Note

Elzinga, Cees H. 2010. ‘‘Complexity of Categorical Time Series.’’ *Sociological Methods & Research* 38:463-481. (DOI: 10.1177/0049124109357535)

This version is the Corrected Version of Record. It has been corrected to include Figure 3 and Table 5, which were inadvertently omitted from the article in the original print version.

Sociological Methods & Research
38(3) 463–481

DOI: 10.1177/0049124109357535
http://smr.sagepub.com

Complexity of Categorical Time Series

Cees H. Elzinga[1]

Abstract

Categorical time series, covering comparable time spans, are often quite different in a number of aspects: the number of distinct states, the number of transitions, and the distribution of durations over states. Each of these aspects contributes to an aggregate property of such series that is called *complexity*. Among sociologists and demographers, complexity is believed to systematically differ between groups as a result of social structure or social change. Such groups differ in, for example, age, gender, or status. The author proposes quantifications of complexity, based upon the number of distinct subsequences in combination with, in case of associated durations, the variance of these durations. A simple algorithm to compute these coefficients is provided and some of the statistical properties of the coefficients are investigated in an application to family formation histories of young American females.

Keywords

categorical time series, complexity, sequence comparison, sequence analysis

Introduction

Many demographers and sociologists study categorical time series that consist of sequences of encoded events, the events themselves belonging to one

[1]Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

Corresponding Author:
Cees H. Elzinga, Faculty of Social Sciences, VU University, Metropolitan Building Room N516, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
Email: ch.elzinga@fsw.vu.nl
*The online appendices are available at http://smr.sagepub.com/supplemental.

or more different life domains like work, education, or family formation. Often, such categorical time series come with associated vectors of durations, namely, the times spent in the states.

Some life courses are quite varied or complex: The individuals experience many transitions and many distinct states. Others, within the same time span, stay in one state only for the full observation time, and therefore, such life histories are said to be ''simple'' or ''stable.'' Apparently, life courses or, more general, categorical time series, seem to possess a property that can be used to order them in a meaningful way: as more or less varied, complex, or turbulent. However, the name *turbulence* may not be so well chosen because it also refers to irregularity in the flow of gases and liquids. This irregularity is unbounded and reflects randomness whereas the property that we intend to operationalize does not refer to randomness. Similarly, *variance* and *variation* are less suitable because they are strongly associated to numerical variables. We chose the word *complexity* since, as will appear, our quantification is a special case of what is called *d -complexity* in the combinatorics of strings (e.g., Iványi 1987; Kása 1998). For the moment it suffices to say that in the present context, complexity increases with the number of distinct states and with the number of distinct orderings of states. So, if we encode living single as *S*, living married as *M*, and living married with children as *MC*, the three family formation histories *S M*, *S M MC*, and *S M S M MC* increase in complexity while they might still cover the same time span.

To social scientists, complexity is an interesting property since demographical, sociological, and psychological theories (e.g., Arnett 2000; Buchmann 1989; Lesthaeghe 1995; Shanahan 2000) predict complexity of modern life courses to increase over time; namely, on the average, life courses of younger cohorts should exhibit more complexity than life courses of older cohorts. Similarly, sociological theory predicts that the average complexity of partnership histories should vary cross-nationally, namely, average complexity depends on welfare and legislative systems of the countries involved (Mills 2004). On the other hand, life courses of much older cohorts seem to demonstrate a tendency toward less complexity and more standardization (e.g., Bras, Liefbroer, and Elzinga forthcoming; Uhlenberg 1969, 1974).

The purpose of this article is to define and quantify this property of complexity in a useful way. In particular, we will propose to use a measure already used by Elzinga and Liefbroer (2007) to test hypothesis on de-standardization of family life trajectories. We will present an efficient algorithm for its calculation and discuss some of its properties.

Complexity as we use it here does not necessarily coincide with the complexity as perceived by the one who generated the series of events, nor is it

intended to reflect the complexity as it might be perceived by members of any social class or group. Complexity as it is used here refers to a concept that is comparable to the concept of the variance of a set of numerical observations. So, the advocated index should be sensibly applicable to categorical time time series from any substantive field.

In order to show how the proposed index fits into a more general context, we use the next section to concisely discuss two quite different notions of complexity of strings. In the third section, we discuss the index chosen and an efficient algorithm for its calculation. In the fourth section, we extend the measure to handle durations, and in the fifth section, we demonstrate some of its properties through an application to family life trajectories of three cohorts of U.S. females.

Notions of Complexity

Categorical time series are, ignoring durations, just strings of characters from some alphabet $\mathcal{A}$ that is determined by the application: For example, in case of labor market careers, the characters are symbols or acronyms for the relevant states like employment, unemployment, and so on. Therefore, we first discuss two quite different notions of complexity that directly apply to strings.

The first of these notions derives from what is called algorithmic information theory (e.g., Cover and Thomas 1991), a branch of computer science that deals with structure and randomness in the strings that constitute the input, the programs, and the output of computing. The second approach is combinatorial and originates from problems of string comparison. String comparison is important in microbiology where very long molecules like DNA are compared but string comparison is also relevant in physics, coding and encryption, machine learning, and the social sciences where career-like data such as life courses or mobility patterns are analyzed through ''sequence analysis'' (e.g., Abbott 1995; Elzinga 2005).

We start our discussion by considering two strings on a small alphabet $\mathcal{A} = \{a, b\}$:

$$x = abababababababababab \text{ and } y = abbabaaabbabbbbababaaba.$$

Obviously, the string x is very easy to describe: It consists of 11 concatenations of the substring ab. On the other hand, there seems to be no such simple rule with which y could be described. We conclude that describing x is very easy but that the most efficient description of y seems to be (a copy of) y itself. Observations like these inspired the famous Russian mathematician

Andrej Kolmogorov (1965) to define complexity of an object in terms of the length (i.e., the number of characters) of computer programs that describe (i.e., print) the object. The descriptive complexity $K_C(x)$ of a string x with respect to computer C equals the length $L(p)$ of a shortest program p that, when running on C prints x:

$$K_C(x) = min_{p:C(p)=x}\{L(p)\}.$$

Although Kolmogorov complexity is fundamental to many branches of mathematics, (e.g., Li and Vitányi 2008), it is not a computable function since we do not know how to decide whether or not a program is or is not a shortest program. However, Kolmogorov complexity is bounded from above by the Shannon (1948) entropy $H(x)$ of the sequence. To define this entropy, we need some notation: We write $x = x_1 \ldots x_k$ for a sequence that consists of k characters. Furthermore, we write $\mathcal{A} = \{a_1, \ldots, a_s\}$ for the alphabet of size s. For example, to describe labor market careers, we might use the alphabet $\mathcal{A} = \{e, u, v, n\}$ to denote the states ''employed,'' ''unemployed,'' ''vocational training,'' and ''not available.'' This alphabet has size 4 and all possible careers are described by concatenation of characters from $\mathcal{A}$. So, $x = ueveuen$ refers to a career. Let $p_x(a_i)$ denote the probability that when ''reading'' x, we encounter the character a_i, namely, $p_x(a_i) = Prob(x_j = a_i)$. For the example $x = ueveuen$, we thus have $p_x(e) = 3/7$ and $p_x(n) = 1/7$. Now we are in a position to define the entropy $H(x)$ as

$$H(x) = -\sum_{a_i \in \mathcal{A}} p_x(a_i) \log_2 p_x(a_i).$$

For our example career, we now compute $H(x) = -(\frac{3}{7}\log_2\frac{3}{7} + \ldots + \frac{1}{7}\log_2\frac{1}{7}) = 1.84$. Suppose that we have a career that consists of just one state, say $y = e$. Then the entropy of this sequence equals zero since now $p_y(e) = 1$ and all other probabilities equal zero: $H(y) = -(1\log_2 1 + 0\log_2 0 + \ldots) = 0$ (indeed, we need $0 \log 0 = 0$). On the other hand, for the career $z = uven$ we calculate $H(z) = 2$. So, if there is no uncertainty about the state, like in $y = e$, entropy equals 0 and when uncertainty is maximal because all states are equally likely, we find that entropy is maximal. Apparently, H quantifies a kind of uncertainty or variation in the sequences and is maximal when the probability distribution over the alphabet is flat; this maximum equals $\log_2 s$ for an alphabet of size s.

Now consider the careers $x = ven$ and $y = venvenven$. Clearly, $H(x) = H(y)$ since we have $p_x(a_i) = \frac{1}{3} = p_y(a_i)$. This demonstrates that Shannon (1948) entropy only quantifies the degree of randomness (lack of

structure) and is fully insensitive to transition frequency or the order of the events. This is not what we want: In y, much more is going on than in x.

There is an abundance of measures of complexity, variation, or diversity in categorical data, some well known like the indices of Gini (1921) and Theil (1972), others not so well known (e.g., Gibbs and Poston 1975; Wilcox 1967). All these indices have in common that they resemble Shannon entropy and that only the prevalence of states is considered. An elegant and easy to interpret example of these alternatives is Simpson's (1949) index of diversity:

$$D(x) = \sum_{i=1}^{d} p_x^2(a_i)$$

or, in the form that Blau (1977) used to introduce it into sociology,

$$D'(x) = 1 - D(x).$$

D' equals the probability that when two characters of x are picked, the characters will be different. For the example sequences x and y we obtain $D'(x) = D'(y)$ again. So, we conclude that ''entropies'' do not adequately quantify our intuitions about complexity of sequences.

A quite different approach to the issue of complexity derives from string comparison. The central problem in string comparison is to establish the similarity between strings: Strings are more or less similar when they have more or less parts in common. So, the more parts and the bigger the parts in either string, the more effort it will take to establish similarity. In this context it is only natural to consider strings as more complex if they have more distinct parts. A quantification of complexity then only requires a clear definition of what are distinct parts and a feasible algorithm to count these parts. However, if the definition of the parts would not be relevant to social scientists, the quantification itself would not be relevant. Therefore, we next discuss a few definitions of parts of sequences and their possible relevance to social scientists. To do so, we first need to introduce some concepts.

Previously, we already encountered the concept of ''substring.'' We say that any contiguous part of a string is a substring, special cases being the empty substring λ and the original string itself. With $abacbc = x$, *aba*, *acb*, *bc*, and *a* are substrings of x and they are not the only ones; on the other hand, *abc* is not a substring of x since its characters are not contiguous in x.

We could say that the all the distinct substrings constitute the parts of a string and we could count the number of distinct substrings to quantify the complexity of a string. The reader easily verifies that $x = ababab$ has

12 distinct substrings and that $y = abcdef$ has 22 distinct substrings. So, y would be more complex than x. In the context of computer science, this ''substring-complexity'' has been amply studied by, for example, de Luca (1999), Ferenczi and Kása, Levé and Séébold (2001) and Janson, Lonardi, and Szpankowski (2004).

However, in the social sciences, we are often interested in precedences of events without the restriction that they are contiguous in time. For example, it is relevant to know whether finishing education preceded entering parenthood or the other way around but it is mostly not so relevant whether or not these events were contiguous in time or not. Similarly, it is relevant to see that employment followed vocational training, irrespective of what was in between. So, in defining complexity of strings, we should not confine ourselves to parts of strings that are contiguous in the original string.

This relaxation of contiguity is provided for by a concept introduced by Iványi (1987): the d-substring (see also Kása 1998). A d-substring consists of characters that are separated by at most d positions in the original string and $d > 0$. For $d = 1$, we are back at the ordinary substring but with $d > 1$, some d-substrings are different from the ordinary substrings. For example, the 2-substrings of $x = abc$ are $\{\lambda, a, b, c, ab, ac, bc, abc\}$ and ac is not an ordinary substring of x while a and c are not adjacent. However, it is hard to understand why we should impose any constraint on the size of the separation d, since this separation also depends on the size of the alphabet. If the alphabet consists of very many distinct states or characters, namely, when our observations are very detailed, it is quite likely that any two particular events will be separated by other events.

If d is not limited at all, we allow for parts of strings in which the characters are separated by an arbitrary number of positions in the original string: The only confinement is that the characters, the events, have the same (temporal) order as in the original string. Such parts of strings are called *subsequences*, again with λ and the string itself as special cases. Substrings and d-substrings are special cases of this more general class of subsequences. For example, with $x = euenue$, un and uue are subsequences of x. Easily but tediously, the reader could verify that x has only 18 distinct substrings whereas it contains as many as 48 distinct subsequences.

Generally, u is a subsequence of x, precisely when all the characters of u also appear in x and in the same order and we denote this fact by writing $u \preceq x$. Let us denote the number of distinct subsequences of a string x as $\phi(x)$. The reader notes that $\phi(x) \geq 1$ since we we always have that $\lambda \preceq x$.

We compare $H(x)$ and $\phi(x)$ by looking at some extreme strings. First, we look at the string $x = aaaa$. Since $p_x(a) = 1$, we have that $H(x) = 0$, namely,

the minimum value of $H(\cdot)$. It is not difficult to establish that $\phi(x) = 5$ and one realizes that a string of length 4 could not have less than 5 subsequences (including the empty λ). So for this very ''uneventful'' string, we have that both $H(x)$ and $\phi(x)$ yield minimal values. On the other hand, consider $y = abcd$. We calculate that $H(y) = 2$ and we find that $\phi(x) = 16$. Again, we realize that neither H nor ϕ could have been bigger since the probability distribution over the alphabet $\mathcal{A} = \{a, b, c, d\}$ is flat and since reducing the number of distinct characters in x would certainly reduce $\phi(x)$. So it seems that for extreme strings, H and ϕ produce extreme values. Next we compare $x = abac$ and $y = abca$ to find out that

$$0 < H(x) = H(y) = 1.5 < 2,$$

$$5 < \phi(x) = 14 < \phi(y) = 15 < 2^4.$$

So, interestingly, the combinatorial subsequence-complexity function ϕ discriminates x and y whereas entropy H does not and indeed, in most substantive areas, one would consider x as being less varied or complex than the string y.

Finally, let us concisely consider strings with lengths that exceed the size of the alphabet and therefore have repeated characters. The reader notes that repeating events is a quite common phenomenon: unemployment and reemployment, entering or breaking up partnership, picking up education again after a spell of working, or moving to a new residence are reoccurring events in many lives. We know (e.g., Chase 1976; Flaxman, Harrow, and Sorkin 2004) that given an alphabet $\mathcal{A} = \{a_1, \ldots, a_s\}$, the n -long string w that maximizes ϕ has the cyclic structure

$$w = a_1 \cdots a_s a_1 \cdots a_s \cdots\cdots a_1 \cdots a_{\mathrm{mod}(n,s)},$$

namely, a repeated catenation of the same, fixed permutation of the alphabet. Surprisingly, and despite the description of this string being very short and simple, its entropy is maximal too since the probability distribution over the alphabet is (almost) flat. So again, in these extreme cases, H and ϕ yield comparable results although their rationale is quite different: ''predictability'' versus ''length of parts-list.''

We conclude that both entropy-based and subsequence-based complexity yield comparable results on extreme strings. However, for nonextreme strings, subsequence-based complexity yields quantifications that respect order-differences and are therefore more suitable to characterize social science categorical time series. Hence, it is interesting to describe an efficient

algorithm that counts distinct subsequences and that is fundamental to the method already proposed by Elzinga and Liefbroer (2007).

Complexity by Counting Distinct Subsequences

The discussion in this section is quite informal; in Appendix A, we take a less intuitive approach. However, for a rigorous proof of the correctness of the algorithm, the interested reader is referred to Elzinga, Rahmann, and Wang (2008).

First let us wonder about the boundaries of $\phi(x)$ and try to express these in terms of the length of the string x. To discuss these boundaries, we need the concept of a prefix and some notation. Let x be an n-long string; then the i^{th} prefix x^i of x consists of the first i contiguous characters of x. So, with $x = x_1 \ldots x_n$, $x^i = x_1 \ldots x_i$. For example, with $x = abac$, $x^3 = aba$, and $x^4 = x$. In particular, we always have $x^0 = \lambda$ and $x^n = x$.

Now we can discuss the number of distinct subsequences of an n-long string x that has exactly n distinct characters, namely, no character in x repeats. Thereto, we consider some prefix x^i with its number of distinct subsequences $\phi(x^i)$. We elongate x^i to $x^i x_{i+1} = x^{i+1}$ and wonder about $\phi(x^{i+1})$. Of course, we retain all the $\phi(x^i)$ subsequences of x^i. Let u be one of these, namely, $u \preceq x^i$. Then the elongation $x^i x_{i+1}$ generates $ux_{i+1} \preceq x^{i+1}$ and, because x has no repeating characters, ux_{i+1} must be new. This reasoning pertains to all the $u \preceq x^i$ so we must have that $\phi(x^{i+1}) = 2\phi(x^i)$ and therefore, using $\phi(x^0) = \phi(\lambda) = 1$, $\phi(x^n) = 2^n$, provided x has no repeating characters. For example, $\phi(x^2 = ab) = 4 = |\{\lambda, a, b, ab\}|$ and the new sequences that arise by elongating with $c \neq ab$ are $\{\lambda c = c, ac, bc, abc\}$ so $\phi(x^3 = abc) = 2\phi(x^2) = 8$.

On the other hand, consider an n-long string that consists of n repetitions of one and the same character. Then its distinct subsequences can only be discriminated by their lengths that vary between 0 and n. So, we must have that $\phi(x) = n + 1$ and hence we suspect that

$$n + 1 \leq \phi(x^n) \leq 2^n$$

for all nonnegative n.

In practice, social science categorical time series consist of one to several tens of events, so $\phi(x)$ itself is not a very practical scale because of the very large numbers it would often imply. Therefore, we propose to use the complexity measure $C(x)$ defined as

$$0 \leq C(x^n) = \log_2 \phi(x^n) \leq n$$

to the effect that $C(x) = C(y) + 1$ if $\phi(x) = 2\phi(y)$ and $C(\lambda) = 0$. In a geometrical sense, $C(x)$ is the $\log_2$ of the squared length of a binary-valued vector representing x in a high-dimensional Euclidean space. This directly connects the so defined measure of complexity to the data-analytical tradition of sequence comparison (e.g., Abbott and Tsay 2000; Elzinga 2005; Sankoff and Kruskal 1983; Wang and Lin 2007). In Appendix A, we will make some further comments on this.

Now the only remaining problems are, first, to numerically evaluate $C(x)$ and, second, to demonstrate its usefulness. We will deal with the first problem here and postpone the latter problem to the fifth section.

The key to a feasible algorithm to compute $\phi(x)$ is in the reasoning that leads to finding $\phi(x)$ for a string without repeats: We found that if a prefix x^{n-1} is elongated with a character that is new, namely, does not already appear in x^{n-1}, then that number of distinct subsequences will double as a result of the elongation:

$$\phi(x^n) = 2\phi(x^{n-1}) \text{ if } x_n \not\preceq x^{n-1}.$$

So, we should find a rule that tells us what to do if x_n is *not* new, namely, when $x_n \preceq x^{n-1}$. This must be different from doubling for if we double when x_n is not new, we count too much since some of the subsequences that end on the character x_n already exist. As an example, we discuss calculating $\phi(x = abcbc)$. Clearly, $\phi(x^3) = 2^3$ since in x^3, no characters repeat. But when we elongate x^3 to $x^3x_4 = x^4$, we do not obtain eight new subsequences: The subsequences of x^3 are $\{\lambda, a, b, ab, c, ac, bc, abc\}$ and doubling would mean that we pretend that $\lambda b = b$ and ab are new, which is obviously false. So if we double, we should immediately subtract the number of subsequences that were already found just before introducing $b = x_2$. So, we calculate that $\phi(x^4) = 2\phi(x^3) - \phi(x^1) = 16 - 2 = 14$. Next we elongate x^4 to $x^4x_5 = x^5$. Again, we have the problem of counting too much when doubling $\phi(x^4)$ so we compensate by subtracting the new subsequences that arose when we encountered c before. Hence, our calculation amounts to $\phi(x^5) = 2\phi(x^4) - \phi(x^2) = 28 - 4 = 24$.

Now we see that in general, when x_n is not new to x^{n-1}, we double and compensate by subtracting the new subsequences that arose from elongating with the same character the last time that we encountered it. Let $\ell = \ell(x^{n-1}, x_n)$ denote the position in which x_n was last encountered in x^{n-1}. With this notation, we thus write

$$\phi(x^n) = 2\phi(x^{n-1}) - \phi(x^{\ell-1}) \text{ if } x_n \preceq x^{n-1}.$$

Table 1. Results of Applying Equation 1 to $x = abacbbca = x^8$

i	0	1	2	3	4	5	6	7	8
x_i	λ	a	b	a	c	b	b	c	a
$\ell(x_i)$	0	0	0	1	0	2	5	4	3
$\phi(x^i)$	1	2	4	7	14	26	38	69	134

Note: Initialize $\phi(x^0 = \lambda) = 1$ and $\ell(x_i) = 0$ for all characters. Update ℓ after the application of Equation 1 to the current prefix to $\ell(x_i) = i$ before proceeding to the last character of the next prefix.

We use $\ell - 1$ since $\phi(x^{\ell-1})$ equals the number of new subsequences that arose as a result of elongating with $x_\ell = x_n$. So, now we have two rules, one valid when $x_n \not\prec x^{n-1}$ and one valid when $x_n \preceq x^{n-1}$ and since these two cases are mutually exclusive and exhaustive, we are done: Our algorithm is

$$\phi(x^n) = \begin{cases} 2\phi(x^{n-1}) & \text{if} \quad x_n \not\prec x^{n-1}, \\ 2\phi(x^{n-1}) - \phi(x^{\ell-1}) & \text{if} \quad x_n \preceq x^{n-1} \end{cases} \tag{1}$$

for $n > 0$ and we initialize by setting $\phi(x^0 = \lambda) = 1$. The only complication is that we have to keep track of the last position of the x_i in x^{n-1}. But this can be done ''on the fly'' by initially setting $\ell(a_i) = 0$ for all characters of the alphabet $\mathcal{A} = \{a_1, \ldots a_s\}$ and updating to $\ell(a_i) = k$ when we encounter a_i in the guise of $x_k = a_i$ before proceeding to the next prefix. We illustrate the full procedure again in Table 1 for the string $x = ababcbbca$.

The reader notes that for an n-long string, the algorithm requires exactly n steps to count at most 2^n objects. A more extensive discussion of the algorithm is in Appendix A, a rigorous proof of its correctness and related results can be found in Elzinga, Rahmann and Wang (2008).

Complexity and Durations

In practice, social science categorical time series often come as pairs $(x, \mathbf{t}_x)$ wherein x denotes the string of spells and $\mathbf{t}_x = (t_1, \ldots, t_n)$ denotes a vector of durations of the spells, namely, the times spent in the consecutive states. We say that $t(x) = \sum_i t_i$ equals the total duration of the whole series. In this section, the issue is how these durations can be incorporated in a quantification of variability or complexity. Unfortunately, now there are no general concepts like descriptive or combinatorial complexity to guide our thinking.

We will argue that the complexity of the pair $(x, \mathbf{t}_x)$ *decreases* when the variance of the durations increases. To justify this basic principle, we consider three quite different labor market careers, each covering a duration of 72 months (e.g., McVicar and Anyadike-Danes 2002):

$$x = (ue, (71, 1)),\ y = (ue, (36, 36)),\ z = (ue, (1, 71)).$$

Clearly, x is not much different from the simple $(u, (72))$ and z is very similar to the simple $(e, (72))$. On the other hand, y is, given total duration and the number of spells, maximally dissimilar from both $(u, (72))$ and from $(e, (72))$ and the variance of its durations is, again given total duration and number of spells, minimal. Therefore, we adhere to the principle that for fixed total duration and a fixed number of spells, the less variance of the durations of the subsequences, the more complex the time series is. However, we will deal with the variance of the durations of *all* 2^n subsequences, irrespective of whether these are distinct or not since the same subsequence may occur in several positions on the string and, because of these different positions, have several durations. For example, in $(abab, (1, 2, 3, 4))$, the subsequence ab occurs as x_1x_2, as x_1x_4, and as x_3x_4 with respective durations of 3, 5, and 7 units of time.

We will denote this variance of all subsequence durations as $V(\mathbf{t}_x)$ and discuss how to use it in a measure of complexity $C(x, \mathbf{t}_x)$ of the pairs $(x, \mathbf{t}_x)$. First, we demand that $C(x, \mathbf{t}_x)$ does not depend on the time scale employed in constructing the series. Second, $V(\mathbf{t}_x)$ has a lower and an upper bound and we do not want $C(x, \mathbf{t}_x)$ to depend on these bounds either. The lower bound $V_{min}(\mathbf{t}_x)$ will be attained when all durations of spells are equal and the upper bound $V_{max}(\mathbf{t}_x)$ will be attained when all but one of the spells have a duration of 1 and one spell has a duration of $t(x) - (n - 1)$ units of time. A solution for $C(x, \mathbf{t}_x)$ that respects these restrictions is

$$C(x, \mathbf{t}_x) = \log_2(\phi(x) \cdot T(x, \mathbf{t}_x)) \tag{2}$$

with

$$1 \leq T(x, \mathbf{t}_x) = \frac{V_{max}(\mathbf{t}_x) - V_{min}(\mathbf{t}_x) + 1}{V(\mathbf{t}_x) - V_{min}(\mathbf{t}_x) + 1}. \tag{3}$$

The interpretation of T is that of ''relative variance inverted'' since the difference of V from its lower bound V_{min} is divided by the maximum difference $V_{max} - V_{min}$ and this ratio has been inverted. The ''+1'' in the numerator prevents T to become 0 when $V_{max} = V_{min}$, namely, when there is just 1 spell and

Table 2. Family Formation Status Encoding

Code	Description
S(C)	Living Single (together with at least one child)
U(C)	Living together Unmarried (together with at least one child)
M(C)	Living together Married (together with at least one child))

Note: In total, there are six categories: A code is appended with "C" if the woman lived together with at least one child. For example, "UC" is used for living together unmarried with at least one child and "M" for living married without children.

the ''+1'' in the denominator prevents us from dividing by zero in case $V(\mathbf{t}_x) = V_{min}(\mathbf{t}_x)$, namely, when all spells have equal durations. So defined, $C(x, \mathbf{t}_x)$ satisfies, but not uniquely, all the demands that we put up and it is fully defined in terms of subsequence properties.

Complexity of Family Formation

The data that we use here come from the Family and Fertility Surveys (these data and their quality were amply described by Festy and Prioux (2002) and pertain to 6,066 American females born between 1950 and 1964 and assigned to one of three cohorts: those born in 1950-1954 ($N = 1{,}770$), 1955-1959 ($N = 2{,}148$), and 1960-1964 ($N = 2{,}148$). For each of these women, six distinct states (see Table 2) were used to describe the women's family formation status during 144 months, starting at age 18 and ending at age 30. For example, one of these women generated the pattern

$$\text{S}/60\ \text{U}/18\ \text{M}/7\ \text{MC}/59,$$

namely, the trajectory of a woman that lived single (S) during 60 months, then lived together unmarried (U) with a partner for 18 months, then married, perhaps to the same partner, and after another 7 months, gave birth to the first child. Then, she continued living married with one or more children (MC) for the remainder of the observation period.

On the average, trajectories have 3.35 spells ($sd = 1.71$) and this number ranges from 1 to 16.

To give the reader a feel for the kinds of trajectories in these data, we show those trajectories (ignoring durations) that were shared by at least 5 percent of the females in each cohort (Table 3). Trajectories that were shared by less than 5 percent were taken together in the category ''Various.'' The reader notes the decline of the traditional ''S M MC'' and the rise of unmarried cohabitation, exemplified by the trajectory ''S U M MC.''

Table 3. Family Formation Trajectories (Durations Ignored) Shared by at Least 5 Percent of the Females in Each Cohort

	1950-54	1955-59	1960-64
S	.086	.087	.083
S M	.064	.054	.059
S M MC	.284	.242	.195
S U M MC		.055	.072
Various	.566	.562	.591

Note: Trajectories shared by less than 5 percent of the females were taken together in the category "Various." See Table 2 for description of codes.

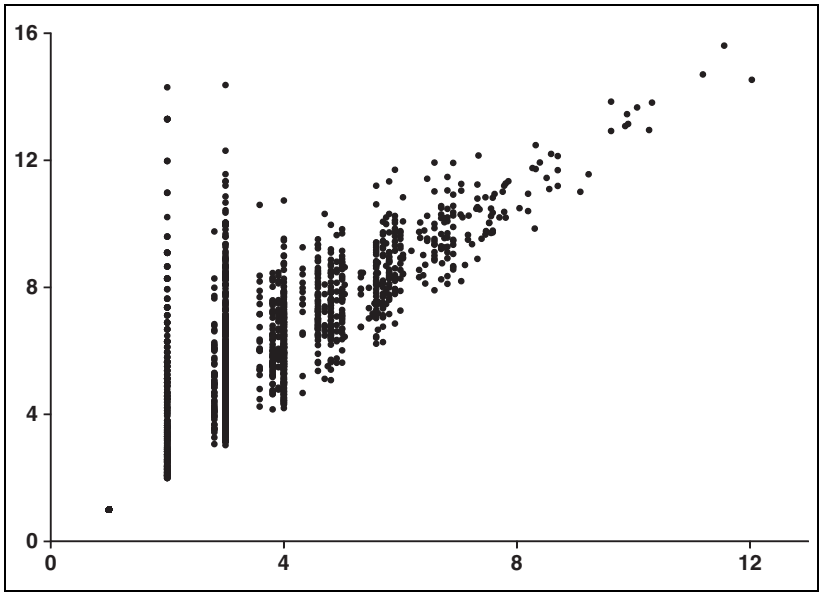

Figure 1. Plot of $C(x)$ (horizontal axis) versus $C(x, \mathbf{t}_x)$ (vertical axis) for the third cohort of U.S. females

To show the effect of including durations into the complexity measure, we show a plot of $C(x, \mathbf{t}_x)$ versus $C(x)$ for the third (youngest) cohort in Figure 1.

From the plot and the fact that Pearson's $r^2 = 0.62$, it appears that incorporating information on duration substantially adds variance to the quantification.

Table 4. Averages of $C(x)$ and $C(x, \mathbf{t}_x)$

Cohort	1	2	3
lo	3.05	3.18	3.36
$\overline{C}(x)$	**3.10**	**3.23**	**3.42**
up	3.16	3.28	3.48
lo	5.16	5.32	5.59
$\overline{C}(x, \mathbf{t}_x)$	**5.26**	**5.41**	**5.69**
up	5.37	5.51	5.78

Note: Also shown are the upper boundary *up* and lower boundary *lo* of the 90 percent BC_a confidence intervals of these parameters.

We calculated $C(x, \mathbf{t}_x)$ and $C(x)$ for all the histories and show the resulting averages per cohort in Table 4.

Furthermore, we bootstrapped 5,000 replications from our data and from these, calculated the 90 percent BC_a confidence intervals (e.g., Davison and Hinkley 1997). The upper and lower bounds of these intervals are also shown. As is apparent from the boundaries of the confidence intervals, the increase of $\overline{C}(x)$ between consecutive cohorts is significant since the confidence intervals of the oldest and the youngest cohort do not overlap. The size of the confidence intervals for $\overline{C}(x, \mathbf{t}_x)$ roughly doubles the size of the confidence intervals for $\overline{C}(x)$: It includes the extra variability that is introduced by incorporating durations as well. But still, there is a significant increase in complexity $\overline{C}(x, \mathbf{t}_x)$ between the first and the last cohort.

In Figure 2, we plotted the smoothed distributions of $\overline{C}(x, \mathbf{t}_x)$ as resulting from the bootstrapping. Apparently, these distributions are clearly separated.

As expected, these averages clearly show complexity to significantly increase over time.

Comments

Evidently, there is no substantive theory from which the coefficients $C(x)$ and $C(x, \mathbf{t}_x)$ derive; only plausible reasoning about the properties of categorical time series that contribute to their being more or less complex were used in their construction. So, the coefficients proposed are the outcome of an attempt to operationalize a notion that is quite difficult to capture.

Since we used the number of distinct subsequences $\phi(x)$ to construct $C(x)$, it seemed natural to use the variance of the durations of the embeddings $V(\mathbf{t}_x)$ to construct $C(x, \mathbf{t}_x)$. In fact the choice to use $V(\mathbf{t}_x)$ is arbitrary and one might argue in favor of straightforwardly using the variance of the state

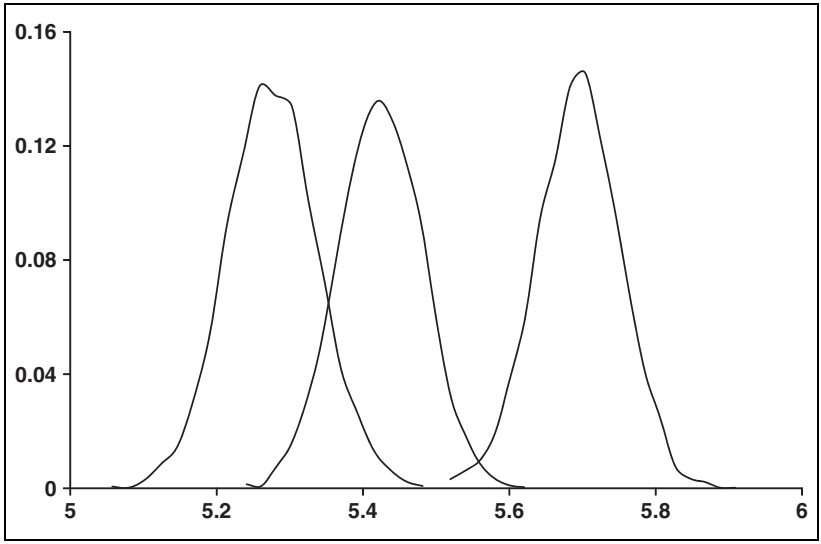

Figure 2. Smoothed distributions of $C(x, \mathbf{t}_x)$ from 5,000 bootstrapped samples for each of the three cohorts of U.S. females

durations to construct a conceptually and computationally simpler alternative to $C(x, \mathbf{t}_x)$. Thereto, we let s_t^2 denote the variance of the t_i and $s_{t,\max}^2$ its upper bound given $t(x)$. Then conceptually and computationally, a simpler alternative to $T(x, \mathbf{t}_x)$ would be

$$1 \leq T'(x, \mathbf{t}_x) = \frac{s_{t,\max}^2 + 1}{s_t^2 + 1}. \tag{4}$$

Remarkably, as is proven in Appendix B, we have that $T'(x, \mathbf{t}_x) = T(x, \mathbf{t}_x)$. This is a valuable result since it shows that we can simply calculate s_t^2 and $s_{t,max}^2$ in n steps instead of laboriously computing $V(\mathbf{t}_x)$ in 2^n steps, therewith retaining the feasibility of computations, even in case of extremely long series.

Our second comment pertains to the way one treats the durations in a categorical time series. It is not uncommon to write a time series $(ab, (3, 2))$ as *aaabb* and indeed, doing so hides the durations of the two spells by presenting five observations. So, processing *aaabb* instead of $(ab, (3, 2))$ makes life a lot simpler since calculating $C(x)$ instead of $C(ab, (3, 2))$ suffices. However, there are a number of reasons for not doing this.

Table 5. $\phi(x)$ as Calculated for the Extended Form of Three Strings and the Variances of the State-Durations

	ϕ	s_t^2
abbbbccc	40	0.22
aabbbccc	48	1.55
abcabcab	177	0.00

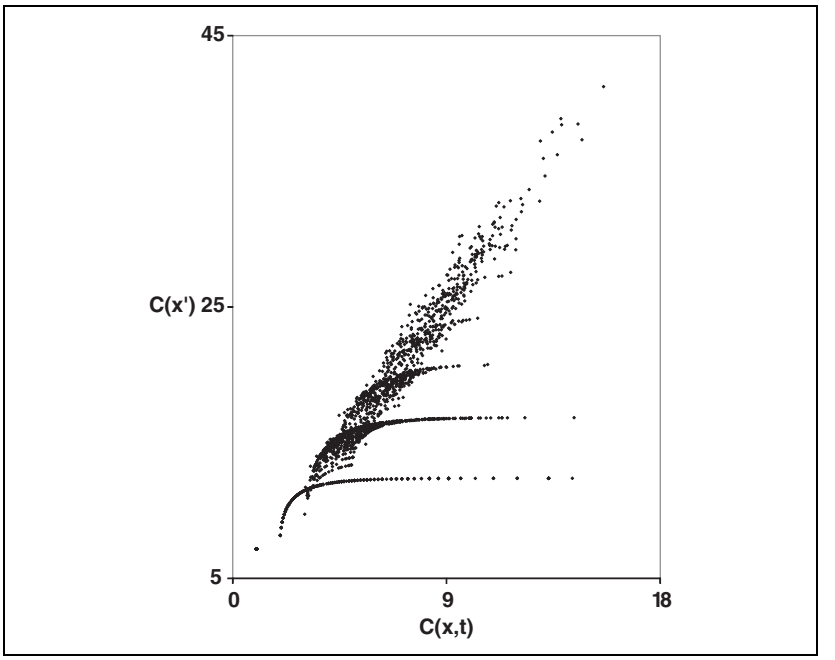

Figure 3. Plot of $C(x, \mathbf{t}_x)$ (horizontal axis) versus $C(x')$ (vertical axis) where the x' denote the extended form of the trajectories of the third cohort of U.S. females

First, when we are discussing and theorizing about life courses or careers, we are in fact discussing spells of cohabitation, of being unemployed or imprisoned and the durations are a property of these spells. By switching to the extended *aaabb*, we are no longer analyzing the spells.

The second reason not to switch to the extended form is illustrated in Table 5. Table 5 illustrates that strings with many adjacent identical characters have relatively small numbers of distinct subsequences. Furthermore, it illustrates that relatively big changes in the compositions of such string only have

a small effect on $\phi(x)$ whereas such changes cause appreciable differences in the variance of the spell durations. These properties of ϕ and s_t^2 in fact generate what we see in Figure 3. In this figure we plotted, for the third cohort of U.S. females, the values of $C(x, \mathbf{t}_x)$ against $C(x')$ (vertical axis) where the x' denote the sequences in extended form. Apparently, lots of sequences that are well discriminated by $C(x, \mathbf{t}_x)$ are poorly, if at all, discriminated by $C(x')$.

However, the results of applying $C(x)$ and $C(x, \mathbf{t}_x)$ demonstrate that these coefficients are stable and sensitive enough to pick up relevant aspects of the data. So, we can start using them as dependent or independent variables in modeling categorical time series. Appropriate software is available as a free package CHESA[1] from the author and as TraMineR[2], written by Gabadinho, Ritschard, Studer, and Müller (2008).

Declaration of Conflicting Interests

The author declared no conflicts of interest with respect to the authorship and/or publication of this article.

Funding

The author received no financial support for the research and/or authorship of this article.

Notes

1. Downloadable from http://home.fsw.vu.nl/ch.elzinga/.
2. Downloadable from http://mephisto.unige.ch/traminer/.

References

Abbott, Andrew, 1995. ''A Comment on 'Measuring the Agreement Between Sequences.''' *Sociological Methods & Research* 24:232-42.

Abbott, Andrew and Angela Tsay. 2000. ''Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect.'' *Sociological Methods & Research* 29:3-33.

Arnett, Jeffrey J. 2000. ''Emerging Adulthood. A Theory of Development from the Late Teens through the Twenties.'' *American Psychologist* 55:469-80.

Blau, Peter M. 1977. *Inequality and Heterogeneity*. New York: Free Press.

Bras, Hilde, Aart C. Liefbroer, and Cees H. Elzinga. Forthcoming. ''Standardization of Pathways to Adulthood? An Analysis of Dutch Cohorts Born Between 1850 and 1900.'' *Demography*.

Buchmann, Marlis. 1989. *The Script of Life in Modern Society: Entry into Adulthood in a Changing World*. Chicago: University of Chicago Press.

Chase, Phillip J. 1976. ''Subsequence Numbers and Logarithmic Concavity.'' *Discrete Mathematics* 16:123-40.

Cover, Thomas M. and Joy A. Thomas. 1991. *Elements of Information Theory*. New York: John Wiley.

Davison, Anthony C. and David V. Hinkley. 1997. *Bootstrap Methods and Their Applications*. Cambridge, UK: Cambridge University Press,.

de Luca, Aldo. 1999. ''On the Combinatorics of Finite Words.'' *Theoretical Computer Science* 218:13-39.

Elzinga, Cees H. 2005. ''Combinatorial Representation of Token Sequences.'' *Journal of Classification* 22:87-118.

Elzinga, Cees H. and Aart C. Liefbroer. 2007. ''De-Standardization and Differentiation of Family Life Trajectories of Young Adults: A Cross-National Comparison Using Sequence Analysis.'' *European Journal of Population* 23:225-50.

Elzinga, Cees, Sven Rahmann, and Hui Wang. 2008. ''Algorithms for Subsequence Combinatorics.'' *Theoretical Computer Science* 409:394-404.

Ferenczi, Sebastien and Zoltán Kása. 1999. ''Complexity for Finite Factors of Infinite Sequences.'' *Theoretical Computer Science* 218:177-95.

Festy, Patrick and France Prioux. 2002. *An Evaluation of the Fertility and Family Surveys Project*. New York: United Nations.

Flaxman, Abraham, Aram W. Harrow, and Gregory B. Sorkin. 2004. ''Strings with Maximally Many Distinct Subsequences and Substrings.'' *The Electronic Journal of Combinatorics* 11:8.

Gabadinho, Alexis, Gilbert Ritschard, M. Studer, and Nicolas S. Müller. 2008. *Mining Sequence Data in R with the TraMineR Package: A User's Guide*. Department of Econometrics and Laboratory of Demography, University of Geneva. Retrieved January 22, 2010 (http://mephisto.unige.ch/traminer/)

Gibbs, Jack P. and Dudley L. Poston Jr. 1975. ''The Division of Labor: Conceptualization and Related Measures.'' *Social Forces* 53:468-76.

Gini, Corrado. 1921. ''Measurement of Inequality and Incomes.'' *The Economic Journal* 31:124-26.

Iványi, Antal. 1987. ''On the d-Complexity of Words.'' *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös nominatae*, *Sectio Computatorica* 8:69-90.

Janson, Svante, Stefano Lonardi, and Wojciech Szpankowski. 2004. ''On Average Sequence Complexity.'' *Theoretical Computer Science* 326:213-27.

Kása, Zoltan. 1998. ''On the d-Complexity of Strings.'' *Pure Mathematics and Applications* 9:119-28.

Kolmogorov, Andrej N. 1965. ''Three Approaches to the Quantitative Definition of Information.'' *Problems of Information Transmission* 1.

Lesthaeghe, Ron J. 1995. ''The Second Demographic Transition in Western Countries: An Interpretation.'' Pp. 17-62 in *Gender and Family Change in Industrialized Countries*, edited by K. O. Mason and A-M. Jensen. Oxford, UK: Clarendon Press.

Levé, Florence and Patrice Séébold. 2001. ''Proof of a Conjecture on Word Complexity.'' *Bulletin of the Belgian Mathematical Society Simon Stevin* 8:277-91.

Li, Ming and Paul Vitányi. 2008. *An Introduction to Kolmogorov Complexity and Its Applications* (3rd Edition). New York: Springer.

McVicar, Duncan and Michael Anyadike-Danes. 2002. ''Predicting Successful and Unsuccessful Transitions from School to Work by using Sequence Methods.'' *Journal of the Royal Statistical Society*. *SeriesA* 165:317-34.

Mills, Melinda. 2004. ''Stability and Change: Partnership Histories in Canada, The Netherlands and the Russian Federation.'' *European Journal of Population* 20:141-75.

Sankoff, David and Joseph B. Kruskal, eds. 1983. *Time Warps, String Edits and Macro-Molecules. The Theory and Practice of String Comparison*. Reading, MA: Addison-Wesley.

Shanahan, Michael J. 2000. ''Pathways to Adulthood: Variability and Mechanisms in Life Course Perspective.'' *Annual Review of Sociology* 26:667-92.

Shannon, Claude E. 1948. ''A Mathematical Theory of Communication.'' *The Bell System Technical Journal* 27:379-423, 623-56.

Shawe-Taylor, John and Nello Christianini. 2004. *Kernel Methods for Pattern Recognition*. Cambridge, UK: Cambridge University Press.

Simpson, Edward Hugh. 1949. ''Measurement of Diversity.'' *Nature* 163:688.

Theil, Henri. 1972. *Statistical Decomposition Analysis*. Amsterdam, The Netherlands: North-Holland Publishing Company.

Uhlenberg, Peter R. 1969. ''A Study of Cohort Life Cycles: Cohorts of Native Born Massachusetts Women, 1830-1920.'' *Population Studies* 23:407-20.

Uhlenberg, Peter R. 1974. ''Cohort Variations in Family Life Cycle Experiences of U.S. Females.'' *Journal of Marriage and the Family* 36:284-92.

Wang, Hui and Zhiwei Lin. 2007. ''A Novel Algorithm for Counting All Common Subsequences.'' Pp. 502-05 in *2007 IEEE Conference on Granular Computing*. Washington, DC: IEEE.

Wilcox, Allen R. 1967. ''Indices of Qualitative Variation.'' Technical Report ORNL-TM-1919, Oak Ridge National Laboratory. Retrieved January 22, 2010 (http://www.ornl.gov/info/reports/1967/3445605133753.pdf)

Bio

Cees Elzinga is head of the Department of Social Science Research Methods of the Faculty of Social Science of the VU University in Amsterdam, The Netherlands. He is especially interested in historical demography, life course research, categorical time series, and combinatorial computer science. Recent publications appeared in the *European Journal of Population* and *Theoretical Computer Science*.