Categorizing Event Sequences Using Regular Expressions

This is a reprint of a paper that first appeared in the IASSIST Quarterly Vol. 21:2 without tables. We apologize to the authors for this error of omission and reproduce the paper here in its entirety.

by Lisa Sanfilippo & John Van Voorhis[*]

Introduction

Researchers who work with large sequential datasets are often limited in the kinds of analytic strategies they can use because of the sheer size of the data. Automated techniques for analyzing sequences were developed in the 1960s by scientists studying DNA, RNA, and proteins. In a classic volume on sequence analysis, Sankoff and Kruskal (1983) demonstrated its potential application for subjects as diverse as bird songs and macromolecules. In other work, Andrew Abbott developed "Optimal Matching" for sequence analysis in the field of sociology.

In this paper, we describe a technique for analyzing sequences using Regular Expression Matching (REM). This technique allows researchers to examine patterns in longitudinal data by condensing sequences of events into smaller, more tractable units. We also briefly discuss the development of a database structure that facilitates this kind of analysis.

Although all sequence analyses compare linear arrangements of symbols, whether in human behavior or DNA, they differ in their assumptions about what makes two sequences similar or different. Sequences in their original form often contain too much detail for useful comparison, since the possible permutations of occurrences can be limitless. Therefore, in all cases researchers must create the rules that define sequence similarity for their analyses.

Methods for determining sequence similarity are often referred to as sequence-matching algorithms. These algorithms are mathematical, and compare sequences without reference to the semantic or theoretical structures that created them. When using such methods, researchers who wish to place their analyses in an appropriate context must carefully define what events represent the phenomena of interest.

The technique described in this paper was developed as an alternative to existing algorithms and allows researchers to identify sub-patterns of events within sequences at the start of their analysis, based on theoretical or practical considerations. Because this technique operates on a single sequence at a time, it is faster than processes that require comparing many sequences to one another.

The Project

To illustrate REM, we will describe how we used it to analyze the sequence of events that led to a child's placement into foster care in three states: Illinois, Michigan, and Missouri. We were looking for systematic demographic and geographic differences among children that correlated with the events they experienced in the child welfare system. REM was developed to describe and compare the pathways the children took through this system. The data were derived from the administrative data systems of the Illinois Department of Children and Family Services, the Michigan Family Independence Agency, and the Missouri Department of Social Services.

Preliminary Data Processing

We received two data extracts from each state: one covering investigations of child abuse and neglect in the Child Protection System (CPS), and the other, services such as foster care to children in the Child Welfare System (CWS).[1]

We began by creating a project database for each state with the same essential structure. Each state's database contained tables for CPS and CWS data and one table for demographic information on the children. Next, we created an event table that contained all of the administrative events for all of the children in each system. We then transformed each child's events into a sequence variable or "history." Finally, we used regular expression matching to formulate "careers" by reducing the history sequences. At each step in the process, we preserved enough information from the previous step to retain flexibility in the subsequent steps. As the categories became broader at each step, the comparability of the data across states increased.

Creating the Event Table
In this analysis we focused on four key administrative events:

(1) indicated investigation, an investigation in which credible evidence of abuse/neglect was found,

(2) unfounded investigation, an investigation in which no credible evidence of abuse/neglect was found,

(3) case opening, when a case was opened for child welfare services, and

(4) placement, when a child was placed in a foster home or institution.

Table 1. Example Event and History Codes

Event Code	History Code	Description
0	0	Birth
1	1	Indicated Investigation
3	3	Unfounded Investigation
9	9	Case Opening
15	B	Foster Home
17	C	Hospital Health Facility
23	C	Department of Mental Health Institution
6	6	Case Closing
18	D	Home of Parent (Ending Placement)

We created one record for every event a child experienced in either the CPS or the CWS. We then coded every record with a number denoting a particular event type (See "Event Codes" in Table 1). These records contained the child's ID, an event date, and an event type code (See Table 2).

Table 2. Example Event Data for One Child

ID	Event Code	Event Date
125	0	Feb 13, 1978
125	3	Dec 16, 1991
125	3	Dec 17, 1991
125	3	Jan 5, 1992
125	1	Feb 2, 1992
125	3	Apr 21, 1992
125	9	May 20, 1992
125	17	May 20, 1992
125	23	Jun 15, 1992
125	18	Apr 17, 1993
125	6	Jul 10, 1993

Creating the History Sequences
We transformed each child's event records into a single sequence of codes, since as separate records the table structure was not appropriate for sequence analysis. To make the programming and its interpretation easier, we used only single-character codes in the history sequence. Although each history code represented a single event, a given code value could represent more than one type of event (See History Codes" in Table 1).

We first reviewed a frequency distribution of the history sequences to identify the most common sequences and to see the repetition of patterns within and among sequences. This review also revealed data entry errors that we could correct or eliminate, such as children receiving services before their birth or children being born multiple times.

Although we had anticipated that the variation in the patterns between sequences would make them unsuitable for analyses in their present form, we had not foreseen the amount of variation in the length of the sequences. For example, examining the distribution of event sequences revealed that many children experienced only one event, while others experienced up to fifty. This wide variation in length made it difficult to make meaningful comparisons among cases and suggested that we needed a method that would not rely solely on whole-sequence comparison. Therefore, we focused our attention on identifying the sub-patterns which we had observed in the sequences.

Creating the Career Sequences
One goal of our research was to elucidate the connection between CPS investigations and a child's subsequent placement in foster care. We had three initial questions: (1) What sequences of investigations **never** resulted in a child welfare case opening and placement? (2) What

sequences of investigations resulted in the child's first placement? and (3) What sequences of events resulted in the child entering the system without an investigation?

Because of our extensive work with the Illinois data and our contact with all three states regarding current and past practices and policies, we had some knowledge of what the most common patterns of events might be.

The following examples illustrate how this prior knowledge provided us with clues about what patterns to focus our attention on:

- We understood that the number of investigations a child experienced was not a critical factor in the caseworker's decision to place the child in foster care. We knew that children with histories composed solely of unfounded investigations were almost never provided with services, despite repeated contact with the department. Therefore, we believed that the number of *indicated* investigations would predict placement better than the raw number of investigations.

- We knew that, in one state, caseworkers were reluctant to remove children from their homes after only one indicated investigation unless they were in imminent danger. Thus, we expected that a child with one indicated investigation would be less likely to be placed into foster care than a child who had two or more indicated investigations.

- In all three states, we knew it was possible for children to experience a case opening and placement without an investigation of abuse or neglect, but we had no information on the frequency of such occurrences.

- Our prior analyses of the foster care data indicated that once in foster care, a child could move between placements numerous times before being returned home. Although the placements could be of different types, the child was still living away from his or her parents. As a result, we chose to treat a series of placements without a return home as one career event.

Regular Expression Matching

It became apparent in looking at the sub-patterns that they could be represented by regular expressions, a notation used widely in the computer science field for specifying and matching sequences.[2] (See Appendix.)

We created a file listing the regular expression patterns we had decided to analyze along with a "career" code for each pattern which is shown in Table 4. We grouped the patterns in passes because we knew that certain patterns occurred only at the very beginning of the history and we needed to control the generation of the matching program. The first pass was used to remove any events that occurred before a child was born. Since we were especially interested in the first series of investigations, we created a pass that only matched to initial investigation sub-sequences. The last pass, which was applied repeatedly until the history was exhausted, contained all of the sub-patterns we were investigating. From this pattern file we generated a series of programs to transform the data.

Table 3. Example History Record

ID	History Sequence
125	0333139CCD6

We used the AWK programming language for both our program generator and the matching programs themselves. An AWK program is composed of a series of pattern and action pairs. It automatically reads through data files one line at a time, and each line is matched against the patterns in the order they are listed in the program. When a line contains data that matches one of the patterns, the action associated with that pattern is executed. The patterns may contain regular expressions, while the actions are written in a language similar to the C programming language.

In our project, the program generator read the pattern file containing the sub-patterns of interest to us and generated a series of programs that used those regular expression patterns to process the history data. Each program in the series corresponded to a particular pass in the pattern file. If a pattern matched to the beginning of a history sequence, the matching characters were removed and the career code for that pattern was appended to the career sequence. The child's id, history, and career were then passed to the next program for the next pass. The final program passed the data back to itself until the history sequence was empty or until a fixed number of passes had been run. If the history sequence was completely matched, a lower case 'x' was appended to the career to indicate completion. An upper case 'X' was appended if more history remained after the maximum pass limit had been reached.

Analyzing the Career Sequences

Since our analysis was limited to examining the sub-patterns that led to a child's first placement, we did not analyze children's entire careers. Instead, we only analyzed the first four career events after a child's birth.

Because the REM approach simply recoded the original history sequences, it preserved the unit of analysis, thus allowing us to attach explanatory variables such as year of first entry into the system, sex, race, and region[3]. Once this information was stored in one file, we aggregated the data by creating a crosstabulation which contained frequencies for every combination of the career sequences and the explanatory variables. These files were relatively small

Table 4 History Sequence Sub-patterns Written in Regular Expressions

Pass	Regular Expression	Career Code	Event Sequence Description
1	.+0	a	A series of events before the birthdate
	0	b	Birthdate
2	3*13*11*	A	One or more indicated investigations in a series
	3*13*	B	One indicated investigation in a series
	3+	C	A series of unfounded investigations
3, 4, 5, 6 etc.	9	D	Case opening
	3*1[31]*	E	At least one indicated investigation in a series
	3+	C	A series of unfounded investigations
	6	F	Case closing
	[A-Z][A-Z]+	G	A series of placements
	[A-Z]	H	One placement

(fewer than 1,000 records) allowing us to import them into a spreadsheet program for final analysis and presentation.

Conclusion

The REM technique described in this paper departs from more common pattern matching methods in that it incorporates theory and practice into the actual matching process. Using this technique, researchers can test their assumptions about the structure of a sequence. It is an iterative technique that allows the analyst to explore patterns in the data and to compare them across populations simply and quickly. Because the process of developing the career file is split into several steps (i.e., creating the event table, creating the history sequences, and pattern matching), it provides many opportunities to check the data and to ensure that the processes are transforming the data correctly.

REM allows the researcher to take a very large dataset and to represent it in a much smaller form, while maintaining the critical details of event order and sequence. For example, in our Illinois database we began with an event file of over 5 million records. Transforming this file into history sequences, career sequences, and finally into a crosstabulation, decreased the size of the file by a factor of 5,000, making it significantly easier to work with.

The REM technique, as written in AWK, can save the researcher hours of processing time, in large part due to: 1) the way AWK reads data files (i.e., it automatically reads a file one record at a time) and 2) the minimal programming it requires. Performing the same analyses using a statistical software package would have required much more extensive programming and perhaps more important, would have restricted the kinds of questions we could have asked in exploring the original data.

Table 5. Example Career Sequence Output

	Output Record		
	ID	**History Sequence**	**Career**
Before Processing	125	0333139CCD6	
After Pass 1	125	333139CCD6	b
After Pass 2	125	9CCD6	bB
After Pass 3	125	CCD6	bBD
After Pass 4	125	6	bBDG
After Pass 5	125		bBDGF
After Pass 6	125		bBDGFx

Special Characters Used in Regular Expressions:

.	Any single character
[ ]	Any single character listed within the brackets
+	Any sequence of one or more of the preceding symbol
*	Any sequence of zero or more of the preceding symbol
|	OR - match one of the expressions the bar separates
( )	Treat the characters in the parentheses as a single symbol

Future Directions

Clearly, REM has a much wider application than what we have illustrated with our project. Our analysis did not utilize REM to its fullest potential. For example, instead of analyzing just the initial sequence of sub-patterns, REM could be used to analyze full careers. We could run a similar process against the career sequences to further shrink the number of categories.

Finally, we did not explore the sub-patterns in as much detail as we could have. For example, we included specific placement event types in our event table and history sequences but did not treat them as separate types. In the future, we can easily compare differences in children's histories following specific types of substitute care placements (e.g., home of a relative, private foster home, group home, etc.) based on this project's current database.

APPENDIX

Regular Expressions

In general, a character in an AWK regular expression matches itself. Some characters with special meanings in our pattern file are listed below along with some examples of their use. See the references for more details.

REFERENCES

Abbott, Andrew. 1995. Sequence Analysis. *Annual Review of Sociology,* 21:93-113.

Abbott, Andrew & Alexandra Hrycak. 1990. "Measuring Resemblance in Sequence Data: An Optimal Matching Analysis of Musicians' Careers." *American Journal of Sociology,* 96(1): 144-185.

Abbott, Andrew & John Forrest. 1986. "Optimal Matching Methods for Historical Sequences." *Journal of Interdisciplinary History,* 16(3): 471-494.

Aho, Alfred V., Brian W. Kernighan, & Peter J. Weinberger. 1988. *The AWK Programming Language*. Reading: Addison-Wesley.

Aho, Alfred V., Jeffrey D. Ullman. 1979. *Principles of Compiler Design.* Reading: Addison-Wesley.

Forrest, John & Andrew Abbott. 1990. "The optimal Matching Method for Anthropological Data: An Introduction and Reliability Analysis." *Journal of Quantitative Anthropology* 2:151-170.

Friedl, Jeffrey E. F. 1997. *Mastering Regular Expressions.* Sebastopol: O'Reilly & Associates, Inc.

Sankoff, David & Joseph B. Kruskal eds. 1983. *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison.* Reading, MA: Addison-Wesley.

NOTES

[1]. Child Protection Systems: in Illinois, the Child Abuse and Neglect Tracking System; in Michigan, the Protective Services Management Information System; and in Missouri, the Child Abuse and Neglect Data System.

Child Welfare Services Systems: in Illinois, the Child and Youth Centered Information System; in Michigan, the Children's Services Management Information System; and in Missouri, the Alternative Care Tracking System.

[2]. Regular expressions (REs) can recognize patterns which are left linear. Patterns, such as a balanced sequence of parentheses, cannot be recognized by REs

Regular Expression Examples:

.	Matches 3, 6, 9, C, D, 6
[31]	Matches 3 or 1
3+	Matches 3, 33, 333, 3333, ...
3*1	Matches 1, 31, 331, ...
(31)+	Matches 31, 3131, 313131, ...
[31]+	Matches 3, 1, 331, 131, 333, 111, ...
(3|1)+	Matches 3, , 33, 333, ..., and, 1, 11, 111, ...

because such patterns require “going-backwards” or maintaining information outside of the RE. For further information see Aho, Kernighan, and Weinberger 1988 in the references.

[3]. In all three states we differentiated the major urban area from the balance of the state.

[*] Paper preseneted at the IASSIST/IFDO 1997 Annual Conference Odense, Denmark May 7, 1997