

FINDING PATTERNS IN SEQUENCES: COMPARISON OF MOTIF EXTRACTION,  
DYNAMIC TIME WARPING, AND HIDDEN MARKOV MODEL APPROACHES, WITH  
APPLICATIONS TO THE TIMSS 1999 VIDEO STUDY

BY

SUJAI KUMAR

M.S., Birla Institute of Technology and Science, 1998

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Education  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2004

Urbana, Illinois

## ABSTRACT

This paper will present three different approaches to finding patterns in sequences—motif extraction, dynamic time warping distances, and hidden Markov models. These methods have been used in fields as diverse as bio-informatics and speech recognition, but are equally useful in analyzing categorical sequential data. Motif extraction is the process of looking for recurring patterns of codes in sequences. Dynamic time warping is an algorithm that finds the distance between pairs of sequences and can be used to find clusters in a set of sequences. Hidden Markov models use a probabilistic approach to studying the common underlying structure of a group of sequences. In the first part of this paper, I provide an overview of each technique and compare their advantages and limitations. In the second part, I apply these three methods to the TIMSS 1999 video study—a rich data set with time-coded information about the events taking place in 638 mathematics lessons from seven countries. The results provide new insights into the data that would not have been possible with traditional methods of prevalence analysis.

## TABLE OF CONTENTS

LIST OF FIGURES .....	v
CHAPTER 1 INTRODUCTION .....	1
CHAPTER 2 METHODS FOR FINDING PATTERNS IN SEQUENCES .....	3
Types of Sequences.....	5
Algorithms for Sequence Analysis and Comparison.....	9
Summary of Methods.....	23
CHAPTER 3 APPLICATIONS TO THE TIMSS 1999 VIDEO STUDY .....	25
Creating Lesson Sequences From the TIMSS Dataset .....	30
Results of Three Sequence Analysis Methods.....	35
CHAPTER 4 DISCUSSION.....	58
Evaluating the Methods .....	58
Differences Between Countries .....	60
REFERENCES .....	62

## LIST OF FIGURES

Figure	Page
1 Assigning codes to instantaneous events .....	6
2 Assigning codes to events that last for clearly marked durations of time .....	6
3 The fundamental sequence for this episode would be A B B C .....	7
4 Web interface to the text symbol pattern discovery tool based on the Teiresias algorithm.....	12
5 Average percentage of eighth grade mathematics lesson time devoted to independent problems, concurrent problems, and answered-only problems, by country:1999 (from TIMSS Report) .....	26
6 Average time per independent problem per eighth-grade mathematics lesson (in minutes), by country: 1999 .....	27
7 Average percentage of eighth-grade mathematics lesson time devoted to various purposes, by country: 1999.....	28
8 Lesson signature for the purpose code “Review”, for all Australian lessons.....	29
9 Number of lessons in each country and average lesson length (in minutes) in the TIMSS 1999 video study .....	30
10 Describing a time point in a lesson using single codes, code pairs, and code triples....	34
11 Motifs in fundamental P code sequences, by country .....	38
12 Motifs in fundamental CI code sequences, by country.....	40
13 Motifs in fundamental Act code sequences, by country.....	42
14 MDS map of DTW distances between fundamental P sequences.....	45
15 MDS map of DTW distances between fundamental P sequences: Japanese and US lessons.....	46
16 MDS map of DTW distances between fundamental P sequences: Czech and Hong Kong lessons .....	47
17 MDS map of DTW distances between fundamental CI sequences .....	48

18	MDS map of DTW distances between fundamental CI sequences: Japanese and Dutch lessons .....	49
19	MDS map of DTW distances between fundamental CI sequences: Czech and US lessons.....	50
20	MDS map of DTW distances between 2% proportion-segmented P sequences .....	51
21	MDS map of DTW distances between 2% proportion-segmented sequences based on P-CI-Act triples.....	52
22	Likelihood of sequences within countries to fit the HMMs for all countries .....	55
23	Likelihood of all lesson sequences to fit the HMMs for all countries.....	56

## CHAPTER 1

### INTRODUCTION

Any episode that extends over time can be described as a sequence of events. Such sequences can provide valuable information about the relationships between events and characteristic patterns of activities. Events that consistently co-occur in the same order can be clues to cause-effect relationships and reveal underlying similarities among different sets of events. Together with more traditional prevalence analyses, sequences can be used to provide a more nuanced and comprehensive picture of the events being studied.

This paper will concentrate only on temporal sequences, ignoring spatial or numeric sequences. The techniques for finding patterns in sequences to be described in this study have been inspired by research in the biological sciences, speech recognition, computer science, and quantitative psychology. I will present and compare a few methods that are especially applicable to data sets in the social sciences, and highlight the kinds of conclusions that we can draw with reference to data from the TIMSS 1999 video study.

The TIMSS 1999 video study (Hiebert et al., 2003), the largest study of its kind, analyzed over 600 videotaped mathematics lessons chosen from seven countries. This immensely rich data source on educational practices around the world has led to many new findings on social processes, pedagogical styles, and mathematical knowledge, and how they differ or are similar across countries. It is thus a perfect dataset for demonstrating the value of sequence analysis in finding patterns within and across sequences of events that make up each lesson.

The first part of this thesis highlights methods and techniques that can be used to find patterns in sequence data. This section deliberately makes no reference to the TIMSS data because the techniques are generic and applicable to a variety of data sources. The section

begins with an overview of what these methods can tell us compared to more traditional statistical techniques, defines the terms used, and then describes three methods for finding patterns in sequences. The first of these approaches is the extraction of motifs, where fast algorithms detect recurring sub-sequences of elements within longer sequences. The second approach uses dynamic time warping, a technique popular in the field of speech processing, for determining how similar two sequences are. The resulting similarity data can then be analyzed to look for clusters of similar sequences. The third and last approach is the use of hidden Markov models in looking for underlying structural similarities in sequences that may have no obvious similarities on the surface.

The second part of this thesis describes the application of each technique – motif extraction, dynamic time warping, and hidden Markov models – to the sequences of lesson events that were coded in the TIMSS study classrooms. The results of these analyses will help answer questions such as:

1. What are the most commonly recurring patterns (motifs) of lesson events?
2. Are some of these motifs more prevalent in some countries compared to others?
3. Do lessons from the same country tend to have similar sequences of lesson events?
4. Are there similarities between sequences of lesson events from different countries?
5. Can we determine any underlying structural similarities in a group of lessons?

Finally, in the last part, I will summarize the advantages and limitations of each analytic method in the context of the TIMSS data, and the factors to keep in mind when applying these methods to other sequence data in the social sciences. I will also discuss the results obtained from the TIMSS data given what we already know about the similarities and differences among the countries in the study.

## CHAPTER 2

### METHODS FOR FINDING PATTERNS IN SEQUENCES

Events that take place over time lend themselves to different kinds of scrutiny. Many statistical analyses of temporal event data concentrate solely on the prevalence of each type of event. This prevalence analysis can be done by recording the frequencies of the occurrence of events or recording the amount of time spent on specific events. These data are useful because they can be processed to indicate the relative importance of one event over another.

However, whether some type of event is more prevalent than another is only one part of the story. The order in which these events happen can also provide us with clues about the cause and effect relationships among the events. Social interactions in particular are highly interactive, and the order of actions, utterances or behaviors in such settings can reveal a lot more about the process being studied than a description that only records the occurrence of an event and not its sequential order.

The importance of looking at the order of events is illustrated in the following example. In the three sequences below, each letter (code) represents an event type that lasted for 1 minute. Each code occurs exactly the same number of times in each sequence.

1. A A A A A A A A A A B B B B B B B B B B C C C C C C C C C C
2. A B A B A B A B A B A B A B A B A B A B C C C C C C C C C C
3. A B C A B C A B C A B C A B C A B C A B C A B C A B C A B C

On the basis of code prevalence alone, these sequences would be considered identical. However, a quick scan of the ordering of codes suggests a very different story. In the second sequence, for example, “AB” is a recurring sub-sequence or motif. The third sequence contains an “ABC” motif. Depending on what the codes represent, these motifs could indicate patterns in lesson activities that would go unnoticed in traditional analyses of code prevalence.

Even if there are no clear motifs to be found, the following examples demonstrate how useful sequence comparisons can be when used in conjunction with prevalence analysis:

1. A A A A A A A A A B B B C C C A A A A A A A A

2. A A A B B B B B B B B C C A A A A A A

The above examples show how two sequences may look fairly different if we compare the amount of time spent on each code type. If each letter represents a minute, then the amount of time spent on B and A is not even proportionally related in the two episodes. A lasts almost twice as long (17 minutes) in the first episode than in the second (9 minutes), while B lasts only about one-third as long (3 minutes in the first episode as compared to 8 minutes in the second). However, from one perspective, both sequences are playing out the same script – starting with event type A, followed by B, followed by C, and ending with A. Methods for comparing observations should be able to account for variations in the amount of time spent on each event and should be able to discover structural similarities in the sequences of events.

Apart from finding motifs and structural similarities, sequences of events can also be used to answer conditional probability questions such as “Given that an event of type A occurred, how likely is it that event B will occur on the next turn”, or “How often is event A preceded by event B within the previous two minutes”. Techniques developed by Allison (1984) and Bakeman and Gottman (1997) deal with these kinds of questions and are especially useful tools for discovering cause and effect relationships among specific events.

The sample sequences presented so far have been short and relatively simple. A quick glance is all that is necessary to find any regularities at the sub-sequence level or to find similarities across sequences. However, if the sequences are longer and more complicated, or if there are many more sequences to be analyzed, it becomes increasingly difficult to look for patterns and similarities without the use of computational methods. Such methods for

discovering patterns in sequences have become immensely popular in the last three decades thanks to newer and faster algorithms and computers. These methods have been developed primarily in fields like bio-informatics and computer science, and their applications range from determining the evolutionary distance between DNA or protein sequences (by measuring the number of mutations needed to get from one to the other), to mining large databases of sales information for identifying buying patterns among customers.

These methods cannot always be applied directly to sequences of observational data in the social sciences because they might only work with numerical data, or only with non repeating sequences, etc. In an effort to work around these limitations, I have adapted a portion of the vast literature on sequence patterns and comparisons to work with the categorical data that is typically found in coded behavioral observations.

In the following pages, I will first define the types of sequences that will be analyzed using these methods. This will be followed by the actual algorithms and methods for finding patterns in the sequences. Given a set of sequences, the goals of the methods presented will be to:

1. Find recurrent motifs.
2. Examine similarities between sequences and determine clusters of sequences.
3. Find underlying structural similarities in a set of sequences.

### Types of Sequences

In recording behavioral observations (such as mother-child interactions or the type of activity in a classroom at a given point of time), there are basically two types of records that can be created. In the first kind, events that occur at specific instances of time are recorded (such as

the moment when a child picks up a toy, or the time when an interruption occurs) as in Figure 1. The timeline shows that the entire episode recorded lasted for 6 minutes and 5 seconds, that an event of type A happened at 0:30 and 2:20, an event B happened at 2:00 and an event C happened at 4:08.

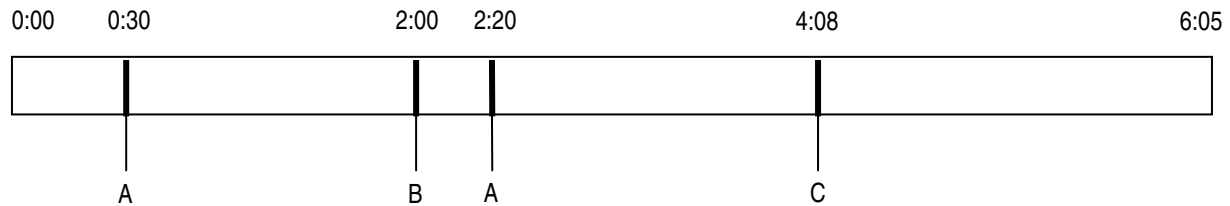


Figure 1. Assigning codes to instantaneous events.

Contrast this example with the second case, where types of activities last for a certain amount of time, and their start and stop times are clearly marked (Figure 2). In this figure, A lasts from 0:00 to 3:00, B lasted from 3:00 to 4:00 and C lasted from 4:00 to 6:00.

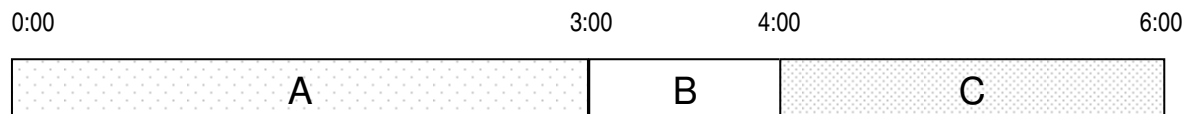


Figure 2. Assigning codes to events that last for clearly marked durations of time.

In this study, I discuss the second type of coding system. An *event* is therefore defined as something that happens over a specified period of time, although some of the methods that we present will also apply to events that occur at specific instances in time.

To simplify the description of the methods, we are going to restrict ourselves to codes that do not overlap at all. A potential problem is that the coding system might not be comprehensive enough to cover every part of the episode. That is to say, there might be parts of

an episode or events that have no code assigned to them, resulting in gaps in the coded data. To ensure that every part of the lesson is coded, we can assign a dummy or blank code to those gaps.

We define a *sequence* as an ordered set of character codes such as:

A B C D A A B B

An *episode*, made up of coded events that occur for specified periods of time, can be recorded as one of three types of sequences: fundamental sequences, time-segmented sequences, and proportion-segmented sequences. These terms are non-standard but provide a convenient shorthand for referring to the process by which they were created.

### *Fundamental Sequence*

In this type of sequence, only the occurrence of an event is recorded, not its duration. It is called the fundamental sequence because it represents the most fundamental aspect of the events—the order in which they occurred. The episode in Figure 2 would be represented as the following fundamental sequence:

A B C

If an episode consists of two distinct events of the same type happening one after the other, then the fundamental sequence can also have duplicate codes (Figure 3). This would be true if the coding system records events such as “student reads a page” as distinct events for each page read.

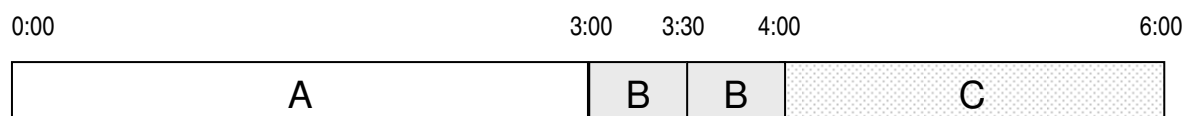


Figure 3. The fundamental sequence for this episode would be A B B C .

A fundamental sequence will represent every event that occurs, even if it is recorded for only a few seconds. In some ways, this can be considered a disadvantage, because it equally weights events whether they last 30 minutes or 30 seconds.

### *Time-Segmented Sequence*

As the name suggests, a time-segmented sequence is obtained by dividing an episode into segments of a fixed length of time. For example, if the time unit chosen is 1 minute, then the resulting sequence would be made up of the code that occurs in the first minute, followed by the code that occurs in the second minute, and so on. Ideally, the time units should be chosen to ensure that only one event can occur within any one time unit. Even so, there may be instances of two codes occurring within the same time unit (at event boundaries, for example). In such cases, the most prevalent code is chosen to represent that time unit.

If the time unit chosen is one minute, then the time-segmented sequence for the episode in Figure 3 would be:

A A A B C C

A 30 second time-segmented sequence for the same episode would be:

A A A A A A B B C C C C

Note that consecutive events of the same type are not differentiated from a single long block of time spent on that event (i.e., both Figures 2 and 3 would result in the same sequence).

### *Proportion-Segmented Sequence*

In a proportion-segmented sequence, the episode is divided into a fixed number of segments, and the time duration of each segment is a set proportion of the episode duration.

For example, a 10% proportion-segmented sequence of the episode in Figure 3 would divide it into 10 segments of 36 seconds each (10% or 1/10th of six minutes) and would look like:

A A A A A B B C C C

As in the time-segmented sequence, consecutive events of the same type are not differentiated from one long event of that type. The advantage of a proportion-segmented sequence is that it provides a good baseline for some of the algorithms discussed in the next section, because all the sequences to be analyzed are the same length so there are no length effects in the calculations.

### *Summary*

The three kinds of sequences that can be created from time-coded episodes differ from each other in subtle ways. Fundamental sequences capture every code used while the time-segmented and proportion-segmented sequences may leave some out depending on the size of the segments chosen. However, the latter two types of sequences are better representations if the relative durations of each event are important. If the exact durations of individual events are important, then the time-segmented sequences are the only ones that capture that information. The choice of sequence type will depend on the dataset and the research questions being asked.

## Algorithms for Sequence Analysis and Comparison

The three approaches described in this section are a) motif extraction, b) dynamic time warping, and c) hidden Markov models. I will provide a short background to each and review some of the related approaches before describing in detail the specific techniques that I will use in analyzing the TIMSS data in the next part of this paper.

### *Motif Extraction*

Motifs are sub-sequences of longer sequences that occur more than once within these longer sequences. A *sub-sequence* in our study is defined as one or more consecutive elements selected from the longer sequence of elements. Thus, “a b” is a sub-sequence of “a b a c u s”, “d

r e s s” is a sub-sequence of “a d d r e s s e s”, and, even though this is a trivial example, “e” is a sub-sequence of “t h e”. This construct is sometimes also known as a *substring*.

The term *sub-sequence* may have a different meaning in some contexts. According to some computer science texts such as Gusfield (1997), a sub-sequence does not have to contain consecutive elements from the parent sequence. For example, “d r s s” is a valid sub-sequence of “a d d r e s s e s” using this definition, because even though the elements in the subset are not consecutive, they maintain the same ordering relationship as the corresponding elements in the parent sequence. This is not the definition that this paper will be using, however.

Before looking for motifs, the shortest acceptable motif length must be specified. In the following example, if the minimum motif length is assumed to be 3, then several repeating sub-sequences can be found:

1. A B A B C B A
2. B C B C A B C
3. A B D A B D

The three motifs in this example are A B C (occurring at position 3 in the first sequence, and at position 5 in the second sequence), B C B (at position 4 in the first sequence and position 1 in the second sequence) and A B D (at positions 1 and 4 in the third sequence). A shorter minimum motif length may yield many more motifs, some of which might be sub-sequences of other longer motifs, but these shorter results are also likely to be less informative and less distinctive.

Several research fields have developed algorithms and tools for detecting and extracting motifs within sequences. However, most of these computational tools have certain limitations that make them unsuitable for detecting patterns in sequence data with categorical codes. I will briefly outline some of these tools and the reasons why they are not appropriate for the purpose

of this study. I will then describe the Teiresias algorithm in detail, and explain why I think it is the best tool for finding motifs in sequences.

Many programs have been developed (HCIL, 2002) for finding patterns in time series data—numeric values that change over time (e.g., a daily temperature reading for some geographical location). However, these tools rely on the quantitative nature of the underlying sequence elements and are thus not useful for categorical code sequences.

Although the bio-informatics literature on finding motifs in DNA and protein sequences is extensive (Gusfield, 1997; Altschul et al., 1997), most of the algorithms look for pre-specified patterns within extremely long sequences of gene or protein information. These methods are very fast and efficient, but because they require a pattern to be specified a priori, they are not useful as general tools for detecting all possible motifs in a sequence.

Perhaps the most comprehensive tool for finding probabilistic patterns in categorical sequence data is *Theme* (Magnusson, 2000). It detects time patterns in behavioral data by aggregating consistencies in the occurrence of events. Unfortunately, the program is currently limited by the fact that it finds these patterns within only one sequence, and not across a collection of sequences.

*Teiresias*. Given the limitations of these other tools, the Teiresias algorithm (Rigoutsos, Floratos, Parida, Gao & Platt, 2000) appears to be the fastest and most appropriate method for detecting motifs in multiple categorical sequences. The IBM Bioinformatics Group (2003) has implemented this algorithm as a stand-alone command line tool for the Windows, Linux and AIX platforms, and as a web-based service (Figure 4).

## IBM Bioinformatics Group - Tools & Content

Figure 4. Web interface to the text symbol pattern discovery tool based on the Teiresias algorithm.

The text-symbol pattern discovery tool shown in Figure 4 takes one or more sequences of characters as input—either letters or digits—and gives back a table reporting the patterns, or motifs. If the dataset from which we want to extract motifs consists of sequences of codes that are made up of words or other symbols, they will first have to be transformed into simple character codes. Each character is treated as a potential motif element, and all punctuation marks and spaces are ignored. For example, the following sequences would be treated identically:

```
AB-AB CD, AB
ABABCDA B
```

The output of this tool is a table with each row representing a motif and the locations where it was found, listed in descending order of its frequency. The motifs found will either be literal patterns like “ABC”, or patterns with wild-cards like “A.BC”. For example, the sequences DACBC and ADBCE only have BC as a literal motif, but if we allow wildcards in the motif

specification, and a “.” can stand for any literal character, then both sequences can be said to contain the motif “A.BC”.

The kinds of motifs found will depend on the following parameters (see Figure 4 for how these parameters are entered):

- L: the minimum number of literals (i.e., non-wild-card characters) in a motif.
- W: the maximum extent spanned by any L consecutive literals. That is, if W is the same as L then no wildcards are allowed, but if W is greater than L by two, then up to two wildcards will be considered when discovering motifs.
- K: the minimum number of times a pattern must occur before it can be reported as a motif. This is a very useful parameter in large data sets because if a pattern only occurs twice in a thousand sequences, it is probably not meaningful.
- Q: the maximum number of times a pattern can occur. The largest possible value is the default and usually there is no reason to change this value.
- Seq-version: if this option is set to “True” (the box is checked), then motifs will be reported if a pattern occurs in at least K different sequences. If this option is set to “False”, a motif is reported if it occurs at least K times, even if all the instances are within the same sequence.

For example, the output for the sequences and parameters entered in Figure 4 is:

```
5      3      ABC 0 2 0 9 1 2 1 8 2 4
4      2      BAB 0 1 0 8 2 1 2 3
3      2      BABC 0 1 0 8 2 3
3      2      ABAB 0 0 2 0 2 2
2      2      ABABC 0 0 2 2
```

Each line of the result reports a motif that was found (the third column). The first column indicates the number of times this motif occurred over all, the second column tells us how many different sequences this pattern occurred in, and the numbers after the pattern report the locations of the patterns. The numbers have to be interpreted in pairs and have the added quirk that all

counts begin from 0. Thus, “0 2” in the first row means that the motif ABC occurred in the *first* sequence at the *third* location.

Perhaps the greatest advantage of the Teiresias algorithm is its speed and its guarantee that the patterns reported are the longest possible. For example, if ABC is reported as a pattern that occurs 10 times, its sub-sequence AB will not be reported unless it occurs more often than ABC.

Once we have found the motifs in a given set of sequences, we can use them in many ways—to find the most commonly occurring motifs, to find out if certain groups of sequences are more likely than others to have certain motifs, to see which motifs are the longest, to determine which motifs occur more often towards the start of a sequence and which towards the end, and so on.

One of the major issues in using a motif detection algorithm such as TEIRESIAS is that it may find too many motifs once we start using wildcards. Wildcards will almost always have to be allowed while exploring any data set because observations in the real world tend to be messy and even scripted activities are likely to be occasionally interrupted by other events. We can avoid having to sift through hundreds of patterns by increasing the minimum motif length, at the risk of missing possibly significant shorter motifs.

If the objective is to study groups of sequences and find out which motifs characterize each group, then the process of finding significant motifs can be automated to some extent. Each pattern found can be treated as a variable, and we can then use discriminant analysis techniques to see which patterns contribute the most to differentiating between groups.

Motif detection can be a powerful tool for detecting regularities in any sequential data. Algorithms such as Teiresias can help find hundreds of motifs, even in relatively short

sequences, depending on the parameters used. It is up to the researchers to bring their own knowledge of the domain and the data to the process when deciding what the minimum motif length should be, how many wildcards to allow, and what the minimum number of occurrences should be.

### *Dynamic Time Warping*

Before describing the dynamic time warping algorithm in detail, I will introduce the concept of distances between sequences and why it is useful.

When we compare two sequences to see how similar or dissimilar they are, we are really asking the question “Are these two sequences the same, and, if so, what accounts for the differences between the two?” This question can be answered by considering the following ways in which two nominally similar sequences can differ from each other:

1. substitutions (also called replacements),
2. deletions and insertions (also called indels),
3. compressions and expansions,
4. transpositions (also called swaps).

Each of these operations is a way to transform one sequence into another. If many such operations are needed to perform the transformation, the two sequences are said to be more distant, or more dissimilar; if no or very few such operations are needed, then the sequences are said to be less distant, or very similar.

Algorithms that find the distance between sequences are better than traditional statistical analyses for comparing how similar sequences are. Unlike monadic variables such as the height or weight of a subject, a score on a test, or the amount of time spent on some activity, distances are dyadic variables that take into account the interaction between the two things being studied.

Monadic variables are extremely useful indicators of many phenomena, and are the cornerstones of traditional statistical analyses, but they provide little or no information when it comes to comparing a large number of sequences of data.

An example of this way of looking at sequence comparisons is the concept of an “edit distance” (Levenshtein, 1966)—a measure of the number of insertions, deletions and replacements of individual elements necessary to transform one sequence into another. If two sequences are identical, their edit distance will be zero because no insertions, deletions or replacements are necessary. In the following example, the two sequences are not identical:

1. A B B A B C D
2. A B A B C E

The edit distance for these two sequences is 2 because one deletion (B) and one replacement (E for D) will transform the first sequence into the second sequence. The edit distance is a symmetrical measure because the distance from the first sequence to the second is the same as the distance from the second sequence to the first.

Edit distances are a good way to compare sequences and are popular tools for detecting misspellings and typographical errors. However, they do not account for compressions and expansions, which are among the main sources of variation in sequences obtained by time-sampling episodes. Episodes of social interactions and other naturalistic phenomena almost always exhibit time compressions and expansions because the natural world does not operate according to a precise time clock.

The following example compares two time-segmented sequences where each code represents the type of event occurring in a unit of time.

1. A A A A B B B B B B B B B B B C C C A A A A A B B
2. A A B B B B C C C C C C C C C C C C A A B

The Levenshtein (1966) edit distance for these two sequences is 14. However, if we assume that compressions and expansions of a code should not be counted as differences, then the distance between these two sequences is 0, and the sequences are identical.

Kruskal and Liberman (1983) describe the dynamic time warping (DTW) algorithm which is a perfect example of a method that does not increase the distance between two sequences for compressions and expansions of an event. At the same time, this method accounts for insertions, deletions and replacements—making it an ideal choice for determining the similarities between two category coded sequences that have been obtained by recording the events in an episode over time.

The DTW distance between two sequences is essentially the same as a Levenshtein edit distance with the important exception that consecutive repetitions of a code are treated as one instance of the code, provided a match for that code is found in the other sequence. If we imagine the duration of each event in a sequence as being elastic, the DTW algorithm finds the smallest number of changes that would transform one sequence into another while stretching or shrinking each code to find the best match possible. The following examples will make this clearer.

1. A A A A A A A A A B C D
2. A B C D

The DTW distance is 0 between the previous two sequences because the run of A's is treated as a single A when matching it to the A in the second sequence.

1. A A A A A B C D
2. E B C D

In the previous case, the DTW distance is 5 because the E needs to be changed to A, and four more A's have to be added to match the two sequences. The algorithm does not treat the repeating A's as one A because there is no A that matches it in the second sequence.

1. A B A A A A A B C D
2. A B B B B C D

In the preceding two sequences, the DTW distance is just 1, because deleting the first B in the first sequence is the smallest change necessary to transform it into the second, because the remaining A's in the first sequence and the remaining B's in the second sequence can be matched with their counterparts.

The details of the DTW algorithm and its various implementations are beyond the scope of this paper, but it provides the best way to compare two sequences that have been obtained from coding episodes in the real world which often have variations in the duration of each event.

Finding the DTW distances between every pair of a large number of long sequences is computationally expensive—the time taken increases at least exponentially as the number of sequences increases—but it is well worth the expense as it gives us a basis for determining clusters of similar sequences. If we have  $n$  sequences, the  $n$  by  $n$  distance matrix created by finding every possible DTW distance can be used to a) create a visual map of the sequences using multidimensional scaling techniques, and b) do a cluster analysis on the sequences for determining an inherent grouping in the sequences.

In a later part of this paper, I will describe how I used the DTW algorithm to find distances between the lesson event sequences in the TIMSS data, and used the resulting distance matrix to discover consistencies and dissimilarities in the sequences in a way that would not have been possible without these kinds of techniques.

## *Hidden Markov Models*

The previous two techniques—motif detection and dynamic time warping distances—were based on deterministic algorithms for analyzing sequences. Hidden Markov models (HMMs) are a probabilistic approach to the problem of determining regularities in sequential data.

HMMs are immensely popular tools for analyzing sequences in domains ranging from speech recognition to protein sequencing. For the purposes of this paper, I will begin with a short overview of how the models represent events in the real world using the classic ‘rainy day-umbrella’ example (adapted from Hausler, 2004). I will then list the kinds of questions that HMMs can help us answer about sequences. Although I will not go into details of the algorithms that create HMMs, I will describe what is needed to start using this technique. At the end, I will summarize the advantages and limitations of using HMMs to find patterns in sequences of behavioral data.

In an HMM, any observed sequence of events is assumed to have been generated by a Markov process—each event is dependent to some extent on what happened on the previous step. For example, if a certain town has only three kinds of weather—sunny, rainy or foggy—then the weather on each day is somewhat dependent on what happened the day before. For instance, given that yesterday was a rainy day, there is a greater likelihood that today will also be rainy, although there is a chance that it may be a sunny day. A basic Markov process—also known as a first-order Markov process—is a simplified representation of reality because it states that the probability of an event is only dependent on the event that occurred on the previous time step, and is not affected by events that happened two or more steps previously. Most events in the real world do depend on what happened two or more steps in the past, but the assumption of

a Markov process simplifies the problem and gives very good results, so we stick with this assumption.

The sequence of states in a Markov process can be observed directly, or it can be observed indirectly by recording the events that occur for each state. Continuing the example from the previous paragraph—consider a man who is locked up in a windowless prison in that same town who cannot tell the weather by direct observation. If we assume that the only way that he can guess the state of the weather is by seeing if the warden brought in an umbrella, then the prisoner is viewing an observable event (presence or absence of an umbrella) that was generated by some unobservable states (sunny, rainy, or foggy weather).

A hidden Markov model consists of a finite set of states (sunny, rainy, or foggy, in our example) and a finite set of observable events (umbrella present or umbrella absent). Each state can only cause one of many possible events to occur at a time. The word ‘hidden’ refers to the fact that the states may be unobservable for some reason. The model also consists of:

1. An initial state probability vector that stores the probability of each state being the starting point of the sequence. In our example, the vector can store any probability values such as Sunny = 1.0, Rainy = 0, and Foggy = 0.
2. A state transition matrix (Table 1) that stores the probability of going from one state to another. For example, the first row gives the probability of a sunny day following a sunny day, a rainy day following a sunny day, a foggy day following a sunny day, and so on.
3. An observation matrix (Table 2) that stores the probability of an observable event occurring, given that the system is in a certain (hidden) state. For example, the first row shows how likely is it that the warden is carrying an umbrella or not, when it is sunny.

Table 1

*State Transition Matrix for a Hidden Markov Model*

State	Sunny	Rainy	Foggy
Sunny	0.7	0.1	0.2
Rainy	0.3	0.5	0.2
Foggy	0.4	0.4	0.2

Table 2

*Observation Matrix for a Hidden Markov Model*

State	Observation	
	Carrying umbrella	Not carrying umbrella
Sunny	0.1	0.9
Rainy	0.8	0.2
Foggy	0.4	0.6

The model described here may be true for a certain season (e.g., summer), and a different model may be needed to describe another season. Given these models, there are three types of problems that can be solved if we are given a sequence of observed events:

1. Match the most likely underlying model out of a set of models. In our example, this would be analogous to the prisoner determining whether it was winter or summer, given a sequence of events recording the presence or absence of the warden's umbrella.
2. Determine the sequence of underlying or hidden states. For example, the prisoner could tell what the weather was like outside for a week after observing the warden's umbrella habits.

3. Determine the model parameters (initial state probability vector, state transition matrix, and observation matrix), also known as training the HMM.

The first step in using an HMM for studying categorical sequence data is to train the HMM using a set of observed sequences. Algorithms for training HMMs have been implemented in a variety of programming environments and are freely available online (Cambridge University Engineering Department, 2003; Murphy, 2003; Schliep, Rungtarityotin, & Georgi, 2003). Typically, a data set will consist of several sequences, all or some of which can be used to train an HMM. If the goal is to understand the underlying processes, then the entire set of sequences should be used. If the goal is to detect similarities between groups of sequences then a random set of sequences from each group should be used. To train an HMM, we need to know the number of observation events possible (i.e., the distinct sequence elements), and we need a good guess for the number of hidden states. Discovering the number of states is not an easy task as there may be two states or twenty. One approach is to train the model using several values and then test the likelihood of the model against the same sequences. The value that results in the highest likelihood should be used.

Even if we can make a good guess for the number of states, we may not know what the states represent. We can often intuit the meaning of a state by checking the probabilities of observable events associated with it in the observation matrix. This process also lets us discover the underlying structure of a set of sequences that look different on the surface. Long and complex sequences can sometimes be characterized by simple state flow diagrams (made from the state transition matrix) that provide new insights into the process being studied.

HMMs are also very useful for determining the group a sequence might belong to. Griffin (2003) describes how he used HMMs to correctly classify distressed and non-distressed marital relationships based on the affect sequences generated during a conversation between spouses. If

a group of sequences are all known to be from the same source then we can train an HMM on that group. Subsequently, when a new sequence is encountered, we can test it against the model to measure the likelihood that the new sequence is from that same group. Classifying new sequences this way might fail if there is too much variance in the sequence patterns in the original group that was used to train the HMM. If an HMM is not able to classify group members correctly, then it is very likely that there were very few sequential consistencies in the group to begin with.

HMMs are not useful when the goal is to detect clusters or groupings in a set of sequences. This is because the training of an HMM requires an a priori grouping of sequences, which is the very thing that we were trying to determine.

### Summary of Methods

The three methods presented in this section address three very different ways of looking at temporal sequences of categorical data and finding patterns in them. The best method will depend on the kind of questions that one hopes to answer.

I began by describing the ways in which sequences can be created from time-coded episodes—fundamental sequences, time-segmented sequences, and proportion-segmented sequences. If we are interested only in the order of events, fundamental sequences are the best way to go. The other two ways have the advantage of encoding duration information (a code that lasts twice as long as another will be represented using twice as many elements in the sequence) but might lose all information about an event if the resolution of the segments is too coarse. In addition, the latter two types of sequences also lose information because they make no distinctions between a single long event and many successive repetitions of the same event.

Both the motif extraction and DTW distance methods allow for a bottom-up approach to finding patterns. Sequences with similar motifs, or sequences with low DTW distances between them, can be clustered closer together and these groupings can be a way to further characterize the dataset. HMMs, on the other hand, are not helpful for determining groupings of the sequences in a dataset. However, given an explicit a priori grouping, an HMM can be trained for each group to succinctly represent the sequential information for that group.

Groups of sequences can also be characterized and used to classify new sequences using the techniques I have described. Motif extraction can be followed by discriminant analysis to determine the motifs that characterize a group of sequences. Similarly, HMMs can be trained on a group of sequences, and new sequences can be tested against different models to classify them.

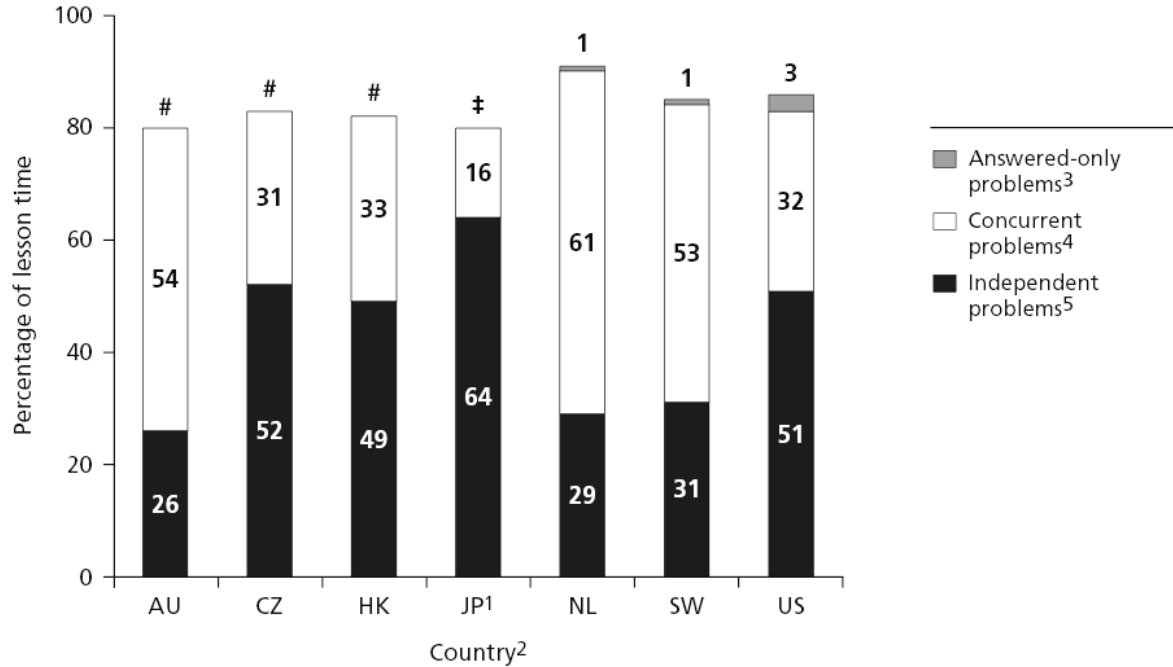
Other methods for analyzing sequences also exist, but these three were chosen because of their potential benefit to researchers in the social sciences who work with sequential data. All three methods are well known in fields like bio-informatics, speech recognition, and perceptual psychology, but I have found only one paper that applies any of these techniques to the analysis of behavioral interactions (Griffin, 2003). I believe that the increased use of these techniques can help us make sense of complex sequential data at a qualitatively different level.

## CHAPTER 3

### APPLICATIONS TO THE TIMSS 1999 VIDEO STUDY

The TIMSS 1999 video study (Hiebert, Gallimore, Garnier, Givvin, Hollingsworth, Jacobs, Chui, Wearne, Smith, Kersting, Manaster, Tseng, Etterbeek, Manaster, Gonzales & Stigler, 2003) provided a close look at teaching practices around the world. The study created and analyzed video records of more than six hundred eighth grade mathematics lessons from national probability samples in seven countries—Australia, the Czech Republic, Hong Kong SAR, Japan, the Netherlands, Switzerland, and the United States—which were then coded on several criteria.

An important aspect of the findings by Hiebert et al. (2003) was the amount of time spent on different types of events in the lessons. Using video records of classrooms, the researchers were able to precisely code the times that specific events happened and analyze the prevalence of these events. Each lesson was coded along several dimensions – the purpose of the teaching activity, the type of classroom interaction (e.g., public, or private, or a combination), and the type of activity (e.g., a problem for the students to work on independently, a non mathematical activity, a problem that was simply answered without any explanation, etc.). By analyzing the average amount of time spent on these activities in each country, they found several interesting differences between the countries participating in the study. For example, the average amount of time spent doing independent problems was highest in Japan and in the US (Figure 5), and the Japanese lessons also had the highest average amount of time spent on each individual problem (Figure 6). The Czech and US lessons spent more time reviewing material compared to the other countries (Figure 7).



#Rounds to zero.

#Reporting standards not met. Too few cases to be reported.

<sup>1</sup>Japanese mathematics data were collected in 1995.

<sup>2</sup>AU=Australia; CZ=Czech Republic; HK=Hong Kong SAR; JP=Japan; NL=Netherlands; SW=Switzerland; and US=United States.

<sup>3</sup>Answered-only problems: US>AU, CZ, HK.

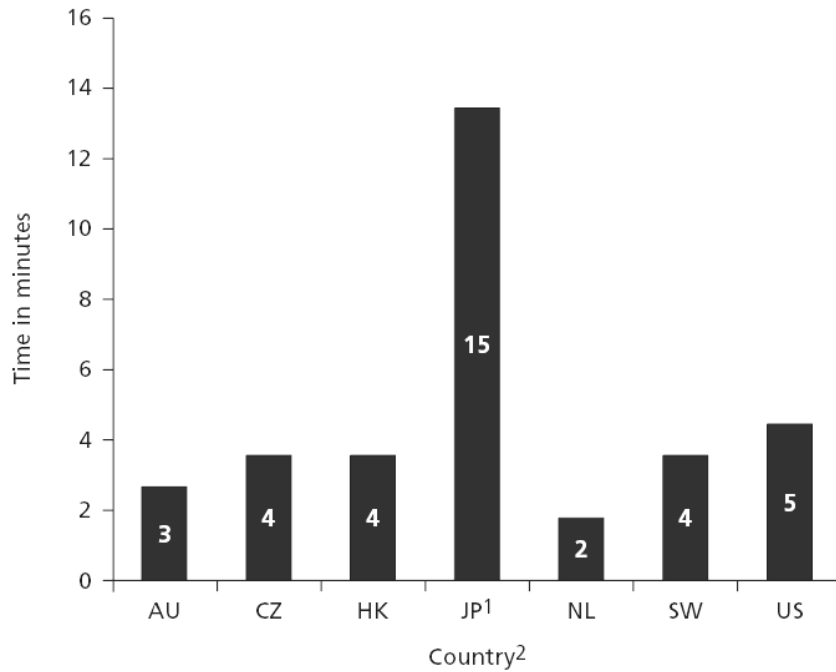
<sup>4</sup>Concurrent problems: AU, NL, SW>CZ, HK, JP, US.

<sup>5</sup>Independent problems: CZ, HK, JP, US>AU, NL, SW.

NOTE: Independent problems were presented as single problems and worked on for a clearly definable period of time. Answered-only problems had already been completed prior to the lesson, and only their answers were shared. Concurrent problems were presented as a set of problems to be worked on privately. For each country, average percentage was calculated as the sum of the percentage within each lesson, divided by the number of lessons. Percentages sum to average percentage of lesson time devoted to problem segments per country (see figure 3.3), although in some cases they do not because of rounding or data not reported.

SOURCE: U.S. Department of Education, National Center for Education Statistics, Third International Mathematics and Science Study (TIMSS), Video Study, 1999.

*Figure 5. Average percentage of eighth grade mathematics lesson time devoted to independent problems, concurrent problems, and answered-only problems, by country:1999 (from TIMSS Report).*



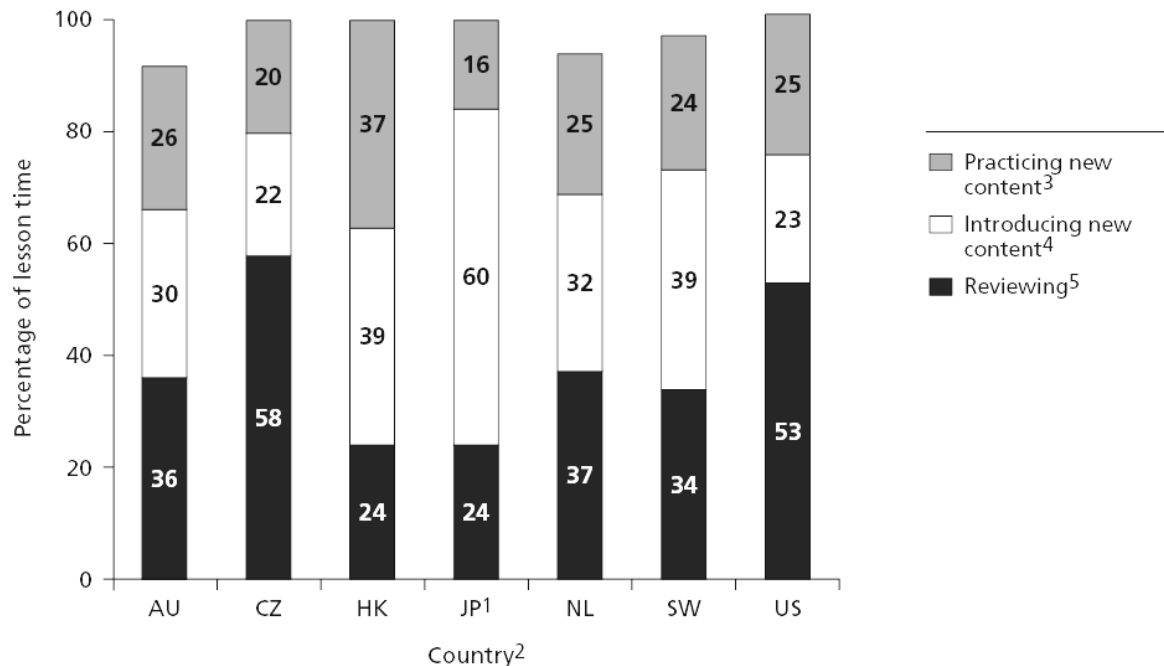
<sup>1</sup>Japanese mathematics data were collected in 1995.

<sup>2</sup>AU=Australia; CZ=Czech Republic; HK=Hong Kong SAR; JP=Japan; NL=Netherlands; SW=Switzerland; and US=United States.

NOTE: CZ, HK, SW>NL; JP>AU, CZ, HK, NL, SW, US. The tests for significance take into account the standard error for the reported differences. Thus, a difference between averages of two countries may be significant while the same difference between two other countries may not be significant.

SOURCE: U.S. Department of Education, National Center for Education Statistics, Third International Mathematics and Science Study (TIMSS), Video Study, 1999.

*Figure 6.* Average time per independent problem per eighth-grade mathematics lesson (in minutes), by country: 1999.



<sup>1</sup>Japanese mathematics data were collected in 1995.

<sup>2</sup>AU=Australia; CZ=Czech Republic; HK=Hong Kong SAR; JP=Japan; NL=Netherlands; SW=Switzerland; and US=United States.

<sup>3</sup>Practicing new content: HK>CZ, JP, SW.

<sup>4</sup>Introducing new content: HK, SW>CZ, US; JP>AU, CZ, HK, NL, SW, US.

<sup>5</sup>Reviewing: CZ>AU, HK, JP, NL, SW; US>HK, JP.

NOTE: For each country, average percentage was calculated as the sum of the percentage within each lesson, divided by the number of lessons. Percentages may not sum to 100 because of rounding and the possibility of coding portions of lessons as “not able to make a judgment about the purpose.”

SOURCE: U.S. Department of Education, National Center for Education Statistics, Third International Mathematics and Science Study (TIMSS), Video Study, 1999.

*Figure 7. Average percentage of eighth-grade mathematics lesson time devoted to various purposes, by country: 1999.*

These types of results highlighted some key differences among the countries and reinforced the findings from the previous TIMSS studies (Stigler & Hiebert, 1999) that teaching is a cultural activity that is carried out differently in different countries.

As I pointed out in the first part of this paper, prevalence analysis tells only a part of the story, especially when it is used for observations of behavioral interactions that take place over time. Givvin, Jacobs and Hollingsworth (2003) describe a simple construct known as a “lesson signature” that takes into account some of the temporal information present in the TIMSS data. A lesson signature (an extract is shown in Figure 8) can be created for each country by first

dividing the duration of each lesson within that country into 100 parts; a graph is then created for each code (event type) that reports the percentage of lessons that exhibit that code at each time point in the lesson.

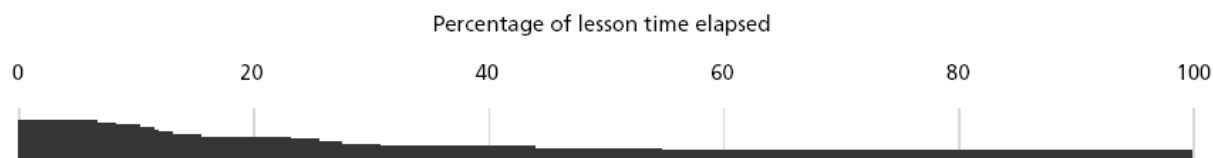


Figure 8. Lesson signature for the purpose code “Review”, for all Australian lessons.

In Figure 8, the horizontal axis shows the duration of the lesson, expressed in percentage points. The vertical axis shows the proportion of lessons exhibiting a specific code (“Review”, in this case) at that point of the lesson. Thus, at the very start of the graph (when 0% of the lesson has elapsed), the curve is at approximately 80%, which means that 80% of the Australian lessons were engaged in review at the very beginning of the lesson. This percentage drops off towards the end of the graph so that only 20% of the lessons are engaged in review towards the end of a class.

Although this representation captures some temporal information, the aggregation of information from several lessons into one data point prevents us from knowing the pattern of events within individual lessons. For example, we cannot tell if any of the 20% of lessons that do “Review” at the end of the class time are the same as the ones that were engaged in review at the beginning of the class.

From the examples presented so far, it is evident that our knowledge of the sequence of events in the TIMSS video study lessons is limited. In this thesis, my goal is to extend the research presented in *Teaching Mathematics in Seven Countries* by finding patterns in the sequences of events in the lessons rather than just analyzing the amount of time spent on a

particular type of event by country. I hope to contribute to a comprehensive account of the TIMSS data that takes into account the sequence of events in a classroom as well.

To achieve this goal, I will first describe the TIMSS video study dataset. After briefly describing the codes used and the way they are assigned to events in the classrooms, I will use the ideas presented in the first part of this paper to show how we can generate sequences that will accurately represent one or more dimensions of the observations.

In the next and most important part of this section, I will apply the three methods of motif extraction, dynamic time warping, and hidden Markov models to the sequences generated from the lesson data. The basic results obtained from these analyses will be further processed using techniques such as discriminant analysis (to determine the most ‘interesting’ motifs), multidimensional scaling (to draw a visual map of the lesson sequences and see which ones are most similar or dissimilar), and cluster analysis (to see if the lessons within a country are similar).

### Creating Lesson Sequences From the TIMSS Dataset

The TIMSS 1999 video study recorded 638 lessons from seven countries. Figure 9 shows the number of lessons recorded in each country and its average length (in minutes).

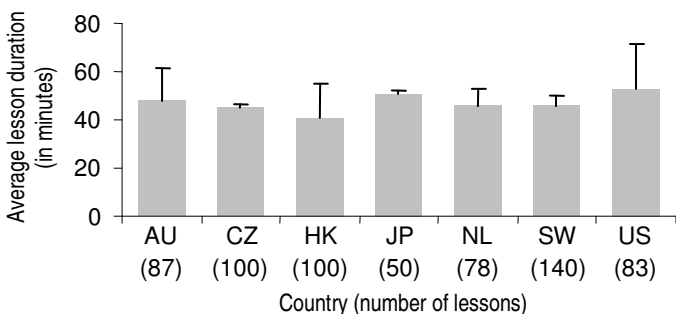


Figure 9. Number of lessons in each country and average lesson length (in minutes) in the TIMSS 1999 video study.

Each lesson in the TIMSS dataset was coded using three sets of codes that captured three dimensions of the events in the classrooms. The first set of codes described the purpose of the lesson at different points of time; the second set described the kind of classroom interaction taking place—for example, whether it was public or private or some combination; and the third set described the type of content activity taking place—for example, a problem to be worked on independently, a mathematical organizational activity such as handing out worksheets, and so on. The following descriptions of the codes are adapted from the TIMSS 1999 video coding manual (LessonLab, 2003). The manual also describes in detail the guidelines followed while coding the lessons, how exceptions were handled, and the inter-rater reliability of the coding process.

Each of the three sets of codes “covered” the lesson—the codes within the set described every part of the lesson, there were distinct starting and ending times for each type of event or activity, and no code overlapped another code within the same set.

The three purpose codes used were:

1. Review (P1) – Addressing content used in previous lessons.
2. New material (P2) – Introducing new content.
3. Practicing new material (P3) – Practicing/applying/consolidating content introduced in the current lesson.

Only one of these three codes could be applied for any given time period, although a code could occur more than once. For example, a lesson could be coded as P1 from 0 min to 11 min, P2 from 11 min to 32 min, P3 from 32 min to 38 min, and P2 again from 38 min to 43 min.

The five classroom interaction codes used were:

1. Public interaction (CI1) – All students participate or listen to a public dialog directed by the teacher or one of the students.
2. Optional, teacher presents information (CI2) – The teacher directs the public dialog, but clearly indicates that the students don’t have to pay attention (e.g., they may

continue to work on their assignment without listening to the teacher). This type of interaction is very infrequent.

3. Optional, student presents information (CI3) – A student directs the public dialog (e.g., works out a problem at the board) and other students listen, but it is optional for them to do so.
4. Mixed interaction, where public and private interaction co-occur (CI4) – The teacher divides the class into groups. Some students are assigned to work privately on problems, while the rest of the class works publicly with the teacher. This pattern is rare.
5. Private interaction (CI5) – The students work on their own or with each other in small groups.

As with the purpose codes, the classroom interaction codes can also occur more than once and two such codes cannot be assigned to the same time period.

The nine content activity codes used were:

1. Non-math (NM) – Non-mathematical or off-topic segments that offer no opportunities for the students to learn math.
2. Math organization (MO) – References are made to mathematics (e.g., math tools, homework, tests, other resources) but no actual content is discussed.
3. Answered-only problem (AO) – These problems have been previously discussed in the class and are not worked on during the present lesson. Answers are shared either verbally or in written form and there is no public discussion of a solution strategy.
4. Independent problem (IP) – Problems where all students spend a known amount of time working by themselves to solve them (i.e., it is clear how much time is spent on each problem)
5. Concurrent problem set-up (CPSU) – Concurrent problems are those where the students spend some private time working on multiple problems, and the exact time spent on each problem is unknown. In the set-up phase, the teacher assigns multiple problems for the students to work on.
6. Concurrent problem seatwork (CPSW) – Students work privately (individually, in pairs, or in small groups) on concurrent problems.
7. Concurrent problem classwork (CPCW) – Students and the teacher actively work on or discuss the concurrent problems publicly as a whole class.

8. Concurrent problem mixed interaction (CPM) – Students solve concurrent problems and are divided into different groups, some of which are working privately and others publicly with the teacher.
9. Non-problem (NP) – Segments that contain mathematical information that is not an explicit problem although it may reference a problem.

The content activity codes also could occur more than once in a lesson, and did not overlap with each other. The key difference between the content activity codes and the previous two types of codes is that the same content activity code can be applied to consecutive parts of a lesson. This will be explained in detail later when I describe how lesson sequences can be generated from the time-coded data.

As the three dimensions of codes are measuring different things, each time point in the lesson can be described using one code from each set. For example, the start of a lesson may be described using the triple P1-CI1-MO—that is, the purpose of the lesson at that point was to review previously learned material, the teachers and students were interacting publicly and addressing the whole class, and the teacher was organizing the math activity but not actually teaching any mathematical content.

We can use as many or as few dimensions of codes as we want to describe a specific point in the lesson. Figure 10 shows a lesson that lasts 40 minutes. The starting and ending times of the codes from all three code sets are shown, and the 7 different ways of describing a specific time point (indicated with a dashed line) in the lesson are described.

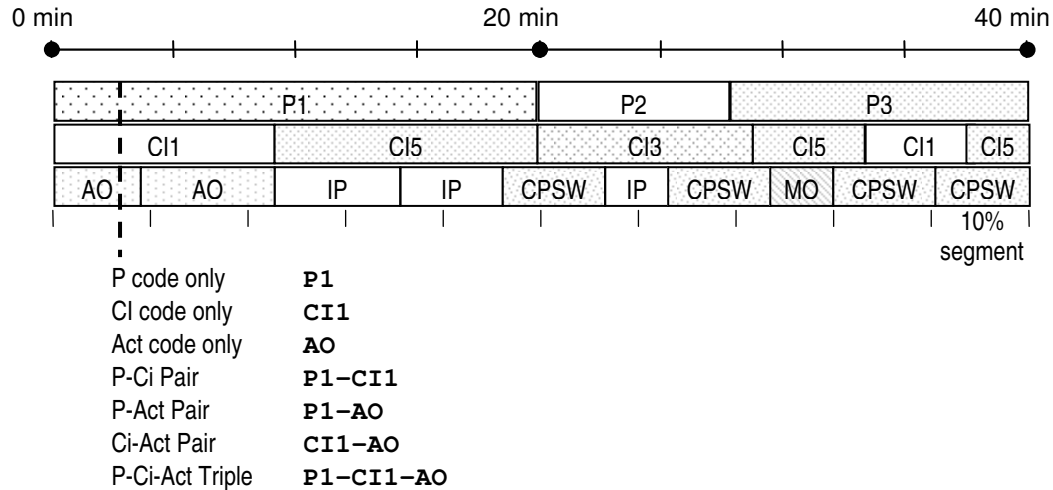


Figure 10. Describing a time point in a lesson using single codes, code pairs, and code triples.

In the first part of the paper, I described three ways of deriving sequences from time-coded event data. The following examples show how these different sequences can be obtained from the lesson in Figure 10.

1. Fundamental sequence (FS), based on Purpose (P) codes alone:  
P1, P2, P3
2. FS based on pairs of P codes and Classroom Interaction (CI) codes:  
P1-CI1, P1-CI5, P2-CI3, P2-CI5, P3-CI3, P3-CI5, P3-CI1, P3-CI5

Note that even if one of the codes in a code pair changes (e.g., the change from P1 CI1 to P1 CI5), the sequence element changes, and so a new element must be added to the sequence.

3. FS based on Activity (Act) codes alone:  
AO, AO, IP, IP, CPSW, IP, CPSW, MO, CPSW, CPSW

Activity codes are the only ones that show repetitions within a FS, because two activities of the same type can occur consecutively. For example, the ‘answered only’ code (AO) can be assigned to a lesson for the first two minutes, and then again for the next two minutes, because two separate ‘answered only’ problems may have occurred. The P and CI codes on their own can never repeat in a FS because they do not describe a type of activity that is repeatable.

4. Time-segmented sequence (TSS) based on 5 minute segments of CI codes alone:  
CI1, CI1, CI5, CI5, CI3, CI3, CI5, CI1

Because of the low resolution of the segments (5 minutes), the initial CI1, CI5 and CI3 events look the same in the TSS as each code is repeated twice, even though their original durations were different. The last CI1 code is also dropped because the

preceding CI5 code dominates the last 5 minute segment of the lesson. This example shows how important it is to choose the right sized time unit for a TSS.

5. Proportion-segmented sequence (PSS) based on 10% segments of Activity (Act) codes alone:

AO, AO, IP, IP, IP, CPSW, CPSW, MO, CPSW, CPSW

As in the TSS, the low resolution of the segment size causes the sequence to leave out a few events. This problem can be fixed by choosing a finer segment size such as 1% or 2%.

To summarize, as many as 7 different code sequences can be created in 3 different ways for a TIMSS lesson depending on whether we wish to choose a single code, a pair of codes, or a code triple as our sequence element. When I analyze sequences in the next part, I will begin with the fundamental sequences of single codes—the simplest possible sequence representation of a lesson—and will subsequently analyze more complex sequences such as a proportion-segmented sequence made up of code triples.

### Results of Three Sequence Analysis Methods

#### *Motif Extraction in TIMSS Lesson Sequences*

The process of finding motifs in the TIMSS lesson sequences involved several steps. The first step was to choose the type of lesson sequence from which the motifs would be extracted. The time-segmented and proportion-segmented sequences were not appropriate in this case because they are sensitive to the segment size and might ignore certain types of codes. To get a true picture of the order in which certain events happened, I looked for motifs in the fundamental sequences. I applied the Teiresias algorithm for motif extraction—described in the first part of this paper—to fundamental sequences based on Purpose (P) codes, Classroom Interaction (CI) codes, and Activity (Act) codes. I padded each lesson sequence with a “B” at the beginning, and

an “E” at the end, so that the results would also include information about characteristic motifs that occurred at the start or the end of a lesson.

The second step was to input the sequences into IBM’s online text-symbol pattern discovery tool (see Figure 4 for a screenshot of the interface) and choose the appropriate parameters as described in the first part of the paper. The IBM tool can work with individual characters or words as sequence. Because the codes were words such as “P1” or “CI2” in our case, I chose the text-word pattern discovery option. I then set  $L = 2$  and  $W = 2$  to find all motifs with two or more sequence elements in them. By setting  $L$  and  $W$  to the same value, I chose not to allow wildcards in the patterns that were discovered, so that the number of patterns would be limited and the patterns would be easier to interpret. Finally, I set  $K = 6$  and  $\text{Seq-version} = \text{True}$  to ensure that a pattern occurred in at least 1% of all 638 lessons before it was counted as a motif.

The third step was to process the results to see how often a pattern occurred, and in what lessons it occurred. The overall goal was to see whether certain motifs of code sequences occur more in some countries than others, therefore this analysis was done by finding the percentage of lessons within a country that exhibited that motif at least once.

Because of the low values of the minimum pattern length, hundreds of patterns were typically extracted by the Teiresias algorithm. To find the most significant patterns, one final step was needed. Each pattern was treated as a quantitative variable set to 1 if the pattern occurred in a lesson, or 0 if it did not. The STEPDISC procedure in SAS was then utilized to determine the patterns that best discriminated between the 7 countries, using stepwise discriminant analysis. The frequency of the patterns selected this way was irrelevant as the procedure only finds those variables that best classify the lessons as belonging to the correct

countries. Thus, some of the patterns chosen occurred in 90% of all the lessons while others appeared as little as 10% of the time. The variables in the STEPDISC procedure entered or left the model using step-wise selection, with  $P < 0.01$  as the criterion for variables to stay.

Despite this stringent criterion, the STEPDISC procedure chose 13 motifs for classroom interactions and 28 for activity codes. Keeping in mind that the focus of this thesis is on methods for finding patterns in sequences rather than a comprehensive interpretation of every result from the TIMSS data, I will select a few motifs as examples of interesting patterns that tell us something about the lessons that would have otherwise been impossible to find out.

*Motifs in fundamental sequences based on purpose (P) codes alone.* Only 5 motifs remained in the STEPDISC model and are shown in Figure 11. Fundamental sequences based on the P codes are generally very short because the purpose of the overall classroom does not change as often as the other codes.

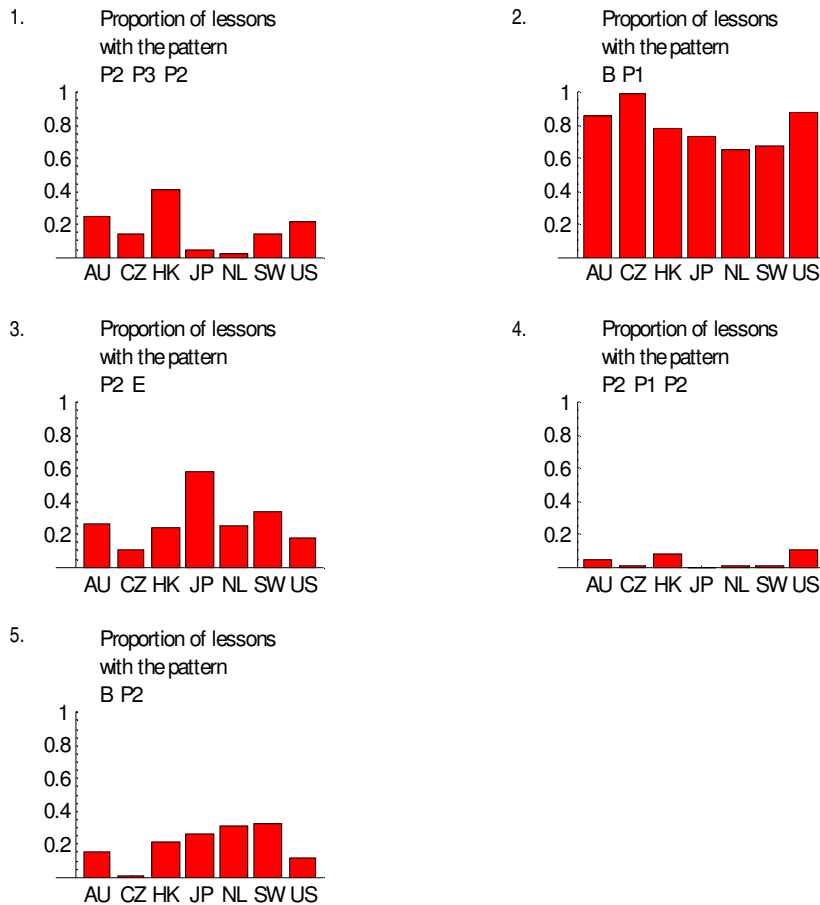


Figure 11. Motifs in fundamental P code sequences, by country.

Over 40% of the Hong Kong lessons introduced new material (P2), practiced the new material (P3) and then went back to introducing further new material in the lesson (P2). Because of the “B” code that padded the start of each lesson sequence, the second motif “B P1” and the fifth motif “B P2” indicate how lessons began. The Czech lessons stand out from the others because almost all of their lessons began with a review. For the third motif “P2 E”, the Japanese lessons show a clear trend towards introducing new material even at the end of a lesson. Only a few lessons in Australia, Hong Kong and the US move from introducing new material back to reviewing older content followed by new material again (motif “P2 P1 P2”). Despite the small number of occurrences of this motif, the STEPDISC procedure selected it for its discriminatory

value as none of the Japanese lessons and almost none of the Czech, Dutch, or Swiss lessons exhibit this pattern.

*Motifs in fundamental sequences based on classroom interaction (CI) codes alone.* Figure 12 shows the first 9 motifs found using the discriminant analysis approach described previously. The first, third, fifth, and seventh patterns are correlated to some extent, but all of them tell a slightly different story. The CI3 code (public information provided by student, optional for other students to pay attention to) occurs mostly in the Czech and Hong Kong lessons, but in Hong Kong, it occurs more often after CI1 (public interaction) than before it.

As the second motif “B CI1” shows, the US lessons are the only ones that sometimes begin with anything other than public interaction. Similarly, the bar graph for the sixth motif “CI1 E” shows how almost all lessons in the Czech Republic, Hong Kong, Japan and Switzerland end with public interaction.

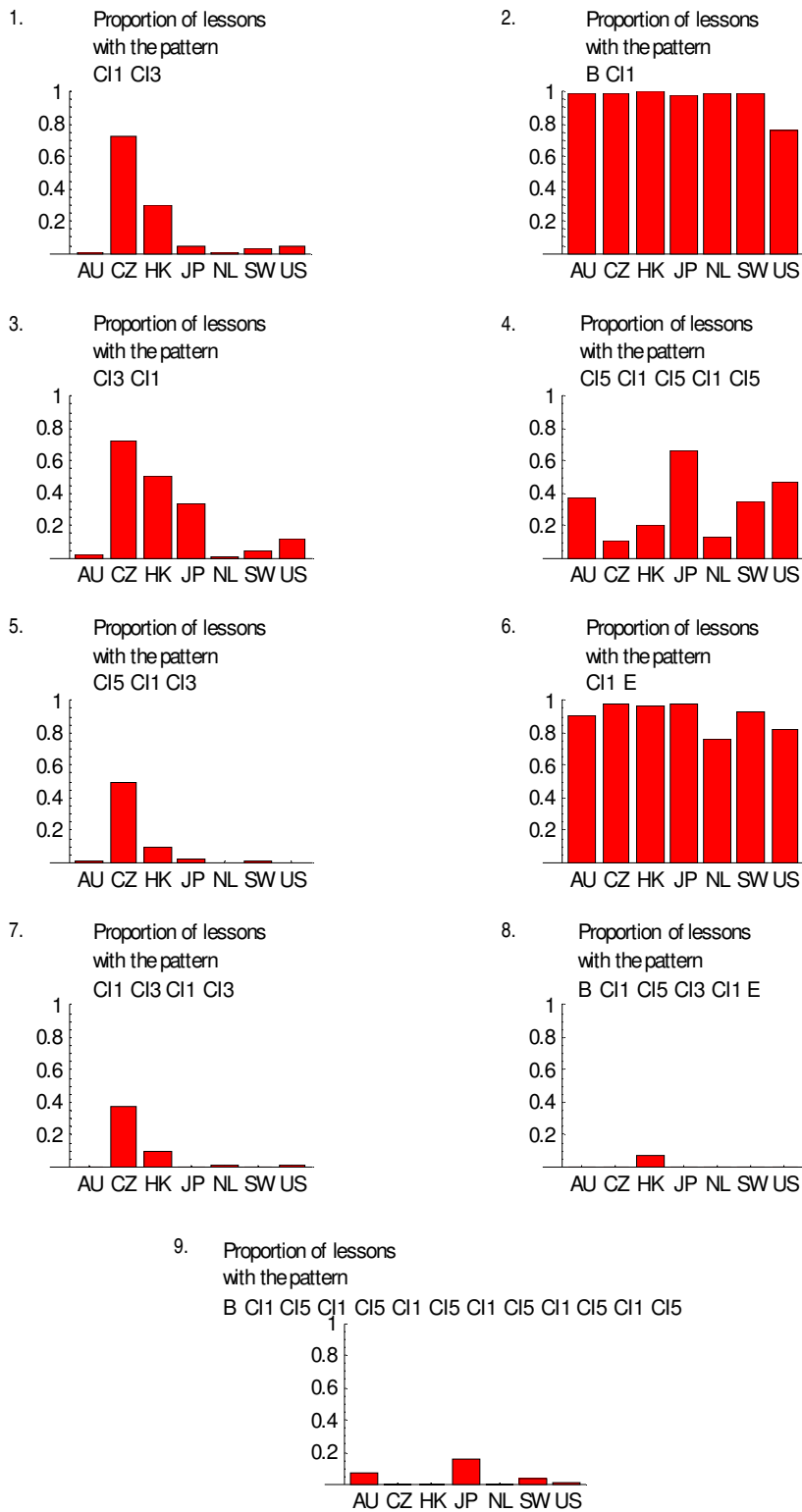


Figure 12. Motifs in fundamental CI code sequences, by country.

The most interesting motifs in this set are the fourth and ninth ones that report the occurrence of alternating public and private interaction. Almost 70% of the Japanese lessons show this alternation, as do approximately 50% of the US lessons and 40% of the Australian lessons. The last motif is equally interesting because it reports that a significant number of Japanese classes switch back and forth between public and private interaction as many as 11 times during a lesson. This kind of information would have been difficult to discover without sequence analysis techniques of this kind.

*Motifs in fundamental sequences based on activity codes alone.* The third and fourth motifs (“CPSU CPSW CPCW” and “B CPSU CPSW CPCW”) are almost identical in Figure 13, with the exception that the latter occurs at the start of a lesson. Despite their similarity, (concurrent problem: set-up, followed by seat work, followed by class work) the fact that the latter motif occurs much less often tells us something about the greater importance of the former motif in the middle of a lesson.

It is also interesting that the US lessons were the only ones that began with CPSW (concurrent problem: seat work) as shown by the second motif.

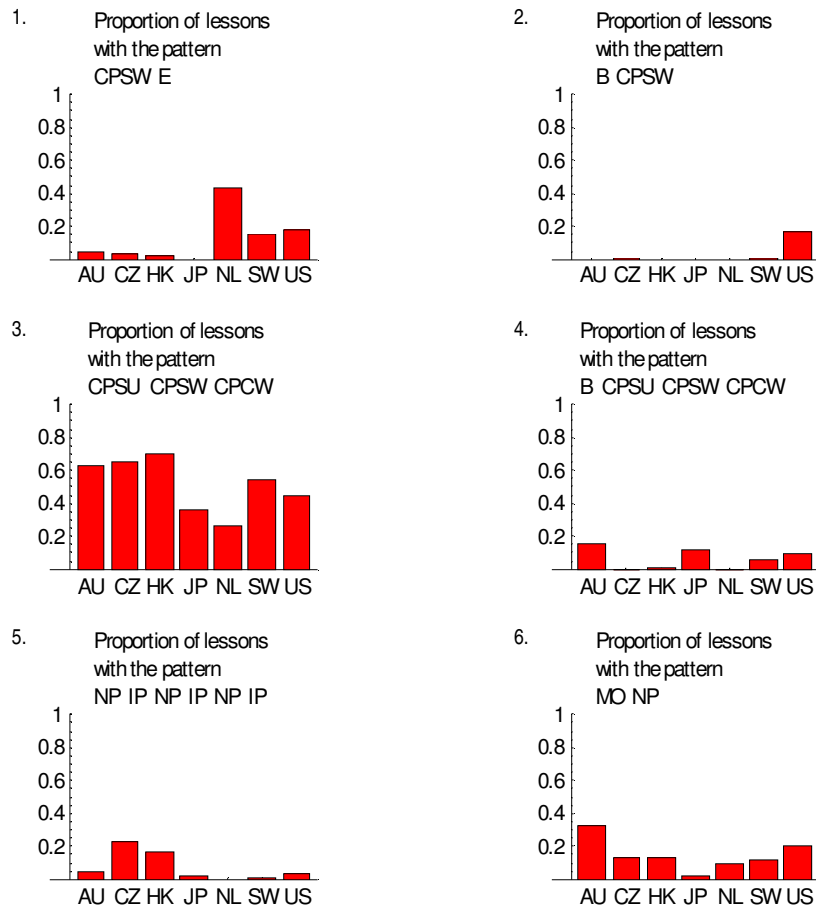


Figure 13. Motifs in fundamental Act code sequences, by country.

*Summary.* The purpose, classroom interaction and activity codes all express some motifs that are more characteristic of some countries as compared to others. For example, the Czech Republic almost always begins with the P1 code (purpose: review), and the Hong Kong lessons are the ones that most exhibit the pattern “P2 P3 P2” where they go on to introduce new material after practicing the previously introduced material.

The observations I have made regarding the motifs found in the TIMSS lessons are meant to be an indication of the kinds of conclusions one can draw with these techniques. Sequence analysis techniques are a useful tool for finding interesting patterns and confirming certain hypotheses. The significance of these findings can only be appreciated with a better, more

qualitative analysis of the lessons that takes into account the relationships between the codes, and a deep knowledge of the classroom dynamics in the countries being studied.

### *Dynamic Time Warping for Comparing TIMSS Lesson Sequences*

The dynamic time warping (DTW) distance between two sequences provides a fairly accurate measure of how dissimilar the sequences are. A distance of 0 implies that the sequences have an identical ordering of their constituent elements because the DTW algorithm accounts for code compressions and expansions.

The overall goal of the DTW distances was to find patterns in the lesson sequences by seeing how similar or dissimilar pairs of lesson sequences are, and to use that information to generate clusters of similar lessons. These clusters would indicate the different ways that a lesson is played out over time, and, if the clusters line up with groupings by country, we can then make a case for country specific sequences of classroom events. Unlike the motif extraction process, where the results were presented by country, this algorithm does sequence comparison and analysis from the ground up, without first assuming that different countries will look different.

I will first describe the overall process of applying the DTW algorithm to the TIMSS data, and then I will present the results of a few of the many analyses carried out, with the help of techniques such as multi-dimensional scaling (MDS) that can create a visual map of the distances between lessons. The value and validity of these results will be discussed in the last section of this paper.

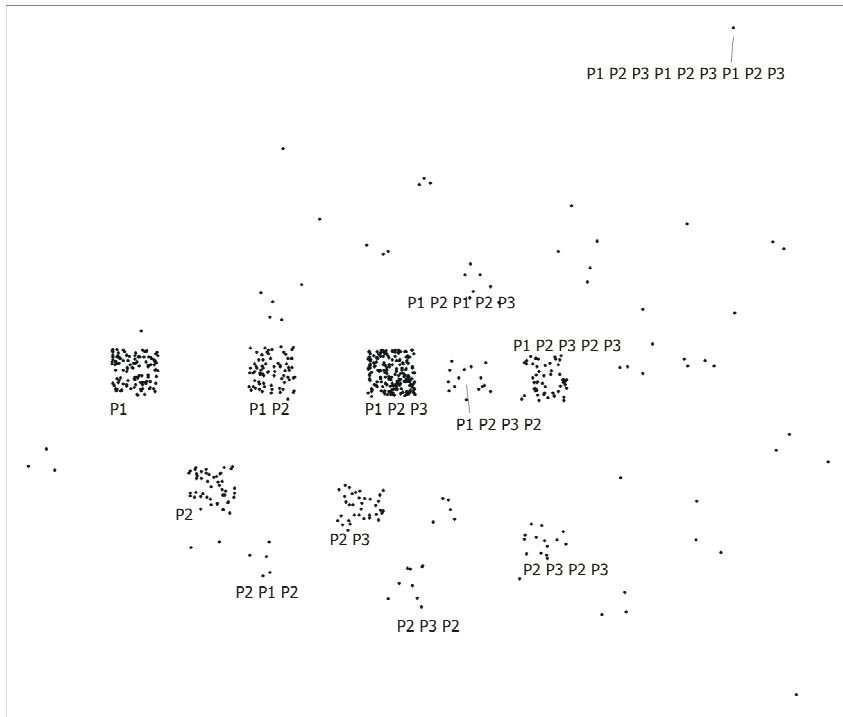
My first step was to choose the types of lesson sequences I was interested in. I ran the DTW algorithm for all three simple fundamental sequences (based on P, CI and Act codes respectively). Because the distance measure from the DTW algorithm is less for shorter sequences and greater for longer sequences, I also tried the algorithm with proportion-segmented

sequences (PSS) because PSSs are all the same length irrespective of the actual duration of the lessons. Each lesson was transformed into a 2% PSS (i.e., it was divided into 50 equal segments) based on P, CI and Act codes alone respectively. I also created a 2% PSS for P-CI pairs. I did not attempt the other pairs (P-Act, or CI-Act) because the individual sequence elements become too numerous to be easily interpretable. However, I did attempt to find DTW distances for the PSS based on code triples, just to see how different the results were from the other analyses.

For each set of lesson sequences chosen, the DTW algorithm was applied to every pair of sequences and a 638 x 638 dissimilarity matrix was created (there were 638 lessons in the dataset). Each dissimilarity matrix was converted into a two-dimensional visual map using MDS (Kruskal & Wish, 1978) with random starts to ensure an optimal solution. The PROXSCAL procedure in SPSS was used to do the MDS analysis. I created this map to help visualize the distances between sequences and to see if any obvious clusters occurred.

The following plots and observations are a demonstration of the kinds of results one can expect by visualizing DTW distances between sequences. The fundamental P sequences are presented and analyzed in more detail than the other results in this section because they are the shortest, and therefore easiest to interpret.

*MDS map of DTW distances between fundamental sequences based on P codes.* Figure 14 shows the two-dimensional space in which all 638 fundamental P sequences were placed as points. The original coordinates of the points frequently overlapped because they represented identical lesson sequences. To aid the visualization of the cluster of lessons that were identical, I jittered each point by a small random amount so that the points spread out but still visibly belonged together.

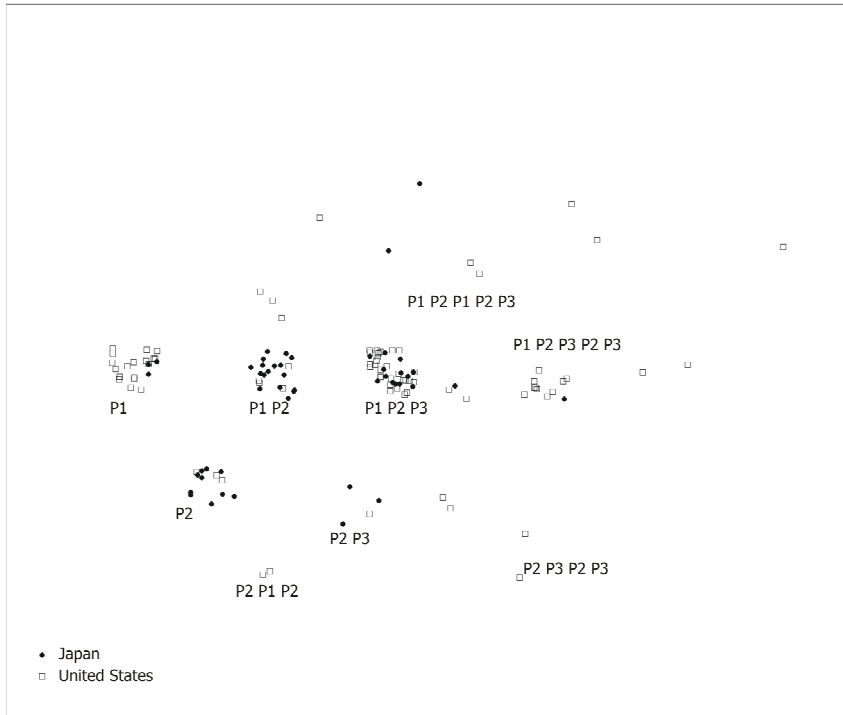


*Figure 14.* MDS map of DTW distances between fundamental P sequences.

The main clusters of lesson sequences are labeled and the “P1 P2 P3” sequence is the most prevalent—“a review, followed by an introduction of new material, and ending with some practice of the new material”. Lessons that are not within the main clusters have slight variations in sequences that were too numerous to individually label. The lesson at the top right of the figure was interesting because it repeated the basic “P1 P2 P3” pattern several times.

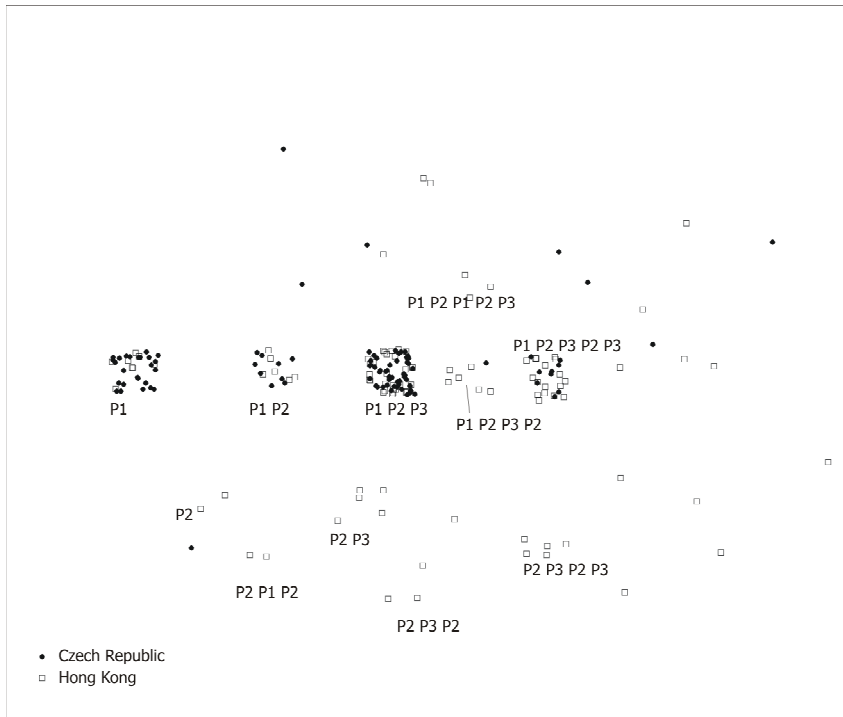
This plot is useful for understanding the different types of lesson sequences that occurred. However, displaying the country information for each lesson will help discover characteristic sequences for each country, similarities between countries and so on. Plotting each point with a different color or symbol makes the graphic difficult to interpret so I picked a few countries and will show them two at a time to facilitate comparisons. I have also made some interactive scaleable vector graphics (SVG) files available online at <http://netfiles.uiuc.edu/sujai/www/svg> where the full color MDS plots can be manipulated to show one or more countries at a time. In

addition, placing the cursor over any of the lessons will exhibit that lesson's sequence to one side of the screen.



*Figure 15.* MDS map of DTW distances between fundamental P sequences: Japanese and US lessons.

The Japanese and US lessons in Figure 15 share some common features. Many lessons from both countries exhibit the “P1 P2 P3” sequence. However, several US lessons do exhibit the “P1” and “P1 P2 P3 P2 P3” sequences whereas practically none of the Japanese lessons show them. In general, more US lessons switch more often between different classroom purposes, whereas the Japanese lessons do not switch as often on this code dimension.



*Figure 16.* MDS map of DTW distances between fundamental P sequences: Czech and Hong Kong lessons.

In Figure 16, similar numbers of Czech and Hong Kong (HK) lessons share sequences like “P1 P2” and “P1 P2 P3”. The difference in the variety of sequences followed is significant, with the Czech lessons basically following four types of sequences whereas the HK lessons are all over the map. Another interesting difference is that very few of the HK lessons and a large number of the Czech lessons stay engaged in Review (P1) throughout the lesson.

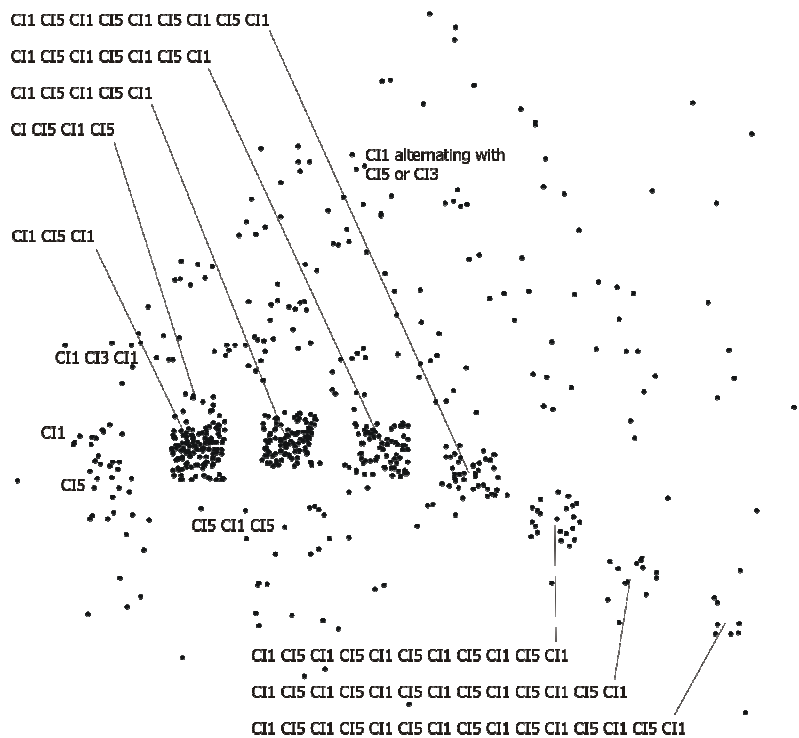


Figure 17. MDS map of DTW distances between fundamental CI sequences.

*MDS map of DTW distances between fundamental sequences based on CI codes only.*

The map of the distances between all lessons based on fundamental CI sequences (Figure 17) shows some interesting clusters. Although most lessons are arranged in clusters that represent alternations between CI1 (public interaction) and CI5 (private interaction) codes, there is a large and diffuse group of lessons that includes the CI3 code in between CI1 and CI5 repetitions. The clusters demonstrate greater frequencies of back and forth switches towards the right of the map.



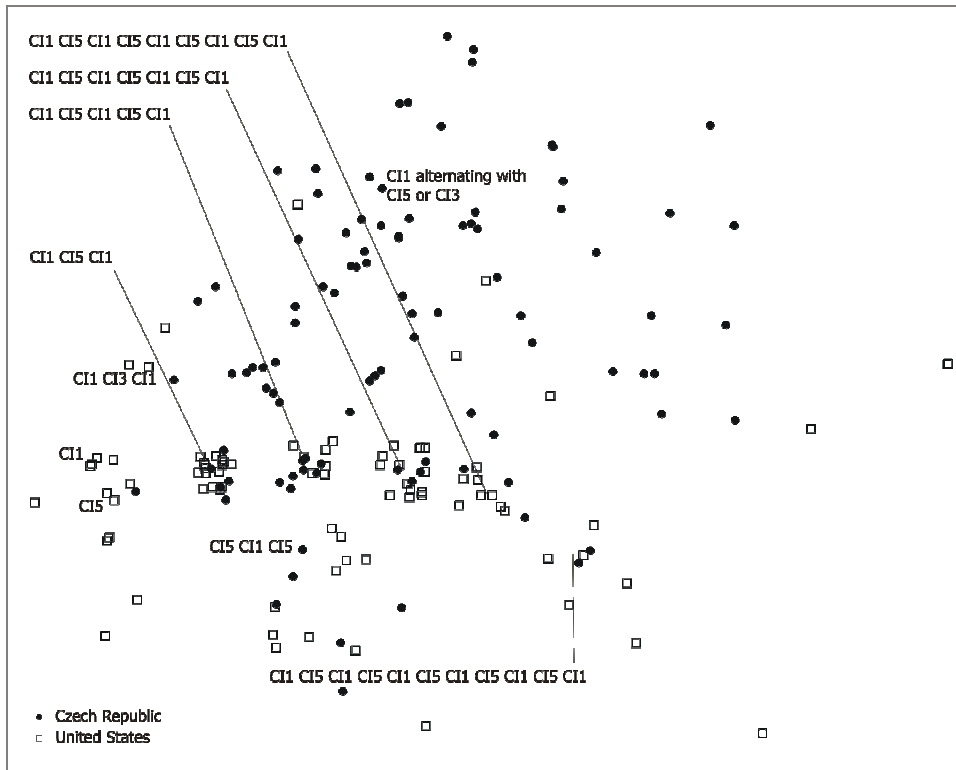


Figure 19. MDS map of DTW distances between fundamental CI sequences: Czech and US lessons.

In Figure 19, I compare US and Czech lessons in the MDS map made up of DTW distances between the fundamental CI sequences. Some Czech and some US lessons have alternating CI1 and CI5 sequences in common along the center of the MDS map. However, the Czech lessons frequently used the CI3 code which placed them in the top half of the map. The US lessons either began with a CI5 code, or they used other codes such as CI2 or CI4, and that put a number of them at the lower end of the map.

*MDS map of DTW distances between 2% proportion-segmented P sequences.* The first few examples of results from the DTW analysis were based on the fundamental P and CI sequences. The next example is an MDS map of the distances between sequences where each lesson was cut into 50 segments and the purpose code of each segment was recorded.

The difference between Figure 20 and the previous MDS representation of distances based on fundamental P sequences is that there are less clearly defined clusters of lessons. I have included this result as an example of how the proportion-segmented sequences capture duration information as well as order information. The elongated cluster in between the “P1” (review) and “P2” (introduction of new material) sequences clearly has different types of lessons at either end, those that spend a long time on review compared to those that spend a long time on introducing new material.



Figure 20. MDS map of DTW distances between 2% proportion-segmented P sequences.

*MDS map of DTW distances between fundamental sequences based on 2% proportion-segmented sequences made up of P-CI-Act code triples. As a final example of the ways in which*

DTW distances let us see patterns in the TIMSS lesson sequences, Figure 21 shows the distances between sequences made up of code triples. In the previous examples the sequence elements were single codes, and calculating the DTW distance between the sequences required that each individual code mismatch contributed a distance of one to the pair-wise distance. In this example, the distance function between sequence elements was modified to account for the fact that the elements were code triples such as “P1-CI5-AO” and not single codes such as “P1”. A difference in each dimension of codes was weighed differently with P (purpose) code differences weighed the most (as they changed the least often in the overall data set), and Act (activity) codes weighed the least (as they changed the most frequently in the overall data set).



*Figure 21.* MDS map of DTW distances between 2% proportion-segmented sequences based on P-CI-Act triples.

The resulting MDS map looks somewhat similar to the one in Figure 20 but the clusters have become even more diffuse due to the greater variety in the sequences representing the lessons.

*Summary.* MDS maps helped visualize the DTW distances calculated between pairs of sequences. Similar lesson event sequences were nearer each other in the MDS map and occurred in clusters, making it easier to visualize these similarities. More detailed descriptions of these clusters are also possible using size and membership analyses, but those were not the focus of this study.

Fundamental sequences based on P codes (purpose) and CI codes (classroom interaction) show up in clear clusters on an MDS map. These maps allow us to quickly identify the different types of sequences of events taking place and to see at a glance the number of lessons that exhibit a particular sequence.

Plotting the lessons by country also enables us to identify sequences that only occur in some countries and not others. This information ties in well with the motifs discovered in the previous section. For example, the motif “B P1” in Figure 11 shows that almost all the Czech begin with the P1. That fact is represented in even greater detail in Figure 16, where the Czech lessons clearly occur in only a few clusters, and all these clusters are for sequences that begin with P1. In addition, we can see which other countries have lessons with sequences similar to the Czech lessons. Similarly, many Hong Kong lessons reported the motif “P2 P3 P2” and they show up clearly in Figure 16 as parts of the clusters of lesson sequences “P1 P2 P3 P2”, “P1 P2 P3 P2 P3”, “P2 P3 P2”, and “P2 P3 P2 P3”. Thus, MDS maps of DTW distances have an advantage over the motifs in that they show every lesson in the dataset because they do not aggregate the information about individual lessons into a single data point.

The MDS maps also allowed me to identify characteristic sequences for some countries. For example, the Dutch lessons usually engage in very simple classroom interaction sequences such as “CI1 CI5 CI1”, rarely switching forms more than four times. However, no single cluster in any of the MDS maps was representative of all of the lessons in a country. Each country typically had lessons in several clusters (e.g., the US lessons in Figure 19), implying that there was no one national script of events that described all the lessons in a country.

#### *Hidden Markov Models for Determining Underlying Structure TIMSS Lesson Sequences*

As with the motif extraction and DTW methods, the first step in using HMMs to represent TIMSS lessons is to decide the kinds of sequences that we want to work with. I chose the fundamental sequences over the time-segmented and proportion-segmented sequences because they contained complete information of each event that took place, and because they did not repeat codes with longer durations. Repeated codes tend to bias the HMM to represent the runs in the sequence more than the transitions between codes.

The P and CI codes yielded interesting results with the motif extraction and DTW methods because it is easier to interpret patterns found in sequences made up of a few codes. HMMs, however, work well with larger numbers of different elements and my goal was to see if I could find simpler underlying structural similarities in complex sequences made up of many different elements. Therefore, I chose to work with the fundamental sequences made up of P-CI pairs. Each P-CI pair describes the purpose of the classroom and the type of classroom interaction occurring at a given point of time.

With three P codes (P1, P2, and P3), five CI codes (CI1, CI2, CI3, CI4, and CI5), two extra codes for encoding gaps, and some combinations that never occur, there are a total of 21 P-CI pairs. The HMM therefore has 21 finite events that can be observed. I attempted to train a

different model for each country using different numbers of hidden underlying states ranging from 4 to 9. Each country’s model was tested against the sequences that were used to train it and the highest likelihood values were obtained when the number of underlying states was seven.

I used the country HMMs to see if the models fit some lessons better than others. For example, if the lessons from other countries did not fit the model for the US as well as the US lessons did, then that would be a point in favor of the argument that each country has a distinctive sequence of underlying states.

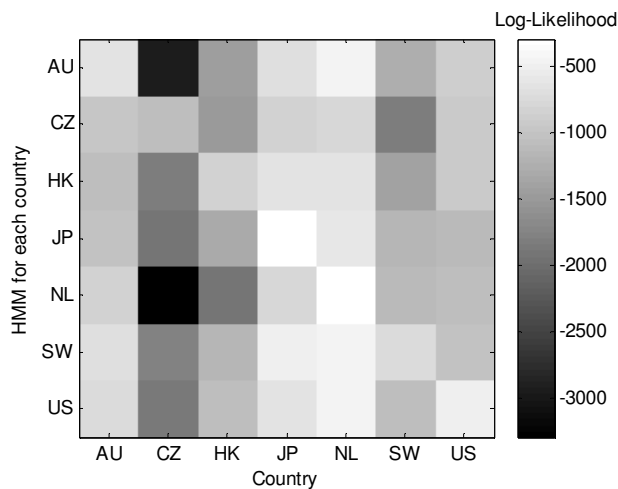


Figure 22. Likelihood of sequences within countries to fit the HMMs for all countries.

Each of the seven rows of the matrix in Figure 22 represents an HMM for a country. There are seven columns representing each country, and each cell indicates the likelihood that the lessons in that column’s country came from the HMM for that row. A lighter shade of gray indicates a higher likelihood and a darker shade indicates a lower likelihood. The bar on the right denotes the precise log-likelihood values associated with each color. Each column answers the question “Which HMM best fits the group of sequences associated with this column?” The matrix shows that the highest likelihood cell in each column is the one associated with that

columns country. For example, column two represents the Czech Republic, and the most likely HMM for this column is in the second row, which corresponds to the Czech HMM. However, reading across a row answers the question “How well do lessons from different countries fit the HMM that this row represents?” The answers to this question demonstrate the inability of the HMMs to correctly classify the lessons. Five of the HMMs—Australia, Czech Republic, Hong Kong, Switzerland, and the US—do not match the group of sequences that was most likely to be associated with that HMM. The overall conclusion I drew was that although it may be possible to train HMMs for each country’s sequences, these models cannot be used to classify new sequences, and thus there is no underlying pattern in the sequences that is unique for any country.

When I tried to classify each of the 638 lessons by testing them against the seven different country models, the results were not at all indicative of any sort of a country specific pattern.

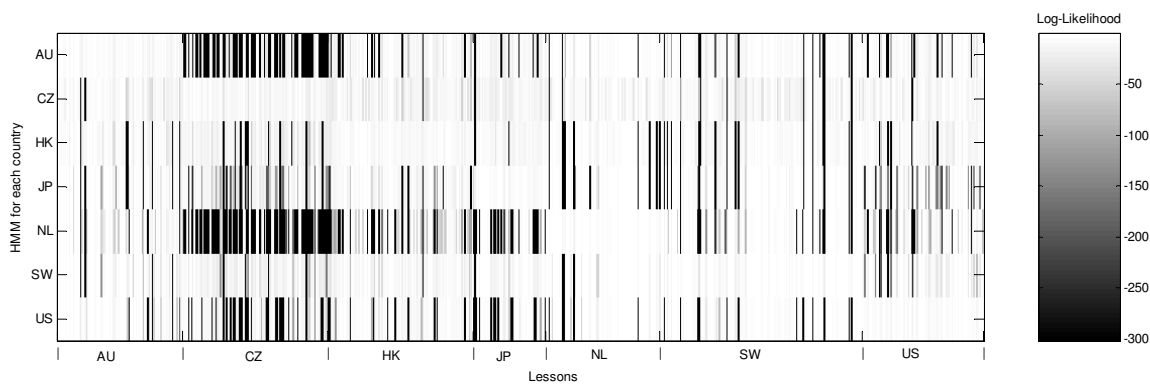


Figure 23. Likelihood of all lesson sequences to fit the HMMs for all countries.

Figure 23 is similar to the previous figure in that each row represents the HMM for a country. However, the columns now represent individual lessons, and the value of each cell in the matrix is an indicator of how likely it is that the lesson (column) came from a particular

HMM (row). The first 87 columns indicate Australian lessons, the next 100 indicate the Czech lessons, and so on. One would have expected to see lighter regions only when the lessons from a country were tested against that country's model, but in this figure, there are light regions in many places. For example, a group of Swiss lessons (right above the label "SW" on the x-axis) fit the models of every country quite well, as did some Dutch and some Australian lessons. This result is most probably a consequence of the length of these lessons, because shorter sequences generally fit any HMM better than a longer sequence.

*Summary.* The goal of the HMM analysis was to see if there were any underlying regularities in the sequences belonging to a country that were not intuitively obvious. An HMM for each country was created by training the model using all of that country's sequences. This HMM frequently matched lessons from another country better than the lessons from that country itself. Because the HMMs have very little ability to correctly classify lessons, it becomes possible to conclude that there are no country-specific sequence regularities. This finding is not surprising in the light of the results from the DTW distance section. The DTW analyses showed how all the lessons from one country did not exhibit one sequence. Instead, the lessons from a country were typically split into two or more clusters of characteristic patterns. Lessons from more than one country were also often found in the same cluster in the DTW results, which would account for the results from the HMM analyses.

## CHAPTER 4

### DISCUSSION

The methods for finding patterns in categorical data sequences presented in the first part of this paper demonstrated some of the wealth of information present in datasets of sequential episodes such as the TIMSS video study. Motif extraction, dynamic time warping and hidden Markov models all have certain characteristics that make them more or less applicable to different types of research questions. This discussion section will raise some specific issues about the methods I used to find patterns in the TIMSS data, as well as general issues about the validity of the results obtained.

#### Evaluating the Methods

The motifs found in the TIMSS data were simple sub-sequences that were often no more than three or four elements long. It is possible that some lessons exhibited more complex motifs that were spread out over the entire lesson and interrupted by other codes. Allowing more wildcards, and tweaking other Teiresias motif detection parameters could find other consistencies as well. The problem of finding too many patterns would still remain, however, and a good way of reporting the most significant patterns is still needed. The STEPDISC procedure in SAS does a reasonable job of short-listing a few lesson sequence motifs, but it might miss some obvious motifs in an effort to find the ones that discriminate best between countries. This selection procedure is also limited by the fact that it only works if you already have a grouping of sequences that you want to find characteristic patterns for.

Distance algorithms have the advantage that they do not rely on an a priori grouping of sequences for interpretation. They are thus ideal for bottom-up approaches to sequence

clustering. The clusters formed using the DTW distance method did not line up well with the country clusters. These results bring into question the claim (Hiebert et al., 2003; Stigler & Hiebert, 1999) that teachers in different countries follow characteristic cultural scripts while teaching a lesson. My results show that there is no one sequence of events characteristic of all the lessons in a country. However, there is evidence that each country can be divided into a few characteristic clusters of lesson sequences. For example, the Czech lessons only exhibit purpose code sequences “P1”, “P1 P2”, “P1 P2 P3”, and “P1 P2 P3 P2”, and no others.

My results do not preclude the possibility of national teaching scripts because it is possible that the purpose codes were too coarse, and the activity codes too fine to really capture the sequence of events in the classroom. Additionally, as the Learners’ Perspective Studies (Clarke & Mesiti, 2003; Shimizu, 2003) have pointed out, patterns of consistencies may occur at the level of a teaching unit and not at the level of a single lesson.

The DTW distance method was most useful for interpreting patterns in sequences that had few elements (such as the purpose and classroom interaction codes). This method is also useful for observing relationships between sequences because it represented each lesson in the entire dataset. As Tufte (1990) points out, aggregate displays of information run the risk of hiding vital information about individual data elements.

HMMs are useful tools for studying sequences provided we know something about the way the sequences are grouped together. In this study, I tried to train HMMs for each country but the models had no predictive power for classifying other lesson sequences. Again, the fact that no consistencies were found could mean either that no consistencies exist, or that the chosen coding system failed to capture them. As in the case of the DTW results, it is also possible that

the consistencies exist at a level of teaching organization higher than the level of an individual lesson.

### Differences Between Countries

Are there clear differences between countries based on the patterns found in lesson sequences? Based on my analyses, I do not think that there are any patterns in the sequences that belong exclusively to one country and not another. When I considered two countries at a time using the DTW method, some clusters of lesson sequences seemed to belong to one country more than the other (e.g., “P1” as a sequence showed up in several US lessons but in hardly any Japanese ones). In addition, although there was little evidence for a single national script, it is very likely that there are three or four national scripts, and the lessons within a country exhibit one of these scripts. I also found that lessons from more than one country had the same sequence of events.

The HMM method for classifying sequences failed to show consistent country differences. The motif extraction method was useful in identifying motifs, but the existence of specific motifs in a country cannot on its own be used as a claim for a national script. Anytime we *begin* with the assumption that the countries are different, we risk interpreting analyses in ways that only confirm our initial assumptions.

It may be true that there are country differences that occur at a level that is not captured by this set of codes or that occur on a different time frame (across the teaching unit and not the individual lesson).

In general, the validity of any of these results depends on the codes used to categorize the events in the classrooms. The codes make assumptions about what is important. For example, a

skillful teacher might use a complex narrative that incorporates review, new material, and practice in a way that is difficult to separate into different sections. Coding systems are also influenced by what seems tractable. For instance, discourse features are notoriously hard to categorize into codes – but might hold the key to the style of teaching and learning in a classroom. Similarly, what is described as a ‘Review’ in one country may look very different from a ‘Review’ in another country. For these reasons, methods like analyses of sequences should be used in conjunction with more qualitative studies in comparative educational research.

The overall goal was to demonstrate that we can find patterns in sequences of behavioral information such as classroom records. There is no Motif extraction and HMMs can both be used to classify groups of sequences and find the patterns or structures that characterize them, if such consistencies exist. Distance algorithms such as DTW, combined with visualization techniques such as MDS, are a better way of comparing individual sequences and seeing how they cluster together. The three methods outlined should be useful to any researcher interested in studying events that occur over time.

## REFERENCES

- Allison, P.D. (1984). *Event History Analysis: Regression for Longitudinal Event Data*. Beverley Hills: Sage.
- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17), 3389-3402.
- Bakeman, R., & Gottman, J. M. (1997). *Observing interaction: An introduction to sequential analysis*. Cambridge: Cambridge University Press.
- Cambridge University Engineering Department (2003). *Hidden Markov model toolkit*. Retrieved March 31, 2004 from <http://htk.eng.cam.ac.uk/>
- Clarke, D.J. & Mesiti, C. (2003). *Addressing the challenge of legitimate international comparisons: Lesson structure in Australia and the USA*. In L. Bragg, C. Campbell, G. Herbert, & J. Mousley (Eds.), *Mathematics education research: Innovation, networking, opportunity*, Proceedings of the 26th Annual Conference of the Mathematics Education Research Group of Australasia, Vol. 1, 230-237.
- Givvin, K. B., Jacobs, J. K., Hollingsworth, H. (2003, April). "Lesson signatures": A visual display of country differences in lesson structure. Paper presented at the annual meeting of the American Educational Research Association, Chicago.
- Griffin, W. A. (2003). Affect pattern recognition: Using discrete *hidden Markov models* to discriminate distressed from nondistressed couples. *Marriage & Family Review*, 34(1-2), 139-163.
- Gusfield, D. (1997). *Algorithms on strings, trees, and sequences: Computer science and computational biology*. Cambridge University Press.
- Hausler, S. (2004). *Hidden Markov models: A tutorial for the course Computational Intelligence*. Retrieved March 31, 2004 from <http://www.igi.tugraz.at/lehre/CI/tutorials/HMM/index.html>
- HCIL (2002). *Visual exploration of time-series data*. Retrieved March 31, 2004 from <http://www.cs.umd.edu/hcil/timesearcher/>
- Hiebert, J., Gallimore, R., Garnier, H., Givvin, K. B., Hollingsworth, H., Jacobs, J., Chui, A. M. Y., Wearne, D., Smith, M., Kersting, N., Manaster, A., Tseng, E., Etterbeek, W., Manaster, C., Gonzales, P., & Stigler, J. W. (2003). *Teaching mathematics in seven countries: Results from the TIMSS 1999 video study* (NCES 2003-013). Washington, DC: U.S. Department of Education.
- IBM Bioinformatics Group (2003). *Teiresias-based Text Pattern Discovery Using Individual Symbols*. Retrieved March 27, 2004 from <http://cbcsrv.watson.ibm.com/Ttspd.html>

- Kruskal, J. B., & Liberman, M. (1983). The symmetric time warping algorithm: From continuous to discrete. In J. B. Kruskal & D. Sankoff (Eds.), *Time Warps, String Edits and Macromolecules*. (pp. 125-162). Reading, MA: Addison-Wesley
- Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling*. Beverley Hills: Sage.
- LessonLab (2003). *TIMSS-R video math coding manual*. Retrieved February 19, 2004 from <http://www.lessonlab.com/timss1999/download/TIMSS%201999%20Video%20Coding%20Manual.pdf>
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8), 707-710.
- Magnusson, M. S. (2000). Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments & Computers*, 32, 93-110.
- Murphy, K. (2003). *Hidden Markov Model (HMM) toolbox for Matlab*. Retrieved March 31, 2004 from <http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html>
- Rigoutsos, I., Floratos, A., Parida, L., Gao, Y., & Platt, D. (2000). The emergence of pattern discovery techniques in computational biology. *Metabolic Engineering*, 2(3), 159-177.
- Schliep, A., Rungtarityotin, W., Georgi, B. (2003). *GHMM: A LGPL-licensed hidden Markov model library*. Retrieved March 31, 2004 from <http://ghmm.sourceforge.net/>
- Shimizu, Y. (2003). *Capturing the structure of Japanese mathematics lessons as embedded in the teaching unit*. Paper presented as part of the symposium "Mathematics Lessons in Germany, Japan, the USA and Australia: Structure in Diversity and Diversity in Structure" at the Annual Meeting of the American Educational Research Association, Chicago, April 21-25, 2003.
- Stigler, J.W., & Hiebert, J. (1999). *The teaching gap: Best ideas from the world's teachers for improving education in the classroom*. New York: Free Press.
- Tufte, E. R. (1990). *Envisioning information*. Cheshire, CT: Graphics Press.