Journal of Classification 22:87-118 (2005)
DOI: 10.1007/s00357-005-0007-6

Combinatorial Representations of Token Sequences

Cees H. Elzinga

Vrije Universiteit, Amsterdam

Abstract: This paper presents new representations of token sequences, with and without associated quantities, in Euclidean space. The representations are free of assumptions about the nature of the sequences or the processes that generate them. Algorithms and applications from the domains of structured interviews and life histories are discussed.

Keywords: Sequence classification; Sequence representation; Sequence analysis; Dynamic programming.

1. Introduction

Token or state sequences are a quite common kind of data, not only in the behavioral sciences but also in other fields such as e.g molecular biology or ethology. Such data then come in the form of a matrix like the one presented below

$$\begin{pmatrix} \alpha & \beta & \gamma & \cdots & \cdots & \beta & \delta & & & \\ \delta & \alpha & & & & & & & & \\ \gamma & \delta & \zeta & \beta & \alpha & & & & & \\ \alpha & \varepsilon & \gamma & \delta & \varepsilon & \delta & \cdots & \cdots & \cdots & \zeta \\ \vdots & & & & & & & & & \\ \vdots & & & & & & & & & \end{pmatrix},$$

Author's Address: Vrije Universiteit, Department of Social Research Methodology, Faculty of Social Sciences, De Boelelaan1081c, 1081 HV Amsterdam, Tel: (+)31 20 598 6889, Fax: (+)31 20 589 6860, e-mail: ch.elzinga@fsw.vu.nl

where each row is a sequence of tokens from a finite alphabet and the tokens are acronyms for disjoint classes of events. Typical examples from the behavioral sciences are encoded transcripts of interviews or life or employment histories; in molecular biology, the symbols in the rows are typically the amino acids A, C, T and G and each row represents (a part of) the DNA of a specific species. In ethology, the tokens could represent different kinds of movements or phrases of song produced by birds engaged in mating or defending their territory. In many instances of such matrices, e.g. when the rows represent life histories, the tokens are associated with a quantity. If the rows represent life histories, that quantity will normally represent the duration of a particular state and the rows of the data matrix will have the form of

$$\left(\begin{array}{ccccccc} \alpha_p & \beta_q & \gamma_r & \cdots & \cdots & \beta_u & \delta_v \end{array} \right),$$

where each subscript represents a positive number of time units. In ethology, the associated quantities could stand for the frequencies of repetition of different kinds of behavior or sound levels of song phrases. In analyzing such matrices, two kinds of questions can be posed. The first, traditional one, is the question of what process or mechanism generated the sequences observed. A model for such data considers each token on its own and the model is supposed to reproduce the sequential character of the data. The second type of question one might raise, is the question of how to classify the objects that produced the sequences, each class supposedly generating its own, typical sequence. If this is indeed the question, then one considers each sequence, instead of each token, as one datum and the first challenge is to find a way to describe these data in such a way that they become amenable to a method of classification. This paper tries to meet this challenge.

Comparing sequences and measuring their distances or similarities is quite common amongst microbiologists and those involved in electronic data transmission. Probably the best known way of mapping equally long sequences into a metric space, is by measuring the Hamming distance between pairs of sequences: the number of positions in which the sequences differ. Hamming distance first arose in electronic data transmission (Hamming 1950) where bit strings are embedded in longer strings (Hamming codes) to facilitate error detection and correction after the string has been transmitted over a noisy channel. The Hamming distance is closely related to the well known Minkowski L_1-distance and has been used in the multivariate analysis of binary data (e.g. Heiser and Meulman 1997). Generalizations of the Hamming distance have been used in the classification of medical syndromes; e.g. Bezem and Keijzer (1997). Transferring the concept to the present context, consider the example sequences given as follows:

$$\begin{aligned} x &= [\alpha, \beta, \gamma, \delta], \\ y &= [\varepsilon, \alpha, \beta, \gamma], \\ z &= [\mu, \nu, \xi, \pi]. \end{aligned}$$

The Hamming distance between these three pairs of sequences equals 4: the sequences have no common tokens in identical positions. However, if these sequences represented bird song phrases, encoded with a musically meaningful alphabet, the ornithologist would probably be disappointed by these distances. For the first two phrases represented would probably sound quite similar and each of them would sound very different from the third one. Yet, their Hamming distance is the same. Similar disappointment would arise if these sequences would have represented strands of DNA, the tokens referring to nucleotides, or employment histories, the tokens referring to different types of jobs. Basically, this disappointment arises from the fact that the Hamming distance is insensitive to common precedence. Let us write $\alpha \gg \beta$, precisely when token α precedes token β. Then the first two sequences exhibit the relations $\alpha \gg \beta \gg \gamma$, which is why the song phrases represented sound similar. This commonality of precedences is of course of no relevance in data transmission: it is irrelevant whether [101101] or [101001] was received when [010010] was sent. Similarly, in comparing patients with different symptom patterns, it is irrelevant in which order the symptoms appear in the list: only their presence or absence counts. So in the classification of symptom patterns, Hamming distance is only natural since the lists of symptoms are considered as sets, not as sequences wherein the order reflects a temporal or spatial pattern. In many other applications, it is precisely this (lack of) commonality of the represented spatial or temporal distribution of events that is of interest and should be reflected by the measured distance or similarity between the sequences. This led to attempts to include commonality of precedences in a distance measure. A very appealing idea is, to ”align” the sequences. Again, consider the above example sequences. The first two can be aligned by inserting a gap-symbol ”$-$” into both sequences, thus creating the transformed sequences

$$\begin{aligned} x' &= [-, \alpha, \beta, \gamma, \delta], \\ y' &= [\varepsilon, \alpha, \beta, \gamma, -]. \end{aligned}$$

Similar transformations are possible by deleting and/or inserting tokens. Counting the minimum number of such transformations, necessary to obtain perfect alignment of the two sequences, leads to to a metric called the edit or Levenshtein distance (of which the Hamming distance is a special case: it just counts deletions). Descriptions of Levenshtein's algorithm can be found in e.g. Gusfield (1997) or Clote and Backofen (2000). The Levenshtein distances between the three example sequences are given by

$$\begin{array}{c} x \\ y \\ z \end{array} \left(\begin{array}{ccc} 0 & & \\ 2 & 0 & \\ 4 & 4 & 0 \end{array} \right).$$

These distances were computed on the basis of the arbitrary assumption that the weight or cost of either edit operation on any character is the same. Indeed, to the ornithologist, this would be a more satisfying solution than the Hamming distances. A slight but important generalization of the Levenshtein distance allows for differentially weighing or costing the different edit operations of deleting, inserting or substituting a character. Methods based upon the Levenshtein distance and its generalizations have come to be known as O(ptimal) M(atching) methods. It is precisely this generalization that made the Levenshtein distance appealing to microbiologists; an appeal that was boosted by the classical paper of Needleman and Wunsch (1970) on a feasible algorithm for arbitrary gap weight and Gotoh's (1982) paper on affine groups of gap weight functions. This appeal was caused by the fact that biologists were able to formulate biological problems or models in terms of suitable cost functions for the edit operations and gap handling, i.e. the geometries implied by the use of OM-methodology were considered as acceptable renderings of biochemical or phylogenetical models. Some 20 years ago, Abbott and Forrester (1986) first introduced OM into the behavioral sciences and Abbott and Tsay (2000) presented a detailed overview of its diverse applications since then. However, the use of OM-methods in behavioral science applications received quite some criticisms (e.g. Dijkstra and Taris 1995; Wu 2000; Elzinga 2003). To illustrate these objections, we cite an example taken from Dijkstra and Taris (1995): they consider the sequences (in our notation):

$$\begin{aligned} x &= [\alpha, \alpha, \alpha, \beta, \gamma, \delta], \\ y &= [\beta, \gamma, \delta, \varepsilon, \varepsilon, \varepsilon], \\ z &= [\tau, \tau, \tau, \tau, \tau, \tau]. \end{aligned}$$

The Levenshtein distances between each pair of these sequences equals 6, which is the maximum distance for sequences of length 6. Dijkstra and Taris (1995) object to the use of edit distance because sequences x and y should be closer to each other than to z since (pp. 216) ”...they share as much as three elements, and in the same order.” This criticism illustrates that the geometry implied by an OM-metric does not necessarily reflect the theoretical notions of the research area in which it is applied. Indeed, the main objection to the transfer of OM-methods to the behavioral sciences has been, that the implied geometries have no empirical interpretation. Furthermore, neither the Hamming metric nor the OM-methods can handle the quantities that are, in the behavioral sciences, often associated to the sequence events.

Therefore, the present paper sets out to construct a representation of token sequences in a metric space, such that the resulting representation is amenable to the majority of today's classificatory techniques and such that it handles associated quantities in a natural and flexible manner. As will become apparent (see Table 1), this representation will lead to a metric that is quite different from from the Hamming-metric and the metrics employed by OM-methods.

2. Representations

This section treats the formal aspects of creating representations for sequences with and without associated quantities. We start with a quite detailed discussion of the representation of simple sequences, i.e. sequences of tokens that have no associated quantities. Later, we will slightly modify the representation in order to allow for representing the associated quantities as well. Let $A = \{\alpha, \beta, \gamma, \ldots\}$ with $|A| > 0$ be a finite alphabet of tokens. A token sequence x is a finite, ordered string of tokens from A. We say that the sequence has length l_x if x has l_x positions occupied by tokens from A. We write $x = [\alpha, \gamma, \mu, \ldots]$ or $x = [\alpha_1, \alpha_2, \ldots]$ with $\alpha_i \in A$. To indicate that a particular token μ occurs in x, we write $\mu = [\mu] \subset_1 x$. Furthermore, we define an empty sequence $\theta = []$. Note that one and the same token may occur more than once in the same sequence as, for example, in $x = [\alpha, \beta, \alpha, \gamma]$ with $l_x = 4$.

Naturally, if $x = [\alpha_1, \alpha_2, \ldots, \alpha_k]$ and $y = [\beta_1, \beta_2, \ldots, \beta_k]$, we define $x = y$ precisely when $\alpha_i = \beta_i$ for all $1 \leq i \leq l_x = l_y$. Given two sequences x and y with $l_x \geq l_y$, we say that y is a subsequence of x, precisely when all tokens from y appear in x and in the same order, i.e. if α precedes β in y, then α precedes β in x too. We then write $y \subset_{l_y} x$ and, of course, $x \subset_{l_x} x$. These definitions of identity and subsequence are natural and direct. Note however, that a particular subsequence $y \subset_j x$ might be embedded in x in several ways. In our example sequence $x = [\alpha, \beta, \alpha, \gamma]$, $[\alpha, \gamma] \subset_2 x$ is embedded in two different ways. It will prove useful to specify and enumerate subsequences in a somewhat less direct way. Thereto, we write X for the set of all finite sequences that are constructable from A and $X_k = \{x \in X | l_x = k\} \subset X$. Furthermore, let P^k be the set of all binary strings of length k: $\mathsf{P}^k = \{\mathsf{p}_i^k\}_{i=1}^{2^k}$ with $\mathsf{p}_i^k = (\mathsf{p}_{i,1}, \ldots, \mathsf{p}_{i,k})$ and $\mathsf{p}_{i,j} \in \{0, 1\}$ for all $1 \leq j \leq k$. Consider the p_i^k as projections: let p_i^k be any such binary string with $\sum_j \mathsf{p}_{i,j} = n \leq k$ and $x = [x_1, \ldots, x_k]$, then $\mathsf{p}_i^k(x) = v \in X$ such that $v \subset_n x$ and $\alpha_j \subset_1 v$ if and only if $\mathsf{p}_{i,j} = 1$. Thus, for each subsequence $y \subset_j x \in X_k$, there is at least one

Table 1. Distance matrices for 6 sequences as indicated in the first column. The distances in the matrix ”Euclidean” represent the Euclidean distances between the vector according to the representation proposed in this paper.

	Euclidean						Hamming						Levenshtein					
$[\alpha,\beta,\gamma,\delta]$	–						–						–					
$[\alpha,\beta,\delta,\gamma]$	2.8	–					2	–					2	–				
$[\beta,\alpha,\delta,\gamma]$	3.7	2.8	–				4	2	–				3	2	–			
$[\beta,\delta,\alpha,\gamma]$	4.0	3.5	2.8	–			4	3	2	–			4	2	2	–		
$[\delta,\beta,\alpha,\gamma]$	4.2	4.0	3.4	2.8	–		3	2	3	2	–		3	2	2	2	–	
$[\delta,\gamma,\beta,\alpha]$	4.7	4.5	4.2	4.0	3.5	–	4	4	4	4	3	–	4	4	4	4	4	–

p_i^k such that $p_i^k(x) = y$. At least one, since some of the tokens of x may repeat in different positions in x: for example, we can generate $[\alpha,\gamma] \subset_2 x = [\alpha,\beta,\alpha,\gamma]$ with $p_1^4 = (1,0,0,1)$ and with $p_2^4 = (0,0,1,1)$. So, the concept of a subsequence can also, but less directly, be defined by applying a projection. Therefore, the enumeration of the different embeddings of a particular subsequence v in a given sequence x is equivalent to the enumeration of the different projections p_i^k defined by $p_i^k(x) = v$. To implement this, we define, for each $x \in X_k$ and all $v \in X_j$ with $1 \leq j \leq k$, the equivalence sets $\mathcal{E}_{v,x}^k = \left\{p_i^k | p_i^k(x) = v\right\}$ and we use the cardinalities $g_{x,k}(v) = |\mathcal{E}_{v,x}^k| \geq 0$ of these sets to enumerate the embeddings of v in x. Combinatorially, the $g_{x,k}$ are multisets (e.g. Stanley 1997) on x. We adopt the convention $l_v > k \Rightarrow g_{x,k}(v) = 0$. Thus, for our example sequence, we have $g_{x,2}([\alpha,\gamma]) = 2$, $g_{x,2}([\beta,\gamma]) = 1$ and $g_{x,2}([\gamma,\alpha]) = 0$. Now we adopt the convention to index the possible elements of X_j in a lexicographic manner, according to the order of the tokens in the alphabet A. So, if $A = \{\alpha,\beta,\gamma\}$, the first element of X_2 would be $[\alpha,\alpha]$ and the 6th element of X_2 would be $[\beta,\gamma]$. For an arbitrary sequence $y \in X$, we can now construct objects, for each $j \geq 1$,

$$y_j = \left(g_{y,j}(x_1),\ldots,g_{y,j}\left(x_{|A|^j}\right)\right) \in \boldsymbol{X}_j \subset \mathbb{N}^{|A|^j},$$

where $\left\{x_1,\ldots,x_{|A|^j}\right\} = X_j$ and $\mathbb{N}$ denotes the natural numbers. For our example sequence $x = [\alpha,\beta,\alpha,\gamma]$ we thus have $x_2 = (1,1,2,1,0,1,0,0,0)$. A vector that completely determines a sequence x of length k is then a vector $\boldsymbol{x}$, the first $|A|$ coordinates of which form the image of $g_{x,1}$, the next $|A|^2$ coordinates are the values of $g_{x,2}$, etc. and the $\left(\sum_{j=1}^k |A|^j + 1\right)^{\mathrm{th}}$ coordinate and all subsequent coordinates are set to 0 since $l_v > k \Rightarrow g_{x,k}(v) = 0$. A vector space $\boldsymbol{X}$ representing X is now easily established by defining the negative $-\boldsymbol{x}$, scalar multiplication and vector addition in the usual way and having the empty sequence $\theta \equiv [\,]$ represented by the zero-vector $\boldsymbol{0}$. This vector space then becomes a pre-Hilbert space by defining the inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \equiv \sum_i x_i \cdot y_i$ where the x_i and y_i now denote the coordinates of $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, and a norm as $\|\boldsymbol{x}\| \equiv \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. Note that each set X_j can analoguously be represented in a vector space $\boldsymbol{X}_j = \{\boldsymbol{x}_j\}$.

Admittedly, the representation suffers from obesity, in the sense that there are very many vectors in $\boldsymbol{X}$ that do not represent any constructable sequence. On the other hand, the representation chosen, ensures that the representation of each particular x is unique. This uniqueness is almost trivially established: for a sequence x with $l_x = k$, there is only one projection in the equivalence set $\mathcal{E}^k_{x,x}$, so if x and y with $l_x = k$ and $l_y = k'$ are different, the sets $\mathcal{E}^k_{x,x}$ and $\mathcal{E}^{k'}_{y,y}$ are different and each x is directly recoverable from $\mathcal{E}^k_{x,x}$ and A. Furthermore, the only property of the sequences used to represent them, is their orderedness: we discern a sequence from a collection of the same tokens by the fact that, in the sequence, certain tokens precede specific other tokens. Listing and counting all the precedences is what is in fact accomplished by the projections and their equivalence sets. Therefore, it seems difficult to represent a more basic property of sequences; at the same time, and for the same reason, it seems hard to represent other properties without including assumptions about the origin or the generation of the sequences. Table 1 shows, for the sequences indicated in the table, the Euclidean distances between the sequence representations as proposed here, and the distances according to the Hamming and the Levenshtein metric. Important is, that the order relations between the distances in each matrix differ from those in each of the other matrices. Note that the method proposed here produces columns that are strictly increasing, contrary to the first two columns in the other two matrices. This implies that the representation proposed sometimes reverses the order of distances when compared with the Hamming metric or the Levenshtein metric.

We now turn to the representation of sequences that have associated quantities. As a running interpretation, we will assume that these quantities represent the time spent in the various states in the sequence. With this interpretation in mind, we see that, with each sequence x with length l_x, there is a vector $\boldsymbol{t}_x = (t(1), \ldots, t(l_x))$ with positive, real valued $t(i)$ representing these quantities. In the representation of a simple sequence, coordinates of $\boldsymbol{x}$ are plain cardinalities of the equivalence sets $\mathcal{E}^k_{v,x}$, i.e. each projection in these sets is assigned an equal weight of 1. Projections represent tuples from x. A straightforward, but arbitrary, way to represent associated time is to assign weights to the tuples, i.e. to the projections, according to the time spent in each state of the tuple. This is easily accomplished by writing the projections as column vectors and defining the multisets

$$g^*_{x,k}(v) = \sum_{\mathsf{p}^k \in \mathcal{E}^k_{v,x}} \boldsymbol{t}_x \cdot \mathsf{p}^k. \tag{1}$$

For example, in the sequence $x = [\alpha_1, \beta_3, \alpha_6, \gamma_2]$, we have $y = [\alpha, \gamma] \subset_2 x$, $\{(1,0,0,1),(0,0,1,1)\} = \mathcal{E}^2_{y,x}$ and $\boldsymbol{t}_x = (1,3,6,2)$ so $g_{x,2}(y) = 3+8$.

As will appear in the next sections, this way of representing time has a number of attractive properties which are nicely expressable in analytical terms, distances have a clear interpretation in terms of sequence similarity and actually constructing the representation is algorithmically feasible. However, it will turn out that there is no clearcut interpretation of vector length like there is in the representation of simple sequences. Of course, (1) can be easily modified by introducing some nondecreasing transformation on its inner sum but we will not dwell upon this. A modification of (1) that seems appealing is

$$g^*_{x,k}(v) = |\mathcal{E}^k_{v,x}|^{-1} \sum_{\mathsf{p}^k \in \mathcal{E}^k_{v,x}} \boldsymbol{t}_x \cdot \mathsf{p}^k. \tag{2}$$

In (2), one computes the average of the times spent in equivalent tuples. Analytically, the properties of (2) are not too difficult to describe and now vector length does have a clear interpretation. But algorithmically, constructing a representation with (2) is not feasible because of the colossal task of enumerating the equivalence sets.

A quite different incorporation of associated quantities could arise from the following considerations. Imagine two individuals, both having been unemployed for two periods of six months each. Hence, the employment history sequence of both individuals would certainly contain the subsequence $[u_6, u_6]$, u standing for being unemployed. Now suppose that the complete employment histories of these individuals would be $[u_6, e_1, u_6]$ and $[u_6, e_{100}, u_6]$, the token e denoting employedness. Most of us will agree that the economical, sociological and psychological difference between these sequences is quite significant. Apparently, the fact that $[u_6, e_1, u_6]$ spans 13 months for the first individual and 112 months for the second individual is decisive. Such considerations could lead to measuring time spent in a tuple as the total time trajectory that starts from the onset of the first state in the tuple and ends with the end of the last state of the tuple. This can be formalised by introducing a transformation $f\left(\mathsf{p}^k_i\right) = \mathsf{q}^k_i = (\mathsf{q}_{i,1}, \ldots, \mathsf{q}_{i,k})$ with $\mathsf{q}_{i,j} = 1$ precisely when $\min_{1 \le j \le k}\{j|\mathsf{p}_{i,j} = 1\} \le j \le \max_{1 \le j \le k}\{j|p_{i,j} = 1\}$ and writing

$$g^*_{x,j}(v) = \sum_{\mathsf{p}^k \in \mathcal{E}^k_{v,x}} \boldsymbol{t}_x \cdot f(\mathsf{p}^k). \tag{3}$$

Neither algorithmically nor analytically, (3) poses big problems; the real problem with this representation is that it maps different sequences onto one and the same vector. Using (3) on sequences like for example $x = [\alpha_q, \beta_p, \alpha_q, \beta_{2p}, \alpha_q]$ and $y = [\alpha_q, \beta_{2p}, \alpha_q, \beta_p, \alpha_q]$ results in $\boldsymbol{x} = \boldsymbol{y}$ as a consequence of measuring ”spanned time” instead of ”occupied time”. It is hard to think of a substantial theory on the data that would justify such mappings of different sequences onto the same point. So, in the sequel, we will restrict ourselves to studying and applying the representation of simple sequences and the representation specified by (1) for sequences with associated quantities. Note that producing a table that is analoguous to Table 1 is not possible since OM-methods cannot handle quantified sequences.

3. Principles of Algorithms

Because of the colossal number of coordinates that results from the representation of a sequence of even moderate length and constructed from a fairly limited alphabet, calculations with such a vector are not practically feasible. For example, writing out in full a vector representing a sequence of length 10, constructed from an alphabet consisting of only 20 tokens, would require $\sum_{i=1}^{20} 10^i > 10^{13}$ figures to write down. Directly calculating quantities like $\|\boldsymbol{x}\|^2$ or $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ is therefore sheer impossible. In this section we will discuss the basic idea of algorithms that do allow for such calculations within a reasonable time; for representations of simple sequences this reasonable time will even appear to be third order polynomial time. These algorithms are not only a prerequisite for the applicability of the described representations; as will become apparent in the next sections, the principle of these algorithms is also very useful when studying properties of the representations. From the previous section, it is clear that the general expression that determines the coordinates of a representing vector is of the form

$$g_x(v) = f(\mathcal{E}^k_{v,x}, \boldsymbol{t}_x). \tag{4}$$

Since $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_i x_i y_i = \sum_i g_x g_y$, where x_i and y_i now denote the coordinates of $\boldsymbol{x}$ and $\boldsymbol{y}$, it is immediate that, in the case of simple sequences, $\sum_i g_x g_y$ enumerates the number of matches obtained when each and every i-tuple from x is compared with each and every i-tuple from y. If x and y are time-coupled sequences, then, if there is a match between a tuple from x and a tuple from y, the properties of f determine the contributions of these tuples to the total of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. So, in this section, we limit our discussion of algorithms to enumerate matching tuples from a pair of, not necessarily different, sequences x and y. The major algorithms and the most important optimisations are discussed in Appendix A to this paper.

To begin with we define, for each pair of sequences x and y with lengths l_x and l_y, an $(l_x \times l_y)$-matrix $\boldsymbol{E}_{x,y} = \{e_{x,y}(i,j)\}$ such that $e_{x,y}(i,j) = 1$ if and only if the i^{th} token from x is identical to the j^{th} token from y and $e_{x,y}(i,j) = 0$ in all other cases (in the sequel, we drop the subscripts x and

y whenever possible). Obviously, if $x = y$, $\boldsymbol{E}$ is symmetric around the main diagonal and $e(i,i) = 1$ for $1 \leq i \leq l_x$. Next, we define an n-path $\wp(i,j) = [\![e(i,j), e(k,l), \ldots]\!]$ as an ordered n-tuple of positive elements of $\boldsymbol{E}$ such that, for $i \neq k$ and $j \neq l$, $e(k,l) \in \wp(i,j) \Leftrightarrow i < k, j < l$ and such that, whenever $e(k,l), e(n,m) \in \wp(i,j)$, either $k > n$ and $l > m$ or $n > k$ and $m > l$. We will say that n is the length of a path $\wp$ if $\wp$ is an n-path. Please note, that, if x and y are of the same length, the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by $l_x - \sum_i e(i,i)$. Now for each path in $\boldsymbol{E}$ there exists an n-tuple in x and an n-tuple in y that consist of the same tokens and the same precedences and, conversely, for all pairs of matching n-tuples from x and y, there exists a unique n-path in $\boldsymbol{E}$. Hence, finding and enumerating matching n-tuples from x and y is equivalent to finding and enumerating n-paths in $\boldsymbol{E}$. A simple dynamic algorithm that enumerates all paths in $\boldsymbol{E}$ is easily constructed: let $a_{i,j}$ denote the number of paths of which the first element $e(i,j) = 1$. Obviously, if $e(i,j) = 1$, we must have $a_{i,j} \geq 1$, equality holding when $i = l_x$ and/or $j = l_y$. So, we have the recursion

$$a_{i,j} = 1 + \sum_{q>i, r>j} a_{q,r}, \tag{5}$$

hence, for simple sequences, we have $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i,j} a_{i,j}$. However, this algorithm is not very efficient; details of this and other algorithms are discussed in Appendix A.

4. Boundaries on $\|\boldsymbol{x}\|^2$

Which are the properties of the representations chosen? One way to gain insight into this question is to study the behaviour of certain quantities under extreme conditions, i.e. how extreme sequences are represented. Therefore we investigate the boundaries of $\|\boldsymbol{x}\|^2$ and, in a later section, of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ in the unit sphere.

In this section we discuss boundaries on $\|\boldsymbol{x}\|^2$, given some fixed sequence length l_x and, for time-coupled sequences, a total time trajectory $L_x = \sum_i t_x(i)$ of some fixed size under the assumption that $|A| \geq l_x$. Some remarks about the case where $|A| < l_x$ will be made in Appendix B.

We start with the simple sequences and consider a sequence x of some fixed length l_x and suppose at least one token of x repeats in x, i.e. there exists at least one token that occurs on at least two different positions in x. Call this token α and suppose its first two occurrences are on positions i and j with $i < j$. Then we have $e(i,j) = e(i,i) = 1 = e(j,j) = e(j,i)$ and $j < i$, i.e. $e(j,i)$ is a subdiagonal element from e with $2 \leq j \leq l_x$. Consider all n-paths in e with $1 \leq n \leq j$ which contain $e(i,j)$ as their last (and possibly first) element and of which all other elements are diagonal. Then the number of these paths equals $\sum_{i=0}\binom{j-1}{i} = 2^{j-1}$. Likewise, the number of paths of which the first element is $e(i,j)$ and all the other elements are diagonal equals $2^{l_x - i}$. Hence, a lower bound on the number of paths in $\boldsymbol{E}$ that contain $e(i,j)$, amounts to $2^{l_x-i+j-1} - 1$. This number is a lower bound since their might be other positive subdiagonal elements with which paths containing $e(i,j)$ could be constructed. Therefore, if we replace the token α, say on position j, by a token from A, say β, that is not occurring in x, and such a token exists since α repeats and $|A| \geq l_x$, we create a sequence y with $\|\boldsymbol{y}\|^2 < \|\boldsymbol{x}\|^2$ since β does not repeat in y, so $e_{y,y}(i,j) = 0$. So, $\|\boldsymbol{x}\|^2$ is minimal given l_x, precisely when no token in x is repeated and $e_{x,x}(i,j) = 1$ if and only if $i = j$. But then the total number of paths in $\boldsymbol{E}_{x,x}$ amounts to $\sum_{i=1}\binom{l_x}{i} = 2^{l_x} - 1 = \|\boldsymbol{x}\|^2$.

Obviously, for simple sequences with some fixed l_x, the maximum of $\|\boldsymbol{x}\|^2$ will be attained when all elements of $\boldsymbol{E}$ are positive, i.e. when x contains just one single token from A that repeats l_x times. But then we have $\|\boldsymbol{x}\|^2 = \sum_{i=1}\binom{l_x}{i}^2 = \binom{2l_x}{l_x} - 1$, since of the first $|A|$ coordinates of $\boldsymbol{x}$, only one will be nonzero and will have a value of $\binom{l_x}{1}$; of the next $|A|^2$ coordinates of $\boldsymbol{x}$, again only one will be nonzero with value $\binom{l_x}{2}$; etc. This fact can also be directly derived from the structure of $\boldsymbol{E}$: since every element of $\boldsymbol{E}$ is positive, every $(i \times i)$-submatrix of $\boldsymbol{E}$ contains exactly one i-path; its diagonal, and there exist $\binom{l_x}{i}^2$ of such submatrices. We summarize the above by stating that, for a simple sequence with $l_x \leq |A|$, we have

$$2^{l_x} - 1 \leq \|\boldsymbol{x}\|^2 \leq \binom{2l_x}{l_x} - 1. \qquad (6)$$

Next we try to determine similar boundaries in case x is a time-coupled sequence. We suppose x to be of length l_x with a total time trajectory of length $L_x = \sum_i t_x(i)$. It is quite obvious now, that $\|\boldsymbol{x}\|^2$ will be minimal when x consists of l_x different tokens with $t_x(i) = L_x/l_x$ for all $1 \leq i \leq l_x$. We take $L_x/l_x = 1$ and determine $\|\boldsymbol{x}\|^2 = \sum_{i=1}^{l_x}\binom{l_x}{i}i^2 = 2^{l_x-2}l_x(l_x+1)$. The latter equality arises since the summands in the middle expression are hypergeometric terms so we determine the closed expression through evaluating the sum by Gosper's algorithm (e.g. Petkovšek, Wilf and Zeilberger 1996, chap 5).

The squared length $\|\boldsymbol{x}\|^2$ will increase when the number of different states decreases to the minimum thereof which equals 2. Then $\|\boldsymbol{x}\|^2$ will further increase when an ever increasing part of L_x is occupied by either of the two extreme states, say the first one. Hence, a sequence with a very big value of $\|\boldsymbol{x}\|^2$ will be of the form $[\alpha_{L_x-w}, \beta_{v_2}, \alpha_{v_3}, \beta_{v_4}, \ldots]$ with $w = \sum_{i=2}^{l_x} v_i$ being extremely small. Therefore, a good approximation of the maximum of $\|\boldsymbol{x}\|^2$ given l_x and L_x will be $\|\boldsymbol{x}^*\|^2$ of the imaginary sequence $x^* = [\alpha_{L_x}, \beta_0, \alpha_0, \beta_0, \ldots] \neq$

$[\alpha_{L_x}]$. The form of $\boldsymbol{E}_{x^*,x^*}$ will be (omitting zero's)

$$\boldsymbol{E}_{x^*,x^*} = \begin{array}{c c} & \begin{array}{cccccc} \alpha & \beta & \alpha & \beta & \alpha & \dots \end{array} \\ \begin{array}{c} \alpha \\ \beta \\ \alpha \\ \beta \\ \alpha \\ \vdots \end{array} & \left(\begin{array}{cccccc} 1 & & 1 & & 1 & \dots \\ & 1 & & 1 & & \dots \\ 1 & & 1 & & 1 & \dots \\ & 1 & & 1 & & \dots \\ 1 & & 1 & & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array} \right) \end{array} = \boldsymbol{E}.$$

All n-paths in $\boldsymbol{E}$ that contain $e(1,1)$ will contribute L_x^2 to $\|\boldsymbol{x}^*\|^2$ and all the n-paths not containing $e(1,1)$ will not contribute to $\|\boldsymbol{x}^*\|^2$. Therefore, we need the sum of the number of n-paths in $\boldsymbol{E}$ that contain $e(1,1)$. That sum is the sum of all n-paths $(k \geq 1)$ in the $(l_x - 1) \times (l_x - 1)$ submatrix of $\boldsymbol{E}$ that emerges when the first row and the first column of $\boldsymbol{E}$ are removed. A simple, closed expression for the number of n-paths in this submatrix could not be found. But a simple expression for the sum of the number of these paths does exist and is given by (Sloane and Plouffe 1995, sequence A025565)

$$\sum_{n=0}^{\lfloor l_x/2 \rfloor} \binom{l_x - 2}{n} \binom{l_x - n}{n+1}.$$

Therefore, in the case of time-coupled sequences, $\|\boldsymbol{x}\|^2$ is bounded by the expression

$$L_x^2 2^{l_x - 2}(l_x + 1)/l_x \leq \|\boldsymbol{x}\|^2 \leq L_x^2 \sum_{n=0}^{\lfloor l_x/2 \rfloor} \binom{l_x - 2}{n} \binom{l_x - n}{n+1}. \tag{7}$$

5. Complexity and Homogeneity

If our target data matrix would consist of ordinary numerical measurements, we would probably start a description of that matrix by mentioning several means and variances. With token sequences, means and variances cannot be so easily defined. But consider the small example sequences presented below, together with the squared lengths of their representing vectors:

$$\begin{array}{ll} v = [\alpha, \alpha, \alpha], & \|\boldsymbol{v}\|^2 = 19, \\ w = [\alpha, \alpha, \beta], & \|\boldsymbol{w}\|^2 = 11, \\ x = [\alpha, \beta, \alpha], & \|\boldsymbol{x}\|^2 = 9, \\ y = [\alpha, \beta, \gamma], & \|\boldsymbol{y}\|^2 = 7. \end{array}$$

We are probably inclined to consider v as the most simple sequence and y as the least simple one, i.e. the most complex one; going down this list of sequences, one would need an increasingly complex statement in ordinary language to fully describe the sequences. From the squared lengths of the representing vectors, it seems that these lengths well quantify these complexities. However, these lengths are not very useful in comparing sequences of different lengths l. We write $x_{\max}$ for the maximum value of $\|\boldsymbol{x}\|^2$ and $x_{\min}$ for the minimum value of $\|\boldsymbol{x}\|^2$, i.e. $x_{\max} \equiv \binom{2l_x}{l_x} - 1$ and $x_{\min} \equiv 2^{l_x} - 1$, and our first attempt to quantify complexity is

$$0 \leq c(x) \equiv \frac{x_{\max} - \|\boldsymbol{x}\|^2}{x_{\max} - x_{\min}} \leq 1. \tag{8}$$

The numerator of $c(x)$ measures the distance between $\|\boldsymbol{x}\|^2$ and its maximum, given l_x, and the denominator relates this distance to the possible range of distances, which itself only depends upon sequence length. However, $c(x)$ as defined in (8) has the very unfortunate property that its resolution is quite limited. For consider a sequence x, comprised of a token, say α, that repeats j times and k tokens different from α. Furthermore, suppose all of these k tokens are different from each other. Under these assumptions, we have $\|\boldsymbol{x}\|^2 = 2^k \binom{2j}{j} - 1$, from which substituting the appropriate boundaries in (6) and using Stirling's approximation (e.g. Knuth 1997) $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, one derives $\lim_{j\to\infty} c(x) = 1 - 2^{-k}$. In applications, this kind of asymptotic behavior implies that in most circumstances, $c(x)$ will be very close to 1. The reason for this behavior is that, with increasing l_x, the upper bound $\binom{2l_x}{l_x} - 1$ increases very much faster than the lower bound $2^{l_x} - 1$: the central binomial increases so rapidly that the relative difference of either subtracting $\|\boldsymbol{x}\|^2$ or $x_{\min}$ from it, does not make a big difference.

Our second attempt to quantify complexity focuses on the ratio $x_{\min}/\|\boldsymbol{x}\|^2$. Evidently,

$$\frac{x_{\min}}{x_{\max}} \leq \frac{x_{\min}}{\|\boldsymbol{x}\|^2} \leq 1 \tag{9}$$

and the lower bound of (9) depends upon $x_{\max}$. This lower bound will rapidly tend to zero as l_x increases, but not rapidly enough to ensure comparability of complexity numbers for shorter sequences of different sequence lengths. Therefore, we map the image of $x_{\min}\ /\|\boldsymbol{x}\|^2$ onto the closed interval $[0, 1]$ by the transformation

$$0 \leq C(x) \equiv \left(\frac{x_{\max} - \|\boldsymbol{x}\|^2}{x_{\max} - x_{\min}}\right) \cdot \left(\frac{x_{\min}}{\|\boldsymbol{x}\|^2}\right) \leq 1. \tag{10}$$

Now, $C(x)$ measures the ratio $x_{\min}/\|\boldsymbol{x}\|^2$, corrected for the relative position of $\|\boldsymbol{x}\|^2$ in the range $x_{\max}-x_{\min}$ and this normalised quantity doesn't suffer from the shortcomings of either (8) or (9). A few examples of the behavior of $C(x)$ as shown in Table 2 illustrate this.

$C(x)$ is akin to the variance σ^2 of a sequence of numbers: only $C(x)$ uses information on all the possible k-tuples whereas σ^2 uses only the squared distance between 2-tuples; tuples of higher order are used to describe higher moments of the number sequence. $C(x)$ will be close to zero if there are many repetitions of tokens just like σ^2 will be relatively small if there are many repetitions of numbers in the number sequence. On the other hand, σ^2 will remain to be greater than zero, even if there are only very few different numbers, just like $C(x)$ will not reach zero as long as there is only a tiny fraction of tokens that differs from the rest.

As discussed in the previous section, the boundaries of $\|\boldsymbol{x}\|^2$ for time coupled sequences are well established, at least when $l_x \leq |A|$. But we also demonstrated that $\|\boldsymbol{x}\|^2$ given l_x and L_x, decreases with an increase in the number of different states in x but increases with an increase in the variation of the times associated with the states. Therefore, the interpretation of a measure like $C(x)$ for time coupled sequences is far from clear.

Let X be a set of m sequences and $\boldsymbol{X} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x}_m)$ be the $m \times n$-matrix of representing (row-)vectors. Of course, one could say that the centroid $\boldsymbol{c}_X = (c_1, \ldots c_n)$ with $c_i = \frac{1}{m}\sum_j^n x_{i,j}$ is characterising the set $\boldsymbol{X}$. However, for most sets X, the centroid does not represent any constructable sequence. So, the best we can do, is to specify those vectors from $\boldsymbol{X}$ that have minimal distance to $\boldsymbol{c}$ and consider this set (since there could be more than one of such vectors) as characterising X. Since our algorithms do not provide us with $\boldsymbol{X}$ but with the $m \times m$-matrix $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m) = \boldsymbol{P}_{XX} = \boldsymbol{XX'}$ of inner products, we find the distance of some $\boldsymbol{x}_j \in \boldsymbol{X}$ to $\boldsymbol{c}$ as

$$\|\boldsymbol{x}_j - \boldsymbol{c}\|^2 = \frac{1}{m^2}\boldsymbol{iP}_{XX}\boldsymbol{i'} - \frac{2}{m}\boldsymbol{p}_j\boldsymbol{i'} + p_{j,j} \equiv d_{j,c}^2$$

where $\boldsymbol{p}_j$ denotes the j^{th} row from $\boldsymbol{P}_{XX}$, $p_{j,j} = \|\boldsymbol{x}_j\|^2$ and $\boldsymbol{i'} = (1, 1, \ldots, 1)$. Obviously, the average $H_X = m^{-1}\sum_i^m d_{i,c}$ is a good descriptor of the homogeneity of $\boldsymbol{X}$. Furthermore, if X and Y are two sets of sequences, then

$$d_{c_x,c_y}^2 = m_x^{-2}\boldsymbol{iP}_{XX}\boldsymbol{i'} + m_y^{-2}\boldsymbol{iP}_{YY}\boldsymbol{i'} - 2(m_x m_y)^{-1}\boldsymbol{iP}_{XY}\boldsymbol{i'}$$

measures the difference in location of the two sets. In applications, one could use one of the variants of the non-parametric Kolmogorov-Smirnov statistic to test for a difference between the distributions of the distances to $\boldsymbol{c}_X$ and $\boldsymbol{c}_Y$.

Table 2. Sequence complexity $C(x)$ (Eq. 9) for some example sequences. The middle column shows the normalisation factor for the ratio $x_{\min}/\|\mathbf{x}\|^2$.

x	$c(x)$	$C(x)$
$[\alpha,\alpha,\beta]$	0.667	0.424
$[\alpha,\beta,\alpha]$	0.833	0.648
$[\alpha,\beta,\gamma]$	1	1
$[\alpha,\beta,\gamma,\alpha,\beta,\gamma]$	0.958	0.610
$[\alpha,\beta,\gamma,\gamma,\beta,\alpha]$	0.951	0.571
$[\alpha,\alpha,\beta,\beta,\gamma,\gamma]$	0.823	0.241
$[\alpha,\beta,\gamma,\gamma,\delta,\epsilon]$	0.963	0.638
$[\alpha,\alpha,\alpha,\alpha,\alpha,\alpha,\alpha,\alpha,\alpha,\beta]$	0.476	0.005

6. Similarity of Sequences

The quest for useful representation of token sequences stems from the apparent need to find ”typical patterns” or ”characteristic sequences”, i.e. sequences that are, e.g. on the average, more similar to a set of sequences than any other sequence.

A substantial part of the criticisms (e.g. Dijkstra and Taris 1995; Wu 2000) raised to the use of OM methods was directed to the notion of sequence similarity that seemed to arise from these methods and not so much to the representation in a Hamming or Levenshtein metric as such. This is perfectly understandable, since all one can say about the relation between a similarity measure and a distance metric is that the one should be nonincreasing with the other. Therefore, given a metric representation, it is not immediate how to derive a similarity measure from it. So, Elzinga (2003) formulated a minimal set of rules that is independent of any representation, to which a similarity measure for simple sequences should adhere. These are:

1. Sequences that have no common tokens are maximally dissimilar.
2. Sequences that consist of exactly the same tokens in the same order are maximally similar.
3. Similarity increases with an increase in the number of common tokens.
4. The more common order there is amongst common tokens, the more similar the sequences are.

To these rules, one should add, as a refinement of rule 2,

> 2a. Sequences with identical tokens in the same order are maximally similar, if the ratio’s of the quantities associated to these tokens in the one sequence are identical to the corresponding ratios in the other sequence.

Rule 2a demands that a similarity measure is time scale invariant. With these rules in mind, a natural candidate for a similarity measure, both for simple and for time-coupled sequences, is

$$s_{x,y} = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|}, \tag{11}$$

i.e. the cosine of the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$ in the unit sphere. Like $C(x)$ is a direct analogue to variance, $s_{x,y}$ is a direct analogue to Pearson's correlation coefficient. It is not difficult to see that (11) indeed adheres to the rules 1-4 as stated above. First, if sequences x and y have no common tokens, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$. So, (11) satisfies rule 1. If sequences x and y are identical we obviously have $s_{x,y} = 1$, so (11) satisfies rule 2 and, because of (1), it also satisfies time scale invariance (rule 2a). Suppose u is the longest common subsequence of x and y. Furthermore, suppose that $\alpha \subset_1 x$ and $\beta \subset_1 y$ and that $[\alpha, \beta] \not\subset_2 u$. Now replace β by α, thus creating y'. Evidently, $\|\boldsymbol{y}'\| = \|\boldsymbol{y}\|$ but $\langle \boldsymbol{x}, \boldsymbol{y}' \rangle > \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ so $s_{x,y'} > s_{x,y}$, hence (11) satisfies rule 3. To see that (11) also adheres to rule 4, consider two sequences x and y, each consisting of the same, non-repeating tokens in different permutations. Furthermore, suppose that $[\alpha, \beta] \subset_2 x$ and $[\alpha, \beta] \not\subset_2 y$. Now interchange α and β in y, thus creating the sequence y'. Again, we have $\|\boldsymbol{y}'\| = \|\boldsymbol{y}\|$ but $\langle \boldsymbol{x}, \boldsymbol{y}' \rangle > \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ so $s_{x,y'} > s_{x,y}$.

It is interesting to investigate some of the numerical properties of $s_{x,y}$, and therewith the properties of the representations, by confronting it with various small example sequences. To start with, we restrict ourselves to simple sequences and show some figures in Table 3.

Indeed, $s_{x,y}$ behaves as expected. Note that the last sequence in Table 3 is a complete revert of the first one. We inspect the behaviour of $s_{x,y}$ in handling complete reverts in some more detail by constructing pairs of sequences x and x' with x consisting of $l_x > 1$ different tokens and x' being a complete revert of x. The results are shown in Table 4.

Note the strange, oscillating behaviour of the OM coefficient. The above results encourage determining some more general properties of $s_{x,y}$. Therefore, we consider pairs of sequences of the form $x = [u, v]$ and $y = [u, w]$ with $u = [\alpha, \beta, \ldots]$ of length $l_u = k$ with all different tokens; v and w have lengths $l_v = j = l_w$ and have no tokens in common with u. To start with, we furthermore assume that v and w consist of j different tokens and do not have common tokens. Clearly, then

$$s_{x,y} = \frac{\sum_{i=1} \binom{k}{i}}{\sum_{i=1} \binom{k+j}{i}} = \frac{2^k - 1}{2^{k+j} - 1}, \tag{12}$$

from which it is immediate that, assuming an unrestricted alphabet,

Table 3. The under diagonal part shows values of $s_{x,y}$ as defined in Eq. 11; the upper diagonal part shows the values of the similarity derived from a unit-cost OM representation. The sequences themselves are shown in the first column.

$[\alpha, \beta, \gamma, \delta]$	1	0.5	0.25	.0	0.25	.0
$[\alpha, \beta, \delta, \gamma]$	0.733	1	0.5	0.5	0.5	.0
$[\beta, \alpha, \delta, \gamma]$	0.533	0.733	1	0.5	0.5	.0
$[\beta, \delta, \alpha, \gamma]$	0.467	0.6	0.733	1	0.5	.0
$[\delta, \beta, \alpha, \gamma]$	0.4	0.467	0.6	0.733	1	0.5
$[\delta, \gamma, \beta, \alpha]$	0.267	0.333	0.4	0.467	0.6	1

Table 4. Similarity between pairs of sequences consisting of l_x different tokens in which the one sequence is a complete revert of the other. OM indicates the values of the similarity index as derived from a unit-cost OM representation and $s_{x,y}$ is as defined in Eq. 11

l_x	2	3	4	5	6	7	8	9	10
OM	0	0.333	0	0.2	0	0.143	0	0.111	0
$s_{x,y}$	0.667	0.429	0.267	0.161	0.095	0.055	0.031	0.018	0.001

$\lim_{k\to\infty} s_{x,y} = 2^{-j}$ (note that the way tokens from u and v are mixed, is not relevant). We determined several of these limits for various compositions of the subsequences v and w and show the results in Table 5.

The reader correctly guesses that the same table results from any composition of the common subsequence u. Indeed, the above table implies that the pair of sequences $x = [\alpha, \alpha, \ldots, \alpha, \gamma]$ and $y = [\alpha, \alpha, \ldots, \alpha, \delta]$ is less similar than the pair $y = [\alpha, \beta, \gamma, \delta]$ and $z = [\beta, \alpha, \gamma, \delta]$ although perceptually, they are almost perfectly similar. But perceptual similarity is perhaps not very relevant in this context and it does not appear in the rules stated above. The fact that the similarity between $[\alpha, \ldots, \alpha, \beta]$ and $[\alpha, \ldots, \alpha, \gamma]$ tends to 0.5 justly reflects the fact that these sequences differ in an important aspect: the one ends with β and the other with γ. If this is felt to be not so important after all, a redesign of the encoding itself is indicated.

We now turn our attention to the behaviour of $s_{x,y}$ in case of time-coupled sequences, i.e. the representation defined by (1). Table 6 shows similarities between the same sequences that were used in Table 3, the difference now being that all states occupy the same, constant amount of time.

Clearly, the values differ from those in Table 3 but the order relations between the figures are exactly the same as those of Table 3. In Table 7, the tokens of each sequence are the same and in the same order but now the times

Table 5. $\lim_{k\to\infty} s_{x,y}$ for sequences $x = [u, v]$ and $y = [u, w]$ where k is the length of the common subsequence u and j is the length of the noncommon subsequences v and w. The compositions of v and w are shown in the first row and column of the Table. $[\alpha, \alpha, ...]$ or $[\beta, \beta, ...]$ mean that v and/or w consist of one single repeating token, not occurring in u, and $[\zeta, \eta, ...]$ or $[\beta, \gamma, ...]$ mean that v and/or w consist of different tokens, not occurring in u. $[\,]$ denotes that v and/or w are empty.

v/w	$[\beta, \beta, ..]$	$[\beta, \gamma, ..]$	$[\,]$
$[\alpha, \alpha, \ldots]$	$\binom{2j}{j}^{-1}$		
$[\zeta, \eta, \ldots]$	$\sqrt{2^{-j}\binom{2j}{j}^{-1}}$	2^{-j}	
$[\,]$	$\sqrt{\binom{2j}{j}^{-1}}$	$\sqrt{2^{-j}}$	1

Table 6. Sequence similarity and the permutation of tokens

$[\alpha_1, \beta_1, \gamma_1, \delta_1]$	1					
$[\alpha_1, \beta_1, \delta_1, \gamma_1]$	0.525	1				
$[\beta_1, \alpha_1, \delta_1, \gamma_1]$	0.25	0.525	1			
$[\beta_1, \delta_1, \alpha_1, \gamma_1]$	0.2	0.363	0.525	1		
$[\delta_1, \beta_1, \alpha_1, \gamma_1]$	0.15	0.2	0.363	0.525	1	
$[\delta_1, \gamma_1, \beta_1, \alpha_1]$	0.05	0.1	0.15	0.2	0.363	1

Table 7. Sequence similarity and the permutation of durations

$[\alpha_{27}, \beta_9, \gamma_3, \delta_1]$	1					
$[\alpha_{27}, \beta_9, \gamma_1, \delta_3]$	0.733	1				
$[\alpha_9, \beta_{27}, \gamma_3, \delta_1]$	0.533	0.733	1			
$[\alpha_9, \beta_1, \gamma_{27}, \delta_3]$	0.467	0.6	0.733	1		
$[\alpha_1, \beta_9, \gamma_{27}, \delta_3]$	0.4	0.467	0.6	0.733	1	
$[\alpha_1, \beta_3, \gamma_9, \delta_{27}]$	0.267	0.333	0.4	0.467	0.6	1

occupied by these states have been shuffled like the tokens were shuffled in the previous table.

As expected, the effect of variation in the precedences (Table 6) is much bigger than the effect of varying the distribution of times occupied with constant precedences (Table 7). Again, the order relations between the figures are exactly those of Tables 3 and 6. The last table demonstrates the effect of shuffling precedences with varying times occupied by the different states.

The similarity between time-coupled sequences appears to exhibit the same kinds of asymptotic behaviour as was discussed above for the simple sequences and summarised in Table 5. We demonstrate this by considering pairs of sequences $x = [u, v]$ and $y = [u, w]$ and computing $s_{x,y}$ for ever increasing lengths of the common subsequence u. We assume $l_x = L = l_y$ and u to be a sequence consisting of $L - j$ different states, each of which occupying just 1 single unit of time. Now, for example, suppose that v and w each contain just 1 single state, different from each other and both not occuring in u, that occupies j units of time.

Then $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1} i^2 \binom{L-j}{i}$ and $\|\boldsymbol{x}\|^2 = \|\boldsymbol{y}\|^2$ with

$$\|\boldsymbol{x}\|^2 = \sum_{i=0}^{L-j} (j+i)^2 \binom{L-j}{i} + \sum_{i=1}^{L-j} i^2 \binom{L-j}{i}.$$

Substituting these expressions into (11) and using Gosper's method (Petkovšek et al. 1996) again, this yields

$$s_{x,y} = 2^{-1} \frac{L(L-j+1) + j(j-1)}{L(L+1) + j(j-1)},$$

hence $\lim_{L\to\infty} s_{x,y} = 2^{-1}$. Again, similar expressions can be derived for various combinations of compositions of v and w and their limiting values are shown in Table 9.

The results of this section and the previous one seem to be sufficiently encouraging to confront the represention with some real world data.

7. Application 1: Encoded Structured Interviews

In this section, we analyse a typical social science example of simple token sequences: encoded structured interviews. A structured interview is an interview in which the interviewer asks preformulated questions in a fixed order or according to some fixed routing and in which the respondent is supposed to answer these questions by choosing one of a number of prescribed response alternatives. All verbal utterances in such an interview can be encoded. The data we discuss here comprise of a subset of data collected and amply described by Draisma (2000). Essential for our present discussion is, that from a number of the questions posed in these interviews, the correct answers were known to the researchers. So answers given by subjects, who were unaware of the researcher's knowledge of the correctness of their answers, could be compared to the correct answers. These interviews were transcripted and encoded according to a multivariable encoding method as described by Dijkstra (1999). From these encoded interviews, 923 Question/Answer-sequences were picked that were complete and of which the correct answers were known. We stripped and simplified the encoded utterances to a very basic, two-variable encoding scheme that describes 'who does what'. Furthermore, we removed introductory statements and compressed identical adjacent codes into one single code. Encoded in this way, the sequences consist of two-character symbols, the first character indicating the 'who' - I(nterviewer) or R(espondent) - and the second character indicating the 'what' - the specific kind of utterance. The utterances were encoded according to a scheme shown in Table 10.

So, the complete alphabet consists of 10 tokens; each of the characters from Table 10, preceded by either I or R. A typical Q/A-sequence then might look like

IQ RC RA IP RA IC IP .

In this example the interviewer poses a question, the respondent comments upon it, then produces an answer of which the interviewer acknowledges perception, again the respondent gives an answer on which the interviewer makes a comment and a closing acknowledgement. The 923 resulting sequences vary in length from 2 to 19 tokens with an average of some 4.4 tokens per sequence and a standard deviation of 2.7 tokens. In Table 11, the 'typical' Q/A-sequence is shown, along with the sequence whose representing vector has the minimal distance to the centroid of all the representing vectors.

Interestingly, the first sequence mentioned in Table 11 is known as the 'paradigmatic sequence' in any structured interview: it represents the ideal sequence where the interviewer clearly states the question as intended and correctly presents all of the possible response alternatives, in which the respondent gives an adequate answer that is acknowledged by the interviewer (e.g. Maynard and Schaeffer, 2002). In fact, this is certainly not the most frequently observed sequence; in practice, all sorts of variations on and violations of this paradigm do occur. Apparently, this paradigmatic sequence is 'recognised' by the representation as the backbone structure of all the sequences. Quite common is the structure that is generated as the sequence closest to the centroid; a variation in which the respondent repeats his answer with the same or different wording, confirming his previous statement. In Table 12 some more details of the representation are shown.

The fact that there is a common backbone structure, the paradigmatic sequence, is reflected in the fact that $\overline{s}_{i,j}$ is reasonably high in view of the fact that there is quite a large standard deviation of the interpoint distances. Note also that the sequence with $d_{\min}$ is roughly 3 times closer to the centroid than the average distance $\overline{d}_{i,j}$. The questions were Yes/No-questions with the additional possibility of choosing 'Don't know' for an answer. In Table 13, we show the

Table 8. Sequence similarity: permutations of tokens and duration.

$[\alpha_{27}, \beta_9, \gamma_3, \delta_1]$	1					
$[\alpha_{27}, \beta_9, \delta_1, \gamma_3]$	0.716	1				
$[\beta_9, \alpha_{27}, \delta_1, \gamma_3]$	0.283	0.401	1			
$[\beta_9, \delta_1, \alpha_{27}, \gamma_3]$	0.203	0.222	0.498	1		
$[\delta_1, \beta_9, \alpha_{27}, \gamma_3]$	0.193	0.194	0.468	0.666	1	
$[\delta_1, \gamma_3, \beta_9, \alpha_{27}]$	0.085	0.086	0.22	0.301	0.453	1

Table 9. $\lim_{L\to\infty} s_{x,y}$ for sequences $x = [u, v]$ and $y = [u, w]$ where L denotes the total time trajectory of both x and y. x and y are supposed to consist of $L - j$ common tokens, each occupying 1 unit of time. u and v consist of common tokens, occupying j units of time. The compositions of v and w are shown in the first row and column of the Table. $[\alpha_j]$ and $[\beta_j]$ indicate that u and v both consist of one token, different for u and v, that occupy j units of time. $[\gamma_1, \delta_1, \ldots]$ and $[\zeta_1, \eta_1, \ldots]$ indicate that u and v both consist of j different tokens, each occupying 1 unit of time. $[\,]$ denotes that v and/or w are empty.

v/w	$[\beta_j]$	$[\gamma_1, \delta_1, \ldots]$	[ ]
$[\alpha_j]$	2^{-1}		
$[\zeta_1, \eta_1, \ldots]$	$2^{-(j+1)/2}$	2^{-j}	
$[\,]$	$\sqrt{2^{-1}}$	$\sqrt{2^{-j}}$	1

Table 10. Coding scheme of types of verbal utterances, either made by the interviewer or the respondent

code	meaning
Q	asks question
A	answers question
P	acknowledges perception
C	comments
R	requests

Table 11. Characteristic sequences for 923 encoded Q/A-sequences. The typical sequence is the sequence with the highest average similarity ($\bar{s}_{\max}$) with all the other sequences. $d_{\min}$ is the smallest Euclidean distance to the centroid of the 923 vectors.

IQ RA IP	$\bar{s}_{\max} = 0.717$
IQ RA IP RA IP	$d_{\min} = 10.37$

Table 12. $\overline{d}_{i,j}$ is the average Euclidean distance between representing vectors, $\overline{s}_{i,j}$ denotes the average similarity, $\overline{C}$ is the average complexity[1] of the sequences and $\bar{l}$ is the average sequence length. sd always denotes the standard deviation of the quantity to the left of it.

n	$\overline{d}_{i,j}$	sd_d	$\overline{s}_{i,j}$	sd_s	$\overline{C}$	sd_C	$\bar{l}$	sd_l
923	52.9	282.4	0.562	0.287	0.852	0.234	4.38	2.69

Table 13. Representation characteristics for differently answered questions. Symbols are explained in the legenda to Table 12.

	correct	incorrect	don't know
n	714	148	61
$\overline{d}_{i,j}$	36.09	51.73	133.06
sd_d	214.45	96.82	513.65
$\overline{s}_{i,j}$	0.603	.475	0.386
sd_s	0.283	0.284	0.235
$\overline{C}$	0.877	0.78	0.975
sd_C	0.215	0.269	0.034
$\bar{l}$	4.08	5.09	6.13
sd_l	2.38	3.45	3.06

Table 14. Characteristic sequences for differently answered questions. Representation details in Table 13.

correct	$\overline{s}_{\max} = .748$	IQ RA IP
	$d_{\min} = 8.41$	IQ RA IP RA
incorrect	$\overline{s}_{\max} = .638$	IQ RA IP
	$d_{\min} = 32.82$	IQ RP RA IP RA IP
don't know	$\overline{s}_{\max} = .524$	IQ RA IP
	$d_{\min} = 13.22$	IQ RP RA IP RA IP

same kind of results as in Table 12, but now for correct, incorrect and 'Don't know'-answers separately. In Table 14, the characterising sequences for these groups of differently answered questions are shown.

From Tables 13 and 14, it is obvious that the three groups of differently answered questions are quite differently represented. The sequences leading to correct answers seem to be shorter and relatively more complex than those which result in an incorrect answer. Furthermore, these sequences seem to be most similar to the paradigmatic sequence and have, on the average, the highest intersequence similarity. A more detailed analysis of these results and sequences is beyond the scope of the present paper. What is important, is that the combinatorial representation of these simple (and simplified) sequences leads to a useful and meaningful description of the data.

8. Application 2: Life Histories of Young Adults

In trying to validate their proposals to quantify sequence similarity, Dijkstra and Taris (1995) and Elzinga (2003) amply discussed the dataset that we will use again in this paper to demonstrate the applicability of the representations described above. These data consist of 494 encoded life histories of adults (244 females and 250 males), interviewed at the age of 26. Encoding was done on 3 variables: living situation, education and employment, according to the scheme presented in Table 15.

As a result of this encoding, each subjects life history consists of a sequence of 3-character states, ranging in length from 2 to 17 states. Since, at the time, neither Dijkstra et al. nor we knew how to handle associated quantities, it was never mentioned that the duration of all the states, except for the last one, was also known. Thus an individuals life history was fully encoded as, for example,

HOO/60, HFO/140, HPP/20, HOF/4, SOF/16, POP/-1

The last event of each sequence was always assigned a duration of -1, meaning that the duration of the last state was unknown since it was the state the subjects were in at the time of data collection. For each subject, gender and a score on a 10-point scale[2] of socio-economic status (SES) was known. These data, without the associated durations, were used by Dijkstra et al. to demonstrate that there is/was a ‘typical’, ‘traditional’ life history sequence, different for males and females:

(HOO) HFO HOO HOF MOF (MOO).

The first event is put between parenthesis because the first episode (lasting invariably for 60 months for each and every subject in the sample) wasn’t mentioned by Dijkstra et al. The last event is put between parenthesis, because this last event was thought to be typical for females and not for men. The above sequences were inferred by Dijkstra et al. from ‘typical’ sequences for males and females that were found by looking at that sequence with the highest average similarity (according to their index γ) with all other sequences in the data. Since the duration of the last state of each sequence is unknown, we decided to

Table 15. Coding scheme of life histories on three variables, according to Dijkstra and Taris (1995). ‘part time’ means that employment or education was less than 4 days per week.

Variable	Category	Code
living	with parents	H
	alone	S
	with partner	P
	married	M
	other	O
education	full time	F
	part time	P
	none	O
employment	full time	F
	part time	P
	none	O

Table 16. Typical patterns of time coupled life histories. The sequences indicated with $\overline{s}_{\max}$ denote the sequences with the maximum average similarity to all other sequences; the sequences indicated with $d_{\min}$ are those sequences of which the distance to the centroid is smallest.

total	$\overline{s}_{\max} = .233$	HOO/60 HFO/214 HOO/2 HOF/24
	$d_{\min} = 8.9$	HOO/60 HFO/194 HOO/18 HOF/28
females	$\overline{s}_{\max} = .205$	HOO/60 HFO/178 HOO/6 HOF/44 MOF/12
	$d_{\min} = 11.8$	HOO/60 HFO/154 HOO/3 HOF/83
males	$\overline{s}_{\max} = .255$	HOO/60 HFO/214 HOO/2 HOF/24
	$d_{\min} = 10.7$	HOO/60 HFO/194 HOO/18 HOF/28

Table 17. Characteristics of representations of time coupled life histories. Distances haven been rescaled (multiplier = 0.006689) such that $\overline{d}_{i,j} = 100$ for the total group. See the legenda to Table 12.

	total	females	males
n	494	244	250
$\overline{d}_{i,j}$	100	103.1	96.9
sd_d	174.3	207.5	134.1
$\overline{s}_{i,j}$	0.0838	.0765	0.0982
sd_s	0.1199	0.1164	0.1310
$\max \overline{s}$	0.2332	0.2051	0.2547

censor[3] all the sequences at 300 months. In Tables 16 and 17, the main results of the analyses are shown.

The general picture from Table 16 corroborates previous findings with the same data without the state durations as reported by Dijkstra et al. (1995) and Elzinga (2003); only the state MOF does not occur, probably because of the inevitable censoring we had to apply. Again, the typical female sequence suggests that women tend to marry at an earlier age than men. Redoing the analysis with the same data recoded such that living together with a partner is equivalent to living married, also produces the state MOF in the typical male sequence. This indicates that young men do start experimenting with partnerships but are less prone to marriage than women. ANOVA (full factorial with gender and SES as fixed effects) with the similarity to the typical female sequence as the dependant variable does not reveal a significant effect of gender but does show a strong and significant effect of SES: the similarity to the typical female pattern diminishes with increasing SES. No interaction between gender and SES was found. The lack of such a gender effect is remarkable since it was reported by Dijkstra et al. (1995) and by Elzinga (2003) when analysing the simple life histories. Of course, a great deal of the similarities between the sequences is due to the fact that for all subjects, there is quite a long period of compulsory education: roughly 180 months in a total time span of only 300 months. So, within the first 180 months of these life histories, many differences should not and do not occur. Therefore, we removed the first 180 months from our data. The effects of this removal are shown in Tables 18 and 19.

As expected, the average similarity and the maximum average similarity do decrease. Remarkably, and as hoped for, the structure of the figures appearing in Tables 18 and 19 is almost identical to those of Tables 16 and 17. The same ANOVA now does reveal, apart from the effect of SES, a signicant effect of gender when the similarity with the female pattern of Table 19 is used as the dependent variable. Hence, our representation does generate a difference between male and female life histories, provided the episode of forced equality is removed from the data.

9. Conclusions

Neither of the applications discussed leads to any new or spectacular sociological insights into life histories of young adults or the turn-taking during structured interviews. On the contrary; nothing came out that was new, unexpected or not found in many other studies. This demonstrates that the representations constructed, both for the simple and for the time-coupled sequences, lead to a useful and meaningful description of the data: the sequences taken as the unit of description. And that was exactly what we set out for: a metric rep-

Table 18. Typical sequences after removing the first 180 months from the data.

total	$\overline{s}_{\max} = .175$	HFO/94 HOO/2 HOF/24
	$d_{\min} = 9.3$	HFO/74 HOO/18 HOF/28
females	$\overline{s}_{\max} = .149$	HFO/58 HOO/6 HOF/44 MOF/12
	$d_{\min} = 12.0$	HFO/41 HOF/79
males	$\overline{s}_{\max} = .203$	HFO/94 HOO/2 HOF/24
	$d_{\min} = 10.7$	HFO/74 HOO/18 HOF/28

Table 19. Characteristics of representations of time coupled life history sequences. Distances have been rescaled (multiplier=0.03628) such that $\overline{d}_{i,j} = 100$ for the total group. See the legenda to Table 12.

	total	females	males
n	494	244	250
$\overline{d}_{i,j}$	100	103.4	96.6
sd_d	166.2	196.5	129.9
$\overline{s}_{i,j}$	0.0535	.0460	0.0681
sd_s	0.1004	0.0967	0.1136

resentation of sequence data, free of any assumption about the nature, the origin or the generator of the sequences and the resulting Euclidean distances or similarities are directly amenable to the application of any of the well known classification tools. The theoretical part of the paper showed how to construct and to compute such representations and it is unequivocal about the arbitrariness of some of the choices made. But this arbitrariness does not differ from the arbitrariness of the standard descriptive tools that we use to describe numerical data: means, standard deviations, correlation coefficients and the like. By investigating the boundary properties of the representation, we have tried to show what the limitations of the representations are. For example, the quantification of similarity does not completely coincide with perceptual similarity. On the other hand, one may wonder what relevance perceptual similarity has, when studying life or employment histories, encoded interviews or stock investment patterns.

The next challenge is to come up with reasonable distribution theories that allow for proper statistical hypothesis testing.

A Algorithms

In Section 3, we explained that, both for simple and for time-coupled sequences, the basis of the algorithms is a dynamic algorithm that uses all n-

paths in an $(l_x \times l_y)$-matrix $\boldsymbol{E} = \{e(i,j)\}$ with $e(i,j) = 1$, precisely when the i^{th} token of x is identical to the j^{th} token from y, and $e(i,j) = 0$ otherwise. For simple sequences, one algorithm, SIMPLE, just adds the number of n-paths in such a matrix, for all $n \in \{1, \ldots, \min\{l_x, l_y\}\}$; the other algorithm, GOBBLE, uses the recursive equation (5) and is slightly more efficient than SIMPLE when e contains many 1's, i.e. when the sequences contain many repetitions of the same tokens. It is not difficult to design a variant of GOBBLE that handles associated quantities; we will not discuss this because of lack of space. For time-coupled sequences, the algorithm should assign a weight to each path, the weight being the product of the functions $g_{x,n}$ and $g_{y,n}$ as defined in (1). Here we discuss the variants of the basic dynamic algorithm since substantial optimisation is possible and necessary to ensure acceptable performance.

A1 SIMPLE

We slightly change our notation and write $\boldsymbol{E}_1$ instead of $\boldsymbol{E}$. Furthermore, we write $a_k(i,j)$ for the number of k-paths $\wp(i,j)$, i.e. paths of length k with first element $e(i,j)$. Then $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_k \sum_{i,j} a_k(i,j)$. Obviously, the sum of the 1-paths $a_1(i,j)$ equals $\sum_{i,j} e_1(i,j)$. The number of 2-paths $a_2(i,j)$ equals, for all i and j, $\sum_{n>i,m>j} e_1(i,j)$ and we define $\boldsymbol{E}_2 = \{e_2(i,j)\}$ with $e_2(i,j) = a_2(i,j)$. Once $\boldsymbol{E}_2$ has been constructed, it is immediate that $a_3(i,j) = \sum_{n>i,m>j} e_2(i,j) = e_3(i,j)$ and we construct $\boldsymbol{E}_3 = \{e_3(i,j)\}$, etc. Hence we compute $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{k=1} \sum_{i,j} e_k(i,j) \tag{13}$$

with

$$e_k(i,j) = \sum_{n>i} \sum_{m>j} e_{k-1}(n,m) \tag{14}$$

and $e_1(i,j)$ as defined above. This algorithm was already described in Elzinga (2003) but that description doesn't use the concept of a k-path and is far less elegant. Although the algorithmic complexity of the algorithm as embodied in (13) is already $O\left(n^3\right)$ where $n = \min\{l_x, l_y\}$, there are two highly efficient optimisations of the algorithm. The first one arises from the fact that, for all i, $e_k(m,i) = 0$ for all $m \geq l_x - k + 1$ and $e_k(i,m) = 0$ for all $m \geq l_y - k + 1$ since the corresponding k-paths do not exist. This implies that the upper boundaries of the summations in (14) decrease by 1 with every next k. Implementing this into (14), saves as much as $\left(4l^3 - 3l^2 - 1\right)/6$ summands in the expansion of (13) in case $\boldsymbol{E}$ is of dimensions $l \times l$. The second optimisation uses the fact that the calculation of $e_k(i,j)$ implies calculation of all $e_k(n,m)$ with $n > i$

and $m > j$. Hence, an implementation of (13) should compute (13) by adding the elements of e_{k-1} 'from below' and 'from the right' instead of taking (14) literally and doing the same computations over and over.

A2 GOBBLE

This algorithm computes $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ on the basis of the recursive equation (5): $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum i, j a_{i,j}$ with $a_{i,j} = 1 + \sum_{q>i,r>j} a_{q,r}$. However, a naive implementation of this algorithm will do the same calculations over and over again and therefore be very inefficient: once $a_{i,j}$ has been computed, all $a_{q,r}$ with $q > i$ and $r > j$ also must have been computed. This is easily avoided by providing the algorithm with a memory, in the form of a $(l_x \times l_y)$-matrix $\boldsymbol{M} = \{m_{i,j}\}$, that stores previously computed $a_{q,r}$ in $m_{q,r}$ for ever bigger values of q and r and adds $m_{i,j}$ to $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ instead of $a_{i,j}$ whenever $m_{i,j} > 0$.

A3 TIMEPATH

This algorithm computes $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ in case of time coupled sequences represented according to (2). Let $\boldsymbol{E}$ be equal to the matrix $\boldsymbol{E}_1$ from SIMPLE and, initially, S:=0. Then the algorithm finds, for each positive $e(i, j) \in \boldsymbol{E}$, all k-paths $\wp_k(i, j)$ with $k \geq 1$. Let $\wp_k(i, j)$ be a particular k-path in $\boldsymbol{E}$ and define

$$s_k = \sum_{e(n,m) \in \wp_k(i,j)} t_x(n) \cdot t_y(m). \tag{15}$$

Suppose $e(u, v)$ is the last element of $\wp_k(i, j)$. The algorithm then searches for the first positive element $e(u + a, v + b)$ in the submatrix $\{e(u + a, v + b)\}_{a \geq 1, b \geq 1}$ and elongates $\wp_k(i, j)$ to a $wp_{k+1}(i, j)$ by appending it with $e(u + a, b + v)$. The algorithm then computes $S := S + s_{k+1}$ with $s_{k+1} := s_k + t_x(u + a) \cdot t_y(v + b)$.

If such an element $e(u+a, v+b)$ does not exist, the k-path is shortened to a path $\wp_{k-1}(i, j)$ by removing $e(u, v)$ and it computes $s_{k-1} := s_k - t_x(u) \cdot t_y(v)$ if $k - 1 > 0$. The algorithm will then try to elongate $\wp_{k-1}(i, j)$ by searching the submatrix $\{e(u + a, v + b)\}_{a \geq 0, b \geq 1}$ for a next positive element. If $k - 1 = 0$, a next positive element is looked for in $\{e(i + a, j + b)\}_{a \geq 0, b \geq 1}$ and starts with a new 1-path, computes s_1, etc. If no more positive elements in $\boldsymbol{E}$ can be found, the algorithm is done and $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = S$.

B $\min \left\{ \|\boldsymbol{x}\|^2 \right\}$ and $c(x)$ with finite alphabets

If x has even length $l_x > |A|$, then a matrix $\boldsymbol{E}_{x,x} = \{e_x(i, j)\}$ of $\boldsymbol{x}$ such that $\|\boldsymbol{x}\|^2$ is minimal given l_x, is generated by a simple algorithm, which is given in pseudo-code follows:

```
**Input**: $m = |A| > 1$, $n = l_x > m$
**Output**: $(n \times n)$ permutation matrix $\boldsymbol{E}_n = \{e_n(i,j)\}$
$e_m(i,j) := 0, \qquad i \neq j,\ \ \forall\ i,j \in \{1,\ldots,m\}$ ;
$e_m(i,i) := 1, \qquad \forall\, i \in \{1,\ldots,m\}$ ;
**for** $k := 1$ ***to*** $n - m$ **do**
    $w = m + k$ ;
    $e_w(i,j) := e_{w-1}(i,j), \qquad \forall i,j \in \{1,\ldots,w-1\}$ ;
    $e_w(w,i) := 0,\ e_w(i,w) := 0 \qquad \forall i \in \{1,\ldots,w\}$ ;
    $e_w(w,k) := 1$ ;
    $e_w(w,i) := e_w(k,i), \qquad \forall i \in \{1,\ldots,k-1\}$ ;
**if** $k$ *even* **then**
    interchange the $w^{\text{th}}$ and the $(w-1)^{\text{th}}$ row;
    $e_w(j,i) := e_w(i,j), j = w - 1 \,\&\, \forall i \in \{1,\ldots,w-1\}$ ;
    $e_w(j,i) := e_w(i,j), \quad j = w \ \&\ \forall i \in \{1,\ldots,w\}$;
**next** $k$ ;
```

This algorithm probably generates a solution if l_x is even; if l_x is even, there seem to exist two different solutions, only one of which will be generated by the algorithm. Probably and seemingly, since we don't know how to prove the correctness of this claim. However, ample numerical calculations did not produce a falsification of this conjecture so we are strongly convinced of its correctness. If there exists a second solution y with $\boldsymbol{E}_{y,y} = \{e_y(i,j)\}$, then $e_y(l_x - i, l_x - j) = e_x(i,j)$ for all $1 \leq i,j \leq l_x$. As an example of the symmetry of the structure of such an ***E*** (and such symmetry always arises if m divides l_x) , we show the solution (omitting zero's) for $|A| = 4$ and $l_x = 12$ as generated by the algorithm above:

$$\begin{array}{cc}
\begin{array}{c}
\alpha\\ \beta\\ \gamma\\ \delta\\ \beta\\ \alpha\\ \delta\\ \gamma\\ \alpha\\ \beta\\ \gamma\\ \delta
\end{array}
&
\begin{pmatrix}
1 & & & & & 1 & & & 1 & & & \\
 & 1 & & & 1 & & & & & 1 & & \\
 & & 1 & & & & & 1 & & & 1 & \\
 & & & 1 & & & 1 & & & & & 1\\
 & 1 & & & 1 & & & & & 1 & & \\
1 & & & & & 1 & & & 1 & & & \\
 & & & 1 & & & 1 & & & & & 1\\
 & & 1 & & & & & 1 & & & 1 & \\
1 & & & & & 1 & & & 1 & & & \\
 & 1 & & & 1 & & & & & 1 & & \\
 & & 1 & & & & & 1 & & & 1 & \\
 & & & 1 & & & 1 & & & & & 1
\end{pmatrix}
\end{array}$$

Table 20: Minimum values for $\|\boldsymbol{x}\|^2$ in the columns labeled $|A| = 2$, etc. for different sequence lengths l_x. For comparison, we also show values of $2^{l_x} - 1$, the minimum of $\|\boldsymbol{x}\|^2$ in case $l_x \leq |A|$.

l_x	$2^{l_x} - 1$	$\|A\| = 2$	$\|A\| = 4$	$\|A\| = 8$
10	1023	10871	1723	1033
15	32767	2124045	97023	34473
20	1.0E06	3.9E09	6.0E06	1.0E06

Evidently, the minimal value of $\|\boldsymbol{x}\|^2$ is much bigger than $2^{l_x} - 1$ in case $l_x > |A|$ and the difference grows rapidly with increasing $l_x/|A|$. Table 20 shows these differences for some values of $|A|$.

Of course, these differences do affect the precision with which $C(x)$ is computed when using the boundaries in (5) for nonrestricted alphabets, especially so when $|A|$ is small. Let $C^*(x)$ denote the complexity of x where $x_{\min}$ is calculated with the above algorithm. Then Figure 1 shows plots of the ratio $C(x)/C^*(x)$ against l_x for various alphabet sizes $|A|$.

These plots clearly show that for smaller alphabets, $C(x)$ underestimates $C^*(x)$ substantially; for alphabets with $|A| \geq 20$, the underestimation seems negligeable for quite a range of sequence lengths.

Notes

1. To calculate complexity, we used the algorithm outlined in Appendix B to calculate $x_{\min}$.
2. We merged the groups with SES-scores of 0 and 1 (4 and 102 subjects respectively) and the groups with SES-scores of 8 and 9 (10 and 2 subjects respectively).
3. Actually, we could also have decided to retain as much as possible of the encoded life histories. For many individuals, this would have meant useful encodings beyond the age of 300 months; for many others, it wouldn't have made any difference. Analyses with different kinds of censoring did not lead to substantive differences in outcomes, so we arbitrarily decided to present the analysis on the data censored at 300 months.

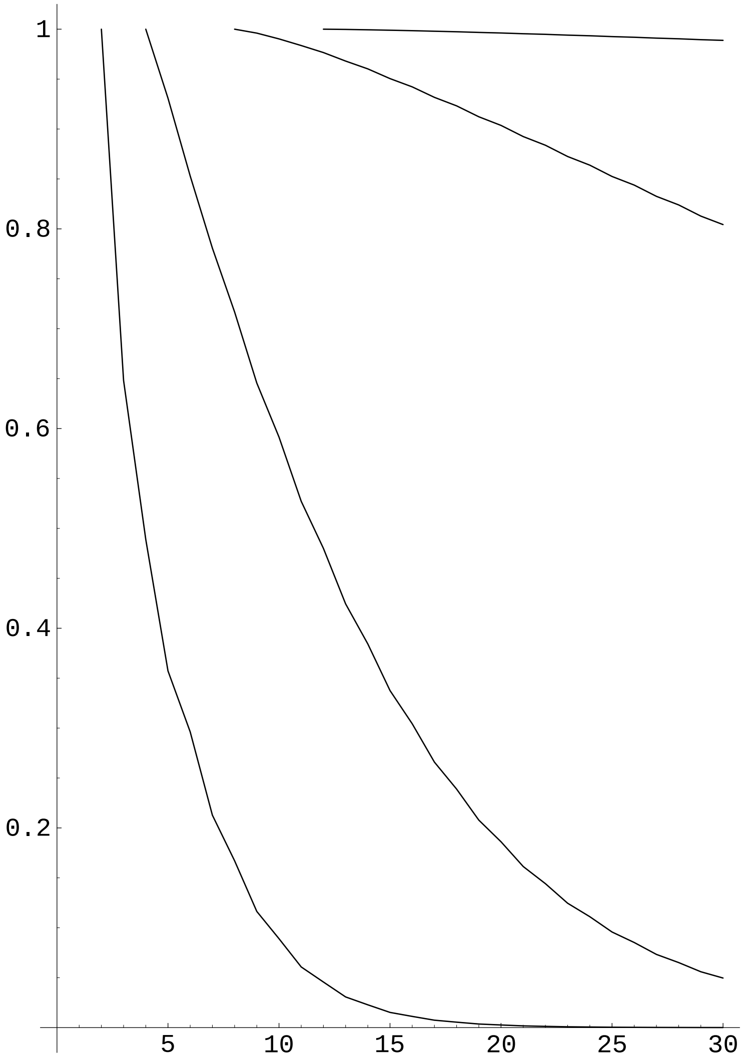

Figure 1. Plots of $C(x)/C^*(x)$ against sequence length (horizontal axis), where $C^*(x)$ uses the algorithm of Appendix B to calculate the minimum vector length given the alphabet size $|A|$. From left to right, the curves pertain to alphabets of 2, 4, 8 and 12 tokens.

References

ABBOTT, A. and FORREST, J. (1986), "Optimal Matching Methods for Historical Sequences", *Journal of Interdisciplinary History, 15*, 471-94.

ABBOTT, A and TSAY, A. (2000), "Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect", *Sociological Methods & Research, 29(1)*, 3-33.

CLOTE, P. and BACKOFEN, R. (2000), *Computational Molecular Biology, An Introduction*, Wiley.

BEZEM, M.A. and KEIJZER, M. (1997), "Generalizing Hamming Distance to Finite Sets" in Patterson, D. (Ed.) *Proceedings PACES/SPICIS'97*, pp. 148-153, Nanyang Technological University, Singapore.

DIJKSTRA, W. (1999), "A New Method for Studying Verbal Interactions in Survey Interviews", *Journal of Official Statistics, 15*, 67-85.

DIJKSTRA, W. and TARIS, T. (1995), "Measuring the Agreement Between Sequences", *Sociological Methods & Research, 24(2)*, 214-31.

DRAISMA, S. (2000), *RESPONSE, A Simulation Model of Question Answering in Survey Interviews*, Febo: Enschede.

ELZINGA, C. H. (2003), "Sequence Similarity: A Non-Aligning Technique", *Sociological Methods & Research, 31(4)*, 3-29.

GOTOH, O. (1982), "An Improved Algorithm for Matching Biological Sequences", *Journal of Molecular Biology, 162*, 705-708.

GUSFIELD, D. (1997), *Algorithms on Strings, Trees and Sequences. Computer Science and Computational Biology*, Cambridge University Press.

HAMMING, R.W. (1950), "Error-Detecting and Error-Correcting Codes", *Bell System Technical Journal, 2*, 147-160.

HEISER, W.J. and MEULMAN, J.J. (1997). "Representation of Binary Multivariate Data by Graph Models Using the Hamming Metric", in E. Wegman and S. Azen (Eds.), "*Computing Science and Statistics*", Interface Foundation of North America, Inc.

KNUTH, D. E. (1997), *The Art of Computer Programming. Vol. 1. Fundamental Algorithms*, Addison-Wesley.

MAYNARD, D.W. and N.C. SCHAEFFER, N.C. (2002), "Standardization and Its Discontents", in *Standardization and Tacit Knowledge, Interaction and Practice in the Survey Interview*", Eds. D.W. Maynard, H. Houtkoop-Steenstra, N.C. Schaeffer and J. van der Zouwen, Wiley.

NEEDLEMAN, S.B. and C.D. WUNSCH (1970), "A General Method applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins", *Journal of Molecular Biology, 48*, 443-453.

PETKOVŠEK, M., H.S. WILF and D. ZEILBERGER (1996), *A=B*, A.K. Peters.

SLOANE, N.J.A. and PLOUFFE, S. (1995), *The Encyclopedia of Integer Sequences*, Academic Press.

STANLEY, R.P. (1997), *Enumerative Combinatorics, Vol. I.* Cambridge University Press: Cambridge.

WU, L. L. (2000), "Some Comments on 'Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect'", *Sociological Methods & Research, 29(1)*, 41-64.