, with each sample spanning 100ms. If a particular terminal, such as red, occurs in a sentence, it is assumed that the corresponding feature appeared in the scene. To simulate the dependence of word occurrences on scene features, each terminal in the grammar was used to divide the sentences into two sets based on whether the terminal occurred. Then the distinctive prototypes associated with each terminal were identified with an additional 4000 samples drawn from each set. The value of α used was 0.05.

Word	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	$\mathcal{P}_8$	$\mathcal{P}_9$	$\mathcal{P}_{10}$	$\mathcal{P}_{11}$	$\mathcal{P}_{12}$	$\mathcal{P}_{13}$	$\mathcal{P}_{14}$	$\mathcal{P}_{15}$
tiny		+				-							-		+
small	-		-										+		
medium	+			-		+	+	+	-	-	-		-	-	-
large	-			+						+	-	+	+		
red		+	+		-		+					-	-	-	
purple	-		-		+	-	-			-		+	+		
green			-									+			+
blue	+				-		-								
circle		-		+		-	-					+	+		
square	+	-				-	+	-	+	-		+	-		
triangle				-	-	+	-			+		-			
rectangle		+		-		+			-	-	+			-	
on	-									+					
under	+	+		-		-						-			+
over			-	+			+			-					
touching					+	-			+					+	-

Table 3: Frequent and infrequent prototypical L-sequences for each of the words in Figure 2 are marked with + and - respectively. Empty cells in the table indicate that the prototype was not distinctive for the word.

Table 3 shows for each combination of terminal and prototype whether the prototype was frequent (+), infrequent (-) or neutral (blank) for the terminal. There are several interesting things about this table. First, all of the prototypes are distinctive for at least one of the words, indicating that the number of clusters is not too large. Second, no two words share the same pattern of distinctive prototypes, and those patterns are often quite different. This suggests that the number of clusters is large enough to capture differences in patterns that are sufficient, at least in principle, to distinguish occurrences of the different words in the speech stream. Finally, because there are large differences between columns, it appears that L-sequence clustering is doing a good job of finding clusters that correspond to significantly different patterns in the speech stream.

To test the utility of the information presented in Table 3, four scene features were selected and 20 new sentences involving each feature were generated (for a total of 80 new sentences). The maximal length frequent subsequence in each of the associated time series

	Hits			
Word	Exact	Over-sized	Under-sized	Misses
`touching`	19	0	0	1
`medium`	16	0	2	2
`triangle`	0	0	18	2
`red`	14	0	0	6

Table 4: The results of applying the distinctive prototypes in Table 3 to identify occurrences of specific words in the speech stream.

was obtained, and its location in the time series with respect to the utterance of the word that denotes the feature was determined. The results are summarized in Table 4. If the maximal length frequent subsequence spanned at least 95% of the utterance it was recorded as an exact hit. If in addition it covered more than 5% of an adjacent word the result was an over-sized hit. Under-sized hits occurred when less than 95%, but more than 0%, of the utterance is covered. If none of the utterance was covered, the result was a miss. The results for `touching` and `medium` are quite good. Virtually all of the hits for `triangle` are undersized. The reason is that `triangle` and `rectangle` share the suffix `angle`, and 80% of the sentences contain one of the two words. Therefore, only `tri` was determined to be distinctive. Even though six occurrences of `red` were missed, the 14 hits were exact and should be sufficient to construct a model (e.g. a hidden Markov model) of the waveform associated with that word.

6 Discussion and Future Work

Interest in time series problems appears to be increasing in several different scientific communities, include machine learning and knowledge discovery in databases. Although we know of no other work directed at identifying distinctive subsequences in time series, many recent results address parts of the problem. For example, (Agrawal *et al.* 1995) and (Keogh & Pazzani 1998) both describe methods for measuring similarity between continuous time series for purposes of clustering and identifying common subsequences. However, these approaches are limited to univariate time series and are therefore not applicable to problems such as the ones described in section 5 in which one time series alone is insufficient for making the appropriate discriminations.

The primary goal of future work is to explore the scalability of this approach to large multimedia databases.

References

Agrawal, R.; Lin, K.; Sawhney, H. S.; and Shim, K. 1995. Fast similarity search in the presence of noise, scaling and translation in time series databases. In *Proceedings of the 21st International Conference on Very Large Databases.*

Berndt, D. J., and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *Working Notes of the Knowledge Discovery in Databases Workshop*, 359–370.

Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1990. *Introduction to Algorithms.* The MIT Press.

Das, G.; Lin, K.-I.; Mannila, H.; Renganathan, G.; and Smyth, P. 1998. Rule discovery from time series. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 16–22.

Houghton, R. A. 1998. Ricky a. houghton. In *Working Notes of the CONALD Workshop on Mixed-Media Databases.*

Jelinek, F. 1997. *Statistical Methods for Speech Recognition.* MIT Press.

Keogh, E., and Pazzani, M. J. 1998. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Working Notes of the AAAI-98 workshop on Predicting the Future: AI Approaches to Time-Series Analysis*, 44–51.

Kruskall, J. B., and Liberman, M. 1983. The symmetric time warping problem: From continuous to discrete. In *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Wesley.

Kruskall, J. B., and Sankoff, D. 1983. An anthology of algorithms and concepts for sequence comparison. In Sankoff, D., and Kruskall, J. B., eds., *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley.

Meza, V., and Favela, J. 1998. Information retrieval by image content and its associated text in a digital library. In *Working Notes of the CONALD Workshop on Mixed-Media Databases*.

Mirkin, B. 1996. *Mathematical Classification and Clustering*. Kluwer Academic Publishers.

Picone, J. W. 1993. Signal modeling techniques in speech recognition. *Proceedings of the IEEE* 89(9):1215–1247.

Sankoff, D., and Kruskall, J. B. 1983. *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley.

Satoh, S., and Kanade, T. 1997. Name-it: Association of face and name in video. In *Computer Vision and Pattern Recognition*.

Suppes, P. 1970. *A Probabilistic Theory of Causality*. Amsterdam: North Holland.