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

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.

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]

[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.

KDD-99 San Diego CA USA

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 produced 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.

The remainder of the paper is organized as follows. Section 2 describes the algorithm for finding distinctive subsequences in multivariate, real-valued time series. Section 3 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 4 concludes and points to future work.

2 The Algorithm

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 and cluster the L-sequences. Then for each of the k resulting clusters, where k is a user-specified parameter, we select a prototype by finding the sequence that minimizes the average distance to all other sequences in the cluster. 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.

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).

The second step of the algorithm is to determine which of the k prototypical L-sequences, $\mathcal{P}_1$ through $\mathcal{P}_k$, 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 using $n_{i,\mathcal{E}}$ and $n_{i,\overline{\mathcal{E}}}$.

The final step of the algorithm uses the fixed-length prototypes 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.

3 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.

3.1 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.

3.2 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 pattern of interest, $\mathcal{P}$, is the subsequence that corresponds 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 section 2 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 a grammar that creates sentences describing objects of various colors, shapes and sizes standing in various spatial relationships to one another. 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.

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.

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.

	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.

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 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.

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 the grammar are marked with + and - respectively. Empty cells in the table indicate that the prototype was not distinctive for the word.

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.

4 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 3 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.

Acknowledgments

This research is supported by DARPA and AFOSRF under contract numbers DARPA/AFOSRF49620-97-1-0485 and DARPAN66001-96-C-8504. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, AFOSRF or the U.S. Government.

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.

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 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.

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.