

**SEQUENCE ANALYSIS: METRIC REPRESENTATIONS OF
CATEGORICAL TIME SERIES**

CEES H. ELZINGA

Department of Social Science Research Methods
Vrije Universiteit Amsterdam,
The Netherlands

Department of Social Science Research Methods
Vrije Universiteit Amsterdam,
(Metropolitan, Room Z439)
de Boelelaan 1081
1081 HV Amsterdam
The Netherlands
telephone: +31-20-5986889
ch.elzinga@fsw.vu.nl

Home:
Veendijk 5
8481 JC Nijetrijne
The Netherlands
telephone: +31-561-481909
c.elzinga2@kpnplanet.nl

ABSTRACT. This paper considers sequence analysis as the problem of constructing metric representations of categorical time series. It is argued that the fundamental problem of sequence analysis is to construct metrics and similarity measures pertaining to attributes of pairs of sequences, such that these attributes are meaningful within the context of substantive social science theories. Four classes of alternatives to Optimal Matching are presented, algorithms are provided and pertaining geometries are discussed. These alternatives preserve attributes that are meaningful and relevant in a wide diversity of substantive applications. The classes of models are extended, and algorithms adapted, to handle duration in two different ways. The metrics are illustrated by applying them to published data sets from the domains of labor market entry careers and family formation histories.

Key Words: Sequence Analysis, Metrics, OM, Sequence Comparison Algorithms, Categorical Time Series

1. INTRODUCTION

In sequence analysis, a single datum consists of a pattern of distinct states or events over time, such as a life history, a sequence of moves in dance or a job career, and the analysis consists of quantifying the (dis-)similarities or distances amongst a set of such patterns. The goal is to use this quantification to efficiently describe the set of patterns observed or use it as a basis to construct a dependent or independent variable for some model of related phenomena. Stated in such general terms, sequence analysis seems to be a useful tool to find answers to interesting and legitimate questions.

Over the past decades, sequence analysis has been identified with Optimal Matching (OM) since that method was the only one available to quantify distances between categorical time series.

The development of OM started in the seventies and the technique has been amply described by Sankoff and Kruskal (1983). Abbott and Forrest (1986) were the first to introduce OM into the social sciences and used it to describe movement patterns in folk-dance and since then, an abundance of applications of OM has appeared in the social science literature. Abbott and Tsay (2000) reviewed many of these and more applications have appeared since (e.g. Stovel 2001; McVicar and Anyadike-Danes 2002; Widmer, Levy, Pollien, Hammer, and Gauthier 2003; Stovel and Bolan 2004; Brüderl 2004; Billari and Piccarreta 2005). A more recent review of the method is provided by Brüderl and Scherer (2005) and some of its statistical properties have been investigated by Wilson (2006). The use of OM in the social sciences has been severely criticized for a variety of reasons (e.g. Dijkstra and Taris 1995; Levine 2000; Wu 2000; Elzinga 2003) and only recently, an alternative to OM was proposed (Elzinga 2005).

Basically, OM expresses distances between sequences in terms of the minimal amount of energy, measured in terms of edit operations, that is required to change two sequences such that they become identical. However, it is not easy to understand what demographical, economical or sociological models could be connected to reasoning about minimal energy that changes social science categorical time series. For such changes of categorical time series simply cannot be observed. Therefore, the most serious criticism that has been raised against the use of OM has been that the metric has no social science interpretation. Of course, the criticisms were raised because OM seems to “work”: its application often generates results that do make sense in a variety of substantive areas.

Another criticism raised to OM has been (e.g. Wu 2000) that it does not handle time in a proper way: OM operates on pairs of strings, the characters representing the states, and the characters/states cannot have additional properties like e.g. duration. OM users work around this problem by creating a (sub-)string of, say, 4 X’s if state X has a duration of 4 units of time.

So, we have the unpromising problem that, on the one hand, social science theories do not lead to spatial representations of categorical time series and that, on the other hand, a practically useful method to generate such spatial representations seems to have no social science interpretation nor does it handle durations in a proper way.

The purpose of the present paper is to present a range of alternatives to OM, alternatives that can be interpreted in terms of the phenomena studied, whose spatial properties allow for the familiar (squared) Euclidean distance and that have feasible algorithms. Furthermore, in

TABLE 1. Encoding of family formation statuses.

Status	Code	(≥ 1 child)
Living Single	S	(SC)
Living together Unmarried	U	(UC)
Living together Married	M	(MC)

a separate section we will discuss how these alternatives can be adapted to handle duration as a state property.

Each of the representations will be discussed by introducing an attribute of a pair of sequences and, each time using the same set of simple rules, we will then use the attribute to construct both a distance and a similarity. This approach has the advantages that the conceptual and geometrical interpretation of the representation are quite transparent and that distances and similarities are, conceptually and geometrically, closely related.

Actually using the methods discussed in this paper requires algorithms that calculate the distances and similarities. We will therefore discuss these algorithms; an elementary understanding of the algorithms is necessary to grasp the potential and limitations of the methods presented and it may open windows to new representations. All of the methods discussed in this paper have been implemented in the form of a software package named XXX that, including a manual, is freely downloadable from <http://home.xyz.ab/pqr/>. So, the methods discussed can be directly put to work in actual research.

We will not attempt to show that one particular method is superior to any other method: to do so, one would need social science theory from which a spatial representation logically follows or that puts non-trivial constraints on such a representation. So, there is no single, external criterion that can be used to evaluate and rank the various methods. But this does not imply that choosing a method is an arbitrary matter: one will have to defend a particular choice by reasoning from substantive issues, the purpose of the analysis and the results obtained with it.

On the other hand, the methods are different and we will try to highlight these differences by applying them to data that pertain to quite different domains: we will use family formation histories and we will use labor market entry careers of school leavers.

The family formation histories cover four birth cohorts of Austrian women, born between 1945 and 1964. These data, 2499 sequences in total, are a subset of the Family and Fertility Surveys, carried out in 25 countries between 1988 and 1999, under supervision of UN-ECE. An evaluation of these data is provided by Festy and Prioux (2002); other subsets of these data were discussed by e.g. Corijn and Klijzing (2001) and Fussell and Gauthier (2005). The family formation histories themselves consist of 144 monthly statuses, each specifying a family formation state for each month between the ages of 18 and 30 years. 6 distinct states were discerned: living Single, living together Unmarried and living together Married and each of these states could be extended with a C, indicating that the woman lived together with at least one Child (too). So, a typical history looks like S/20 M/30 MC/94, meaning that the woman lived, since her 18th, single for 20 months, then lived together with just a husband for 30 months and then extended the family with at least one child. In Table 1 we summarize the encodings of family formation statuses.

TABLE 2. Encoding of labor market entry careers.

Status	Code	Status	Code
Employed	E	School	S
Unemployed	U	Further education	F
Training	T	Higher education	H

The labor market entry career data have been amply discussed by McVicar and Anyadike-Danes (2002). These data consist of 712 records of labor market statuses of young persons in Northern-Ireland, each record consisting of encoded activities of 72 consecutive months and starting from the moment that compulsory education ended (at age 16 in 1993). In Table 2, we summarize the encoding of these labor market statuses.

The rest of this paper is structured as follows. The next section will deal with the concepts of metric and similarity and explains our general approach of defining an attribute and using it to construct a metric and a similarity. In Section 3, pretending that durations are irrelevant, we discuss four fundamental metrics and relate one of them to the representation that OM generates. Finally, we use Section 3 to make some remarks on the geometry that is implied by some of the metrics. In Section 4, we discuss the handling durations and we will adapt the metrics and algorithms accordingly. In the final Section 5, we evaluate and discuss a few technical problems.

2. STRINGS, SIMILARITY AND METRICS

Categorical time series come as strings or sequences of characters, each character denoting one particular state or event. In the sequel, we shall write $\Sigma = \{a, b, c, \dots\} = \{\sigma_1, \dots, \sigma_p\}$ for the finite set of characters, i.e. the alphabet, from which the sequences are constructed. For the set of labor market entry careers, the alphabet $\Sigma = \{E, U, T, S, F, H\}$ is defined by Table 2 and is of size $|\Sigma| = 6$. The sequences themselves arise by concatenation of characters from Σ and we will write $\Sigma^* = \{u, v, w, x, y, z, \dots\}$ for the set of all sequences that can be constructed from Σ .

Similarity is a concept that applies to pairs of objects, in the present context to pairs of strings. If we say that two objects are similar, we mean that these objects have a certain amount of some attribute in common. If the objects are perfectly similar with respect to that attribute, the objects must possess that attribute to the same degree. If the objects are not perfectly similar, then the amount of attribute shared cannot exceed the minimum of the amount of attribute possessed by either of the objects. Hence, if we presume that we have a number $A(x, y)$ that quantifies the amount of attribute commonly possessed by objects x and y , we must have that

$$A(x, y) = A(y, x), \tag{1}$$

$$0 \leq A(x, y) \leq \min\{A(x, x), A(y, y)\}$$

and a quantification of similarity $s(x, y)$ should be at least weakly monotone with $A(x, y)$. For example, we might identify the attribute $A(x, y)$ with the number of distinct common

TABLE 3. Similarities according to (3) (lower diagonal) and (4) (upper diagonal) for the sequences as shown in the first column and $A(x, y)$ equals the number of distinct common characters.

S U T	1	.16	.25
U H F E	.29	1	.20
S E	.41	.35	1

states encountered in x and y . Evidently, this non-negative number cannot exceed the smallest number of distinct states as encountered in either sequence and the attribute is clearly symmetric.

In general, we will not consider an attribute that satisfies (1) to be a similarity measure for similarity not only depends upon the amount of attribute shared but also on the amounts of attribute present in either of the objects. For example, consider the labor market entry careers (durations ignored)

$$x = \text{S U T}, y = \text{U H F E}, z = \text{S E}. \quad (2)$$

Despite the fact that $A(x, z) = 1 = A(y, z)$, we have that $s(x, z) > s(y, z)$, simply because y has more distinct characters than x . So, we have that the similarity $s(x, y)$ should be decreasing in $A(x, x)$ and $A(y, y)$ and increasing in $A(x, y)$. Because of interpretability we might further demand that similarity $s(x, y)$ is bounded on some closed interval, e.g. $0 \leq s(x, y) \leq 1$, and that, for the sake of comparability, that $s(x, y)$ is dimensionless or unit-free. With these restrictions on the structure of a similarity measure, a multitude of possibilities remains open. We mention just two of these: first, there is the obvious

$$s(x, y) = \frac{A(x, y)}{\sqrt{A(x, x) \cdot A(y, y)}}. \quad (3)$$

Clearly, the measure is decreasing in $A(x, x)$ and $A(y, y)$, it is increasing in $A(x, y)$, dimensionless and bounded by the closed interval $[0, 1]$. Note that the denominator is just the geometric mean of the amounts of attribute in x and in y . A second possibility is the so called Tanimoto-coefficient (Tanimoto 1957):

$$T(x, y) = \frac{A(x, y)}{A(x, x) + A(y, y) - A(x, y)}, \quad (4)$$

measuring the ratio of the amount of common attribute to the total amount of attribute present in both objects. Clearly, the Tanimoto-coefficient is nonlinear in $A(x, y)$, and the more so the shorter the sequences are, whereas $s(x, y)$ behaves linear and $T(x, y) \leq s(x, y)$. In Table 3, we show both coefficients for the example careers shown above. It should be stressed that (3) and (4) are just examples of similarity indices and that there is abundant literature on similarity measures (e.g. Gower 1971; Gower and Legendre 1986; Batagelj and Bren 1995; Holliday, Hu, and Willett 2002; Wang 2006). For reasons to become clear, we will adopt the habit of using (3) as our similarity measure.

Above, we connected the concept of a quantified attribute to a similarity measure. Below, we will connect quantified attributes to the notion of a distance function or metric.

Like similarity, distance is a concept that pertains to pairs of objects and, again like similarity,

it is non-negative and symmetric too. However, for two identical objects, we have that similarity is maximal while distance is minimal. In general, when distance and similarity derive from the same attribute, we must have that distance and similarity are anti-monotone, i.e., if two objects are quite similar, their distance must be quite small.

Formally, we say that a function $d(x, y)$ that maps pairs $(x, y) \in \Sigma^* \times \Sigma^*$ into the real numbers is a metric or, equivalently, a distance if the mapping satisfies

- a. $d(x, x) = 0$ (uniqueness of place),
- b. $d(x, y) > 0$ if $x \neq y$ (uniqueness of elements),
- c. $d(x, y) = d(y, x)$ (symmetry),
- d. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality).

If such a d exists, the pair (Σ^*, d) is called a metric space and d is said to be a metric on Σ^* . For a metric to be useful, it must pertain to an attribute that is meaningful in the subject area where it is applied. So, the problem arises of which attributes to select and how to quantify these?

Fortunately, it is not difficult to prove that if an attribute $A(x, y)$ satisfies the boundary conditions (1), then this is a sufficient condition for the function

$$d(x, y) = A(x, x) + A(y, y) - 2A(x, y), \quad (5)$$

to be a metric over Σ^* . This is an interesting fact and we will use it throughout the rest of this paper since it allows us to directly construct metrics from attributes that satisfy the quite simple boundary conditions of (1). For example, with the interpretation of $A(x, y)$ as the number of distinct common states of x and y , it is almost immediate that $A(x, y)$ satisfies the boundary conditions and hence that (5) is a metric. This metric expresses distance as the number of distinct non-common states of x and y since we have that

$$d(x, y) = (A(x, x) - A(x, y)) + (A(y, y) - A(x, y)). \quad (6)$$

Interestingly, this metric is a squared Euclidean distance. To see this, we take the example labor market entry careers again. These sequences were constructed from an alphabet Σ containing 6 characters. Of these characters we arbitrarily fix the permutation to $\sigma_1 \cdots \sigma_6 = \text{EUTSFH}$. Now we assign a 6-dimensional vector $\mathbf{x} = (\dot{x}_1, \dots, \dot{x}_6)$ to each career x such that $\dot{x}_i = 1$ if the i^{th} character from Σ occurs in x and $\dot{x}_i = 0$ if this character does not occur in x . So, we construct the vectors

$$\begin{aligned} \text{SUT} = x &\rightarrow \mathbf{x} = (0, 1, 1, 1, 0, 0), \\ \text{UHFE} = y &\rightarrow \mathbf{y} = (1, 1, 0, 0, 1, 1), \\ \text{SE} = z &\rightarrow \mathbf{z} = (1, 0, 0, 1, 0, 0). \end{aligned}$$

Now it is not difficult to see that our attribute $A(x, y)$ equals the inner product of the representing vectors:

$$A(x, y) = \mathbf{x} \cdot \mathbf{y} = \sum_i \dot{x}_i \dot{y}_i. \quad (7)$$

Hence, we have that

$$d(x, y) = \sum_i \dot{x}_i^2 + \sum_i \dot{y}_i^2 - 2 \sum_i \dot{x}_i \dot{y}_i \quad (8)$$

$$= \sum_i (\dot{x}_i - \dot{y}_i)^2. \quad (9)$$

Apparently, $d(x, y)$ equals the squared Euclidean distance between x and y in the space spanned by the representing vectors. Now we also easily see that $A(x, x)$ equals the squared length $\|\mathbf{x}\|^2$ of the vector \mathbf{x} and that

$$s(x, y) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}, \quad (10)$$

i.e. $s(x, y)$ equals the cosine of the angle between the representing vectors \mathbf{x} and \mathbf{y} ; the smaller the angle, the greater the similarity.

So, constructing metrics and similarities according to the formal structures (5) and (3) on the basis of quantified attributes that satisfy the boundary properties (1) definitely has advantages:

- the resulting distance has the substantive interpretation of a total amount of non-common attribute and
- is geometrically interpretable as a squared Euclidean distance,
- the similarity has the substantive interpretation of a relative amount of common attribute and
- the geometric interpretation of the size of an angle between representing vectors in a Euclidean space,
- the amount of attribute possessed by each sequence corresponds to the squared length of the representing vector in a Euclidean space.

Given the example interpretation of $A(x, y)$ as the number of distinct common states, the numerical evaluation of $A(x, y)$ is cheap. For suppose that we have two sequences x and y of lengths m and n and that the size of the alphabet equals $|\Sigma| = p$. Then it is not difficult to see that the amount of time that is required to calculate $A(x, y)$ is proportional to $p(m+n)$; i.e the algorithmic complexity is linear in the length of the sequences.

So, we conclude that our example attribute leads to a metric that is easy to calculate and that has a straightforward interpretation. But does this metric make any sense?

No, it does not make sense in most social science applications. For consider the labor market entry careers $x = \text{SUE}$ and $y = \text{SEU}$; one now easily calculates $d(x, y) = 0$ and $s(x, y) = 1$ whereas, obviously, x points at successful integration while y represents a failure. Yet, our metric does not see a difference! Similarly, in the domain of family formation, the histories $u = \text{SC MC}$ and $v = \text{MC SC}$ are quite different, to say the least, but again we would calculate $d(u, v) = 0$. Obviously, the metric fails because it does not recognize differences in the order of events. A metric that is formally simple and elegant does not guarantee that it will have any practical value. We only introduced it because it allowed us to easily illustrate some concepts that we will rely on throughout the rest of this paper. In the next section, we will introduce more subtle attributes that do, albeit in varying degree, use the order of events in categorical time series. However, in the last section, we will show that capitalizing on timing instead of capitalizing on order may lead to interesting results as well.

3. ATTRIBUTES FROM SUBSEQUENCES

In this section, we will treat five basic metrics, including OM. It will appear that four of these and a special case of OM can be constructed from quantified attributes of a pair of strings, using the structure (1). Therefore, we will introduce and label the metrics by discussing these attributes.

A few remarks on our notation seem justified. Today, there is no standard notation for concepts and properties of strings. Our notation closely follows the notation used in Rozenberg and Salomaa (1997) but does not coincide with it. The reason is that we deal with abstract properties of strings on the one hand and, on the other hand, with the more familiar structures like sets, linear algebra and number systems for which there is a well-established notation. Therefore, our notation sometimes seems to be ambiguous. For example, we use $|\cdot|$ in two ways: to indicate the length of a string, in which case the argument is lower-case, and to denote the cardinality of sets, in which case the argument is upper-case. Similarly, we use x^i to refer to a prefix and we also use x^p to indicate the p^{th} power of a real number. Such ambiguity can only be avoided at the cost of introducing non-standard notation or funny fonts and characters. We do not want this. We trust that the context of the notation used will adequately alert the reader to change his frame of reference if necessary.

3.1. LCP: The Longest Common Prefix. If we compare the lives of two individuals, these lives almost invariably start in the state “living together with at least one parent”. If we confine ourselves to partnership histories, most histories will start with the persons living without a partner. Similarly, if we confine ourselves to educational/employment careers, most careers start at school. In general, categorical time series that start in the same state will be less different, the longer the series follow the same pattern of transitions: $x = abcd$ and $y = abce$ will be considered as more alike or as less distant than x and $z = abef$. This very simple but sensible idea is formalized by a metric that uses the length of the first common pattern of states that includes the first state: the length of the longest common prefix. In order to formalize this idea, we first introduce some concepts and notation. We already introduced the set of all strings Σ^* . This set always contains a special string that has no characters: the empty string $\lambda \in \Sigma^*$. If x is a string that consists of n (not necessarily different) symbols, we say that x is n -long and we denote this fact by writing $|x| = n$. So, if $x = abac$, we have that $|x| = 4$ and we always have that $|\lambda| = 0$. To refer to the symbol on the i^{th} position of a string x , we write x_i and for the n -long string x , we write $x = x_1 \cdots x_{|x|} = x_1 \cdots x_n$. So, if $x = abac$, we have that $x_1 = a = x_3$ and $x_{|x|} = c$.

For any n -long string x and $0 \leq k \leq n$, we write $x^k = x_1 \cdots x_k$ and we call x^k the k^{th} prefix of x . Indeed, if x is n -long, we have $x^n = x$ and $x^0 = \lambda$ and if $x = abac$, we have that $x^3 = aba$. Now consider pairs (x, y) of, not necessarily equally long, strings and suppose that u is a k -long prefix of both x and y ; i.e. $x = ux_{k+1} \cdots x_n$ and $y = uy_{k+1} \cdots y_m$. Then we say that the prefix u is common to x and y .

With these concepts and notation, we are in a position to define an attribute $A(x, y)$ that satisfies the boundary conditions (1) and we will use that attribute to define a simple metric over Σ^* . Thereto, we assign to each pair of sequences (x, y) a set $\mathcal{P}(x, y)$ that contains all common nonempty prefixes of x and y : $\mathcal{P}(x, y) = \{u \neq \lambda : x^{|u|} = u = y^{|u|}\}$. Since the prefix of any length is unique, the size of this set corresponds to the length of the longest common

TABLE 4. Distances $d_{\mathcal{P}}$ (lower diagonal; equation (12)) and similarities $s_{\mathcal{P}}$ (upper diagonal; equation (13)) between the family formation histories as shown in the leftmost column.

$v = \text{S U}$		0.82	0.71	0.63	0.0
$w = \text{S U M}$	1		0.87	0.77	0.0
$x = \text{S U M MC}$	2	1		0.89	0.0
$y = \text{S U M MC SC}$	3	2	1		0.0
$z = \text{U M MC}$	5	6	7	8	

prefix of x and y . So, we define the attribute of the length of the longest common prefix as

$$A_{\mathcal{P}}(x, y) = |\mathcal{P}(x, y)| \quad (11)$$

It is now easy to see that this attribute satisfies the boundary conditions (1). Therefore, we use it to define the metric

$$\begin{aligned} d_{\mathcal{P}}(x, y) &= A_{\mathcal{P}}(x, x) + A_{\mathcal{P}}(y, y) - 2A_{\mathcal{P}}(x, y) \\ &= |\mathcal{P}(x, x)| + |\mathcal{P}(y, y)| - 2|\mathcal{P}(x, y)| \\ &= |x| + |y| - 2|\mathcal{P}(x, y)| \end{aligned} \quad (12)$$

and its associated measure of similarity

$$s_{\mathcal{P}}(x, y) = \frac{A_{\mathcal{P}}(x, y)}{\sqrt{|x| \cdot |y|}}. \quad (13)$$

Obviously, $d_{\mathcal{P}}$ measures distance between strings in terms of the number of symbols of both strings that do *not* belong to a common prefix.

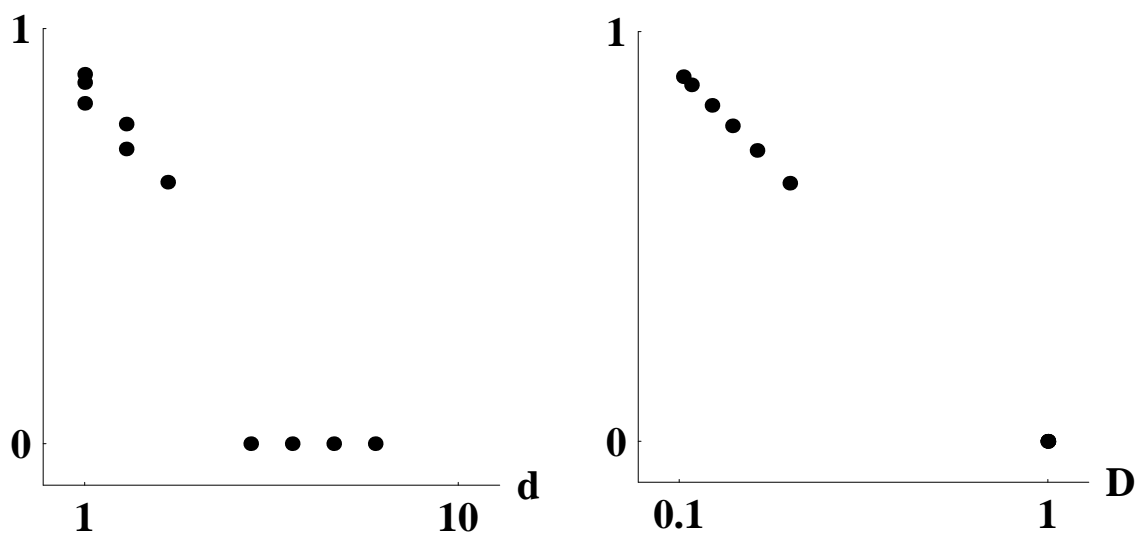
It is interesting to study the consequences of this construction when applied to a very small subset of our family formation histories. This subset and the resulting numbers are shown in Table 4 and in the left panel of Fig. 1, we plotted the similarities $s_{\mathcal{P}}$ against the distances $d_{\mathcal{P}}$. As is immediate from this plot, the relation between similarity and distance is only weakly monotone: Distances range from 5 to 8 while similarity equals zero for these pairs of histories and for identical distances, we observe differing similarities. So, the distances and similarities are poor indicators of each other. The reason for this state of affairs is of course in the structure (12) of the Euclidean metric since the amounts of attribute $A_{\mathcal{P}}(x, x)$ and $A_{\mathcal{P}}(y, y)$ possessed by either sequence play a major role. Geometrically, we have that, for a fixed angle between representing vectors, distance is fully determined by the lengths of the vectors. So, if we want a distance that is insensitive to length or, equivalently, to the absolute amount of attribute of either sequence, we have to scale all the vectors to unit length and then consider their distances. This implies that we construct vectors $\mathbf{x}' = \mathbf{x} / \|\mathbf{x}\|$ and determine $d(\mathbf{x}', \mathbf{y}')$:

$$\begin{aligned} d(x, y) &= d(\mathbf{x}', \mathbf{y}') = \mathbf{x}'^T \mathbf{x}' + \mathbf{y}'^T \mathbf{y}' - 2\mathbf{x}'^T \mathbf{y}' \\ &= 2(1 - s_{\mathcal{P}}(x, y)). \end{aligned} \quad (14)$$

But then it is more convenient to consider the further rescaled

$$D_{\mathcal{P}}(x, y) = 1 - s_{\mathcal{P}}(x, y). \quad (15)$$

FIGURE 1. Plots of similarities (vertical axes) against distances (horizontal axes). In the left panel, the distances and similarities have been taken from Table 4. In the right panel, the distances from Table 4 have been normalized according to equation (15).



Clearly, and as shown in the right-hand panel of Fig.1, this so-called normalized distance is a perfect dissimilarity measure: similarity and normalized distance are now strongly anti-monotone. Furthermore, the reader notes that the normalized $D_{\mathcal{P}}$ is unit-free whereas $d_{\mathcal{P}}$ is not.

Of course, the effect of switching to normalized distances will strongly depend upon the set of sequences; the small set of Table 4 was selected to show the effects discussed. In the next section, we will concisely discuss an application of the LCP-metric where normalizing has only negligible effects.

3.2. An application: LCP reversed. In this section, we apply the LCP-metric to the labor market entry careers as discussed in the Introduction to this paper (see also Table 2). However, judging the successfulness or unsuccessfulness of a career is impossible if one ignores the durations of the different states; being unemployed is fairly insignificant if only lasting for a few months. So we included durations in a way (MST) that will only be explained in the section on handling durations.

Including durations, we calculated distances and normalized distances for all pairs of the 712 careers and then clustered the careers through the K-means algorithm (e.g. Hartigan 1975) into either 5 (e.g. McVicar and Anyadike-Danes 2002) or 7 (e.g. Brzinsky-Fay 2006) clusters. Of course these 4 cluster solutions are different. How much different is expressed by the Jaccard-coefficient (Jaccard 1912) which calculates the proportion of pairs of sequences that are assigned to the same cluster in two different clusterings. These Jaccard-coefficients are shown in Table 5. Note that applying the normalized LCP-metric instead of the unnormalized version leads to quite different clustering solutions for the the Jaccard-indices are only .44 for the 5-cluster solutions and .64 for the 7-cluster solutions. In the upper part of Table 6 we

TABLE 5. Clusterings of labor market entry careers. Jaccard coefficients J for clusterings as obtained with a K-means algorithm applied to squared Euclidean distances. Each solution was the optimal one taken from 100 runs with randomly chosen initial configurations, i.e. the solution with maximum $R^2 = SS_{Between}/SS_{Total}$. The R^2 -values are shown in the last column. The acronyms in the first column pertain to either LCP- or LCT-metrics; the subscript “n” indicating that the metric was normalized and the digit pertains to the number of clusters.

metric	J								R^2
LCP-5	1								.21
LCP-7	.36	1							.28
LCP _n -5	.44	.60	1						.23
LCP _n -7	.37	.64	.53	1					.29
LCT-5	.15	.14	.16	.13	1				.35
LCT-7	.13	.12	.14	.12	.62	1			.42
LCT _n -5	.15	.14	.16	.13	1.0	.62	1		.35
LCT _n -7	.13	.12	.14	.12	.61	.97	.61	1	.42

TABLE 6. Clusters of labor market entry careers. Average durations in months per state per cluster and the cluster-sizes (last column) when using the normalized LCP- (upper part) and the normalized LCT-metric (lower part). The averages for all 712 subjects are shown in the middle line.

cluster	T	E	U	F	H	S	size
1	25.76	39.95	5.11	1.05	0.00	0.13	91
2	6.73	31.76	7.99	14.19	6.86	4.47	385
LCP _n -5 3	2.91	27.81	4.53	25.06	11.28	0.41	95
4	0.65	12.84	3.44	4.09	23.87	27.09	95
5	0.0	69.80	2.20	0.0	0.0	0.0	46
Average	7.40	32.21	6.18	11.70	8.40	6.10	712
1	10.77	25.92	8.21	11.48	9.37	6.23	139
2	10.55	16.45	23.69	12.14	3.35	5.80	111
LCT _n -5 3	1.83	37.15	1.25	28.75	0.0	3.02	102
4	0.22	3.47	0.97	14.08	36.67	16.59	117
5	9.83	54.78	1.58	3.33	0.06	2.41	243

describe the clusters by showing their sizes and by showing the average duration (in months) spent in each of the six states. The averages of these durations for all of the 712 careers are also shown. Figures are shown in bold-face if they are far above these overall-averages. So, we observe that the first cluster is dominated by careers in which the amount of time spent in training by far exceeds the average time spent in training. Note that in none of the clusters, much more than an average amount of time is spent in unemployment.

Careers that are unsuccessful are those that end with prolonged spells of unemployment. If a cluster of such unsuccessful careers exist, we conclude that the LCP-metric, normalized or not, does not generate such a cluster in a 5-cluster solution, neither is such a cluster generated by a 7-cluster solution. However, the LCP-metric focusses on a common substring that starts on the first position of both sequences. On the other hand, if a young man or woman fails to integrate into the labor market, this will only become apparent later in his or her career. Therefore, a straightforward application of the LCP-metric can hardly be expected to lead to a cluster of unsuccessful careers: such careers will not be picked up as being very similar. So, we also applied the LCP-metric to the reversed careers or, equivalently, we used the attribute of the Length of the Longest Common Tail (Postfix) to build a metric with. Unnormalized and normalized distances were calculated according to this LCT-metric and these distances were used again to generate 5- and 7-cluster solutions. These results are shown in the lower parts of Tables 5 and 6. According to the Jaccard-coefficients, the clusterings are totally different from those based on LCP-distances as the pertaining coefficients in Table 5 do not exceed .16. We also note that the normalized and unnormalized distances do not lead to different clusterings: the pertaining Jaccard-coefficients equal .97 and even 1.0 for the 7-cluster solutions. We also note that the resulting clusters are more homogeneous than those resulting from the LCP-distances, as is demonstrated by the R^2 -coefficients as shown in the last column of Table 5. The most interesting result obtained with this “reversed LCP” is shown in the lower part of Table 6 as the second row now shows a cluster in which time is predominantly spent in unemployment. So, probably, the careers in this cluster are the unsuccessful careers - those with much unemployment concentrated in the last part of the 72-months period.

Apparently and remarkably, even with a most simple metric like “reversed LCP”, we are able to pick out a small group (less than 16%) that failed to integrate into the labor market within 72 months.

3.3. Substrings and Subsequences. A prefix is a part of a string that consists of adjacent characters, the first of which also appears on the first position of the string from which it was taken. Now we might relax or even drop either of these requirements and use the result to define a new attribute and a new metric. For example, we might drop the requirement that the rows of adjacent characters start on the same position in either of the sequences involved. This would leave us with the attribute that is known as the longest common substring (e.g. Gusfield 1997) and has a prominent place in computational biology. Instead, we might drop the requirement that the characters of the common object are all adjacent in both sequences. However, we will not be concerned with all these subtle variations and simultaneously drop both requirements: the common object may not have its first character appearing on the first position in either sequence and its characters need not be adjacent in the sequences involved. This leaves us with an object known as a (longest) common subsequence which we will concisely discuss in the next subsection. Because the concept of subsequence is so central to this paper, we first discuss some formal definitions and notation that are related to subsequences.

We say that the string v is a substring of a string x if there exist (possibly empty) strings u and w such that $x = uvw$. This definition of a substring implies that the symbols that are adjacent in a substring v are also adjacent in the string x itself. For example, if $x = abac$,

$u = bac$ and $v = ab$ are substrings of x but $w = aac$ is not. Clearly, the prefixes of x are special substrings of x . Hence, the concept of substring generalizes the notion of prefix.

We will not confine ourselves to substrings but instead take the generalization one step further to the idea of subsequence. Before we discuss a formal definition, we give some examples. Let $x = abac$: then λ , $u = b$, $v = bac$ and $w = bc$ are all subsequences of x . Note that we already said that $v = bac$ is a substring of x : every substring of x is also a subsequence of x . Next, we note that $w = bc$ is a subsequence of x although it is not a substring of x . Only in $w = bc$, the symbols occur in the same order as they occur in x . These identical orders are fundamental to the idea of subsequence: a subsequence of an n -long string x arises by taking away $0 \leq k \leq n$ symbols from x . If we take $k = 0$, we have that x is a subsequence of itself; if we take $k = n$, we have that λ is a subsequence of x and if $0 < k < n$, we end up with a string of symbols that occur in the same left-to-right order as they occurred in the original string. Formally, we say that $u = u_1 \cdots u_k$ is a k -long subsequence of x if there exist (possibly empty) strings $v_1, \dots, v_{k+1} \in \Sigma^*$ such that $v_1 u_1 v_2 \cdots v_k u_k v_{k+1} = x$ and we write $u \in x$ to denote this fact.

Now consider the strings $x = abac$ and $y = baca$. It is immediate that e.g. $u = ac$ is a subsequence of both x and y : we say that u is a common subsequence of x and y and we write $u \in (x, y)$. It is equally obvious that e.g. λ and $v = bac$ are common subsequences of x and y too. So, there is a nonempty set $\mathcal{S}(x, y)$ of common subsequences of x and y : $u \in \mathcal{S}(x, y)$ if and only if $u \in (x, y)$. $|\mathcal{S}(x, y)| \geq 1$ since we always have that $\lambda \in \mathcal{S}(x, y)$ for any pair (x, y) . Clearly, the set of all common prefixes is a subset of the set of all common subsequences: $\mathcal{P}(x, y) \subset \mathcal{S}(x, y)$.

Although each and every string occurs at most once in $\mathcal{S}(x, y)$, some subsequences occur more than once in a particular string. For example, with $x = abac$ and $y = baca$, we have that $ac \in \mathcal{S}(x, y)$ since $ac \in x$ and $ac \in y$. But we observe that ac is embedded twice in x : as $x_1 x_4$ and as $x_3 x_4$. It will prove convenient to have a special notation for the number of embeddings of particular subsequences in particular strings. Therefore, we will write $|x|_u = r$ if u has r distinct embeddings in x . In particular, we write $|x|_{ac} = 2$ and $|y|_{ac} = 1$.

In the subsections to come, we will discuss different properties of the set of common subsequences $\mathcal{S}(x, y)$ and use them as attributes that allow for the construction of metrics.

3.4. LCS: The Longest Common Subsequences. The first property of the set of common subsequences that we will use to construct a metric with, is the length of a longest element of \mathcal{S} . Thereto, it is instructive to consider 3 example family formation histories and their longest common subsequences (LCS's):

$$\begin{aligned} x &= \text{S U S M S U}, \\ y &= \text{U S SC MC}, \\ z &= \text{S U M S SC UC MC}. \end{aligned}$$

The LCS of the first pair of histories (x, y) is $u = \text{U S}$ with a length (LLCS) of only two states. This is not a big length in view of the fact that x and y have lengths of, respectively, 6 and 4 states. Therefore, most readers will not consider this pair to be very similar.

The pair (x, z) has its LLCS equalling 4 since the LCS equals $u = \text{S U M S}$ and the pair (y, z) also has its LLCS equalling 4 since y completely appears in z as $z_2 z_4 z_5 z_7$. The LCS of y and z could not have been longer since $|y| = 4$, so we are inclined to say that, of the three

TABLE 7. Distances and similarities for the family formation histories shown in the leftmost column (see Table 1 for encodings). The left table contains the quantities $d_{\mathcal{L}}$ (lower-diagonal part) and $s_{\mathcal{L}}$ (upper-diagonal part); the right table shows the quantities $d_{\mathcal{P}}$ and $s_{\mathcal{P}}$ according to the same format.

histories	x	y	z	x	y	z
S U S M S U = x		.41	.62		0	.31
U S SC MC = y	6		.76	10		0
S U M S SC UC MC = z	5	3		9	11	

pairs, the pair (y, z) is the most similar. Formalizing these intuitions amounts to defining the attribute

$$A_{\mathcal{L}}(x, y) = \max\{|u| : u \in \mathcal{S}(x, y)\} \quad (16)$$

Clearly, $A_{\mathcal{L}}$ satisfies the boundary conditions (1) so we can construct a metric and a similarity in the usual way:

$$\begin{aligned} d_{\mathcal{L}}(x, y) &= A_{\mathcal{L}}(x, x) + A_{\mathcal{L}}(y, y) - 2A_{\mathcal{L}}(x, y) \\ &= |x| + |y| - 2A_{\mathcal{L}}(x, y), \end{aligned} \quad (17)$$

$$s_{\mathcal{L}}(x, y) = \frac{A_{\mathcal{L}}(x, y)}{\sqrt{|x| \cdot |y|}}. \quad (18)$$

In Table 7, we show these quantities as calculated for the three example histories just discussed and compare these quantities with $d_{\mathcal{P}}$ and $s_{\mathcal{P}}$. The figures in the table illustrates a consequence of the fact that the LCS is always at least as long as the LCP. So, we have that $A_{\mathcal{L}}(x, y) \geq A_{\mathcal{P}}(x, y)$, equality holding only when either $x = y$ or when $A_{\mathcal{L}} = 0$. Consequently and with the same conditions for equality,

$$d_{\mathcal{P}}(x, y) \geq d_{\mathcal{L}}(x, y), \quad (19)$$

$$s_{\mathcal{P}}(x, y) \leq s_{\mathcal{L}}(x, y). \quad (20)$$

For shorter strings, evaluating $A_{\mathcal{L}}(x, y)$ is easy: take a particular subsequence of x , check if it occurs in y , proceed to the next subsequence and retain the longest one. However, in practice, this is not a feasible procedure since an n -long string has 2^n subsequences to check. Fortunately, there is a simple recurrence, probably due to Sankoff (1974) that, given $|x| = m$ and $|y| = n$, determines $A_{\mathcal{L}}(x, y)$ in an amount of time that is proportional to the product mn . Sankoff's algorithm is based upon a very simple observation: the value of $A_{\mathcal{L}}(x^i, y^j)$ is at least as big as that of $\max\{A_{\mathcal{L}}(x^{i-1}, y^j), A_{\mathcal{L}}(x^i, y^{j-1})\}$, the difference being at most equal to 1 if $x_i = y_j$. This observation leads to the recurrence

$$A_{\mathcal{L}}(x^i, y^j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ \max\{A_{\mathcal{L}}(x^{i-1}, y^j), A_{\mathcal{L}}(x^i, y^{j-1})\} & \text{if } i, j > 0 \text{ and } x_i \neq y_j \\ A_{\mathcal{L}}(x^{i-1}, y^{j-1}) + 1 & \text{if } i, j > 0 \text{ and } x_i = y_j \end{cases} \quad (21)$$

TABLE 8. Average similarities between family formation histories of four cohorts of Austrian women. Durations have been ignored ($s_{\mathcal{L}}$) in the first part of the table and included ($s_{\mathcal{L}}^*$) in the second part. Characteristic sequences are those that have the highest average similarity in the cohort.

Cohort	Size	$\bar{s}_{\mathcal{L}}$	sd	Char. Seq.	$\bar{s}_{\mathcal{L}}^*$	sd	Char. Seq.
1945-49	539	.67	.15	S M MC	.49	.13	S/50 M/9 MC/85
1950-54	558	.64	.14	S M MC	.44	.12	S/45 M/16 MC/83
1955-59	650	.60	.12	S U M MC	.41	.11	S/53 U/6 M/8 MC/77
1960-64	752	.57	.12	S U M MC	.39	.10	S/53 U/11 M/12 MC/68

A recurrence like (21) implies a dynamic algorithm: an algorithm that starts from a solution of a trivial subproblem, in this case $A_{\mathcal{L}}(x^0, y^j) = 0 = A_{\mathcal{L}}(x^i, y^0)$, and then proceeds to use this solution to solve ever bigger subproblems until it finally solves the target problem, in this case the evaluation of $A_{\mathcal{L}}(x^m, y^n)$. In fact, all of the algorithms to be discussed in this paper are of the dynamic type. Today, algorithms exist that are even faster than (21) (e.g. Eppstein et al. (1992a), Eppstein et al. (1992b), Rick (2000)) but these are not advantageous for the relatively short strings that we deal with in the social sciences.

3.5. De-Standardization and Similarity of Family Formation. In recent decades, we have seen dramatic changes in the transition to adulthood in many western countries. The timing and time-order of many important events, like e.g. leaving the parental home, entering partnership and entering parenthood, has changed and family formation has become a more extended and complicated sequence of events since many phenomena like e.g. cohabitation, living single, staying childless, extramarital parenthood and partnership dissolution, have become more accepted (e.g. Liefbroer and Goldscheider 2007). It has been suggested (e.g. Shanahan 2000; Brückner and Mayer 2005) that these changes have led to an increased de-standardization and complexity of the transition into adulthood. Elzinga and Liefbroer (2007) hypothesized that increased de-standardization should be manifest in family formation histories becoming more dissimilar over time. Elzinga and Liefbroer (2007) calculated average similarities between such trajectories for 19 Western countries, using a metric based on the number of matching subsequences (NMS) and did find the hypothesized decrease in most of the countries studied. In Table 8, we present the result of similar calculations, using the similarity $s_{\mathcal{L}}$, both ignoring and including information on durations. Indeed, ignoring and including durations, similarities clearly decrease and the characteristic sequences develop as expected since they include cohabitation in younger cohorts and show a tendency of postponing parenthood. Apparently, such backbone structures are easily and convincingly revealed by a simple, LCS-based metric.

3.6. Optimal Matching and Subsequences. At first sight, it may seem strange to discuss OM in the context of subsequence-based metrics. However, it will appear that there is a close connection between OM and $d_{\mathcal{L}}$.

OM generates an edit-distance: it measures the minimal amount of edits needed to transform a pair of strings (x, y) into a possibly different pair (x', y') such that x' and y' are equivalent

in some sense. Given a pair of strings (x, y) , the outcome of the evaluation of the edit-distance depends on the admissible edit-operations, the alphabet and the cost of applying the admissible edit-operations to the elements of the strings involved. This is not a very specific description, so let us be a bit more precise.

First, OM uses an alphabet that not only contains all of the events/symbols observed but additionally contains a “neutral” gap-symbol “-”. So, $\Sigma_{\text{OM}} = \Sigma \cup \{-\} = \{\sigma_1, \dots, \sigma_d\}$ with $\sigma_1 = -$. This gap-symbol allows us to be precise about what we mean with “equivalent” strings: we say that the n -long strings x and y are equivalent if they have equivalent symbols in the same position, i.e.

$$x \sim y \quad \text{if} \quad x_i \approx y_i, i \in [n] \quad (22)$$

and

$$x_i \approx y_i \quad \text{if} \quad (x_i = y_i \text{ or } x_i = - \text{ or } y_i = -). \quad (23)$$

Hence, according to the above definition, the strings $x = a - - - ba - c$ and $y = -dabacc$ are equivalent.

Finally, OM requires a $d \times d$ substitution-cost matrix $\mathbf{W} = \{w_{ij}\}$ wherein the w_{ij} specify the cost of substituting symbol σ_j for symbol σ_i . Interestingly, if we restrict \mathbf{W} such that

$$w_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \text{ and } i = 1 \text{ or } j = 1 \\ 2 & \text{if } i \neq j \text{ and } i, j > 1 \end{cases}, \quad (24)$$

the general OM-algorithm (e.g. Sankoff and Kruskal 1983; Clote and Backofen 2000) reduces to the Levenshtein (1966) algorithm. The latter algorithm is easily specified by first defining

$$L(x^i, y^j) = \min \{d_{\text{OM}}(x^{i-1}, y^j), d_{\text{OM}}(x^i, y^{j-1}), d_{\text{OM}}(x^{i-1}, y^{j-1})\} \quad (25)$$

and with which d_{OM} is dynamically computed through

$$d_{\text{OM}}(x^i, y^j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ L(x^i, y^j) & \text{if } i, j > 0 \text{ and } x_i = y_j \\ L(x^i, y^j) + 1 & \text{if } i, j > 0 \text{ and } x_i \neq y_j \end{cases}. \quad (26)$$

Now compare Sankoff’s algorithm (21) with Levenshtein’s (26): these algorithms are closely akin for they use the same 3 conditions on the prefixes and their elongations and both algorithms look for an extremal value in an $(m \times n)$ -array. Only the augmentation conditions have been reversed; the Sankoff-algorithm adds 1 if $x_i = y_j$ whereas the Levenshtein-algorithm adds 1 if $x_i \neq y_j$. Effectively, d_{OM} now counts the number of symbols not belonging to a common subsequence, i.e. the minimally required number of insertions of the gap-symbol, whereas Sankoff’s algorithm counts the number $A_{\mathcal{L}}$ of symbols of a longest common subsequence. Therefore, with \mathbf{W} as specified in (24), this implies that

$$d_{\text{OM}}(x, y) = |x| + |y| - 2A_{\mathcal{L}}(x, y) = d_{\mathcal{L}}(x, y). \quad (27)$$

TABLE 9. Family formation histories (left panel), their LLCS's (middle panel) and their numbers of distinct nonempty common subsequences (right panel).

	$A_{\mathcal{L}}$			$A_{\mathcal{S}}$		
$x = \text{U M MC}$	3			7		
$y = \text{S SC MC}$	1	3		1	7	
$z = \text{S M MC SC}$	2	2	4	3	5	15

This is remarkable since $d_{\mathcal{L}}$ derived from intuitive reasoning about similarity and subsequences whereas d_{OM} derived from reasoning about the energy required to align the sequences. Apparently, these reasonings touch. In bio-informatics, reasoning from minimal energy to change in a particular direction can be connected to electro-mechanical models about what is likely to happen to peptide-like substances. However, it is not easy to understand what demographical or sociological models could be connected to reasoning about minimal energy that changes social science categorical time series. For the change of a series U M MC into U S U cannot be observed since it simply does not happen. Therefore, coherent reasoning about the structure of \mathbf{W} will always pose a problem for social scientists. A potentially interesting alternative edit-distance was recently proposed by Bookstein, Kulyukin, and Raita (2002). In their proposal, the OM's substitution operation is replaced by a shift operation that allows to account for near-matches.

The reader notes that, with the cost-matrix as specified in (24), OM has a natural similarity measure: $s_{\mathcal{L}}$.

3.7. NCS: The Number of Distinct Nonempty Common Subsequences. Using LLCS to compare sequences in fact implies that we use the (relative) length of common backbone to determine distance or similarity. In Table 9, we show three examples of family formation histories and, in the middle panel, their LLCS's. Clearly, according to (17) and (18), x and y are equidistant to or equally similar to z since x and y are of the same length and both have an LCS of length 2 with z . However, this may be a disappointing result since y has three states in common with z whereas x and z have only 2 common states. Furthermore, in x , only one of the three pairs of states corresponds to a pair of states in z whereas in y , we find two pairs of states that occur in z too.

So, although in terms of LLCS, x and y are equidistant from z , there is more common order in the pair (y, z) than in the pair (x, z) . This is expressed in the right hand panel of Table 9 where we show the number $A_{\mathcal{S}}(x, y) = |\mathcal{S}(x, y)| - 1$ of distinct common nonempty subsequences: $A_{\mathcal{S}}(x, z) = 3 < A_{\mathcal{S}}(y, z) = 5$. If it is important to consider all common order instead of only the LLCS, the quantities

$$d_{\mathcal{S}}(x, y) = A_{\mathcal{S}}(x, x) + A_{\mathcal{S}}(y, y) - 2A_{\mathcal{S}}(x, y), \quad (28)$$

$$s_{\mathcal{S}}(x, y) = \frac{A_{\mathcal{S}}(x, y)}{\sqrt{A_{\mathcal{S}}(x, x) \cdot A_{\mathcal{S}}(y, y)}}, \quad (29)$$

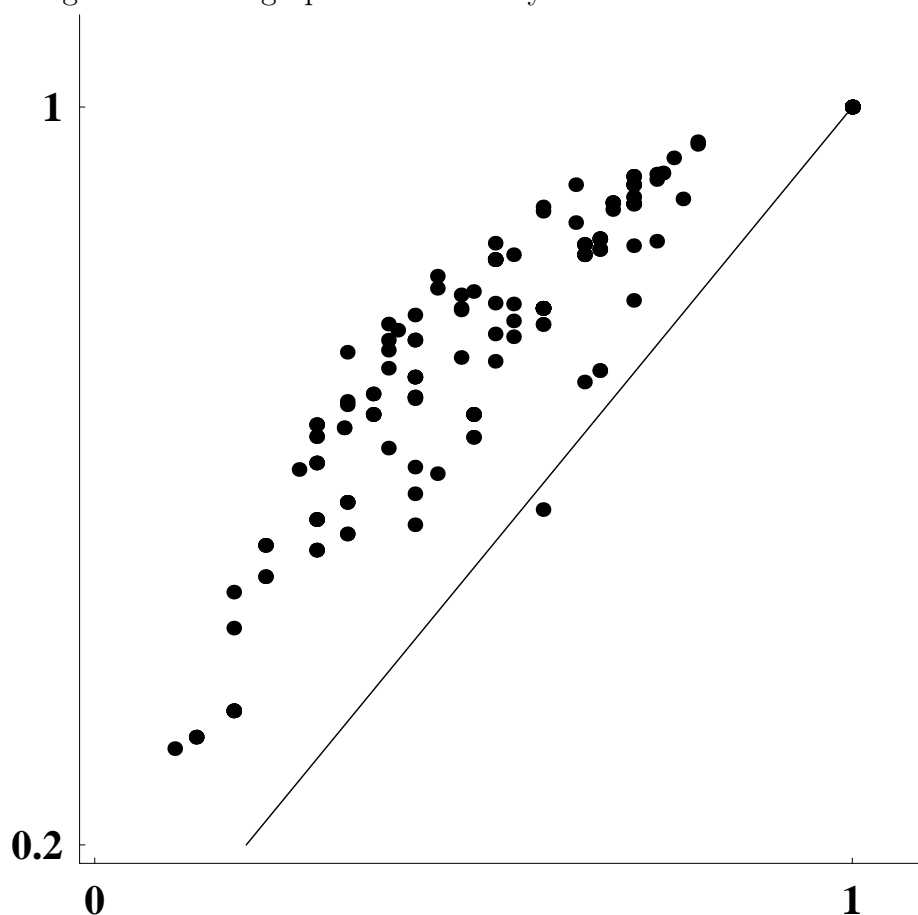
$$D_{\mathcal{S}}(x, y) = 1 - s_{\mathcal{S}}(x, y) \quad (30)$$

can be used to measure distance and similarity in the usual way.

But does it really matter? Does it really make a difference whether we use the $D_{\mathcal{L}}$ or $D_{\mathcal{S}}$? In

Figure 2, we show a plot of the 190 distances $D_{\mathcal{L}}$ and $D_{\mathcal{S}}$ as calculated for 20 histories of the family formation data. Apparently, it matters indeed. For Figure 2 shows that the metrics

FIGURE 2. Plot of the normalized distances $D_{\mathcal{L}}$ (horizontal axis) against the normalized distances $D_{\mathcal{S}}$ (vertical axis) as calculated from 20 family formation histories taken from the FFS data (durations ignored). Spearman's $\rho = 0.943$. The straight line is the graph of the identity function.



$D_{\mathcal{L}}$ and $D_{\mathcal{S}}$ are not monotone: the order of distances between many pairs of histories has been reversed by switching from the one metric to the other. Why do we see this picture? First, one should be aware of two simple facts:

$$\begin{aligned} D_{\mathcal{S}}(x, y) = 0 &\Leftrightarrow D_{\mathcal{L}}(x, y) = 0, \\ D_{\mathcal{S}}(x, y) = 1 &\Leftrightarrow D_{\mathcal{L}}(x, y) = 1. \end{aligned} \tag{31}$$

Therefore, Spearman's ρ will always be positive for such plots if these extremes occur in the data. Furthermore, if $A_{\mathcal{L}}(x, y)$ is small for a given pair of sequences, there cannot be very many common subsequences since all subsequences are subsequence of at least one longest common subsequence. Similarly, if $A_{\mathcal{L}}(x, y)$ is close to its attainable maximum, there cannot be very many non-common subsequences and thus we may not expect to see non-monotonicity of high amplitude. On the other hand, if $A_{\mathcal{L}}(x, y)$ is somewhere in between zero

and it's maximum, quite some variation in the number of common subsequences is possible. As can be seen in the graph, there is much scatter along lines perpendicular to the horizontal axis, i.e. variation in D_S at constant D_L . This is reflected in the fact that there are only 29 distinct values of D_L whereas we have 79 distinct values for D_S . Hence, another way of formulating the non-monotonicity of the metrics is to say that D_S provides for much more detailed information about $\mathcal{S}(x, y)$ than D_L does.

The straight line in the plot stresses the fact that for one pair, we found $D_L > D_S$: $x = \text{S M}$ and $y = \text{U M S}$. So, the fact that we found $D_S > D_L$ in most cases is a property of the data, not of the metrics.

As with the metric d_L , and given an alphabet Σ , the maximum of d_S depends upon the lengths of the sequences involved since the maximum of $A_S(x, x)$ depends upon $|x|$ and $|\Sigma|$ (Flaxman, Harrow, and Sorkin (2004) specified, given $|\Sigma|$ and $|x|$, an x that maximizes $A_S(x, x)$ and the maximum itself). Therefore, we presented the normalized version of d_S in (30).

Since d_S arises from cardinalities of sets, it is not difficult, writing $\mathcal{S}(x) = \mathcal{S}(x, x)$ for brevity, to see that

$$d_S(x, y) = |\{\mathcal{S}(x) - \mathcal{S}(y)\} \cup \{\mathcal{S}(y) - \mathcal{S}(x)\}|, \quad (32)$$

i.e. the cardinality of the symmetric set difference $\mathcal{S}(x) \Delta \mathcal{S}(y)$. Interestingly, Yianilos (2002) proved that, for arbitrary sets U and V , the quantity $|U \Delta V| / |U \cup V|$ is a metric. Because of (32), this implies that an alternative normalization of d_S is provided by

$$\begin{aligned} D_{T;S}(x, y) &= \frac{d_S(x, y)}{d_S(x, y) + A_S(x, y)} \\ &= 1 - T_S(x, y), \end{aligned} \quad (33)$$

i.e. a normalization on the basis of the Tanimoto-coefficient (4).

Now there seems to be only one problem left: the actual calculation of $A_S(x, y)$. This is not a trivial problem since if the shorter sequence has length n , there are $2^n - 1$ nonempty subsequences that may or may not be distinct and that may or may not be common. Elzinga (2007) provides for an algorithm that evaluates $A_S^+(x, y) = A_S(x, y) + 1$, i.e. size of $\mathcal{S}(x, y)$ including the empty string λ , in an amount of time that is proportional to the product mn , given $|x| = m$ and $|y| = n$. To present it here, we need a function that returns the position of the last occurrence of a specific character, say $b \in \Sigma$, in a sequence:

$$\ell(x, b) = \begin{cases} \max\{i : x_i = b\} & \text{if } b \in x \\ 0 & \text{if } b \notin x \end{cases} \quad (34)$$

Now let $|x| = m$ and set $\ell_y = \ell(y, x_m)$ and $\ell_x = \ell(x^{m-1}, x_m)$; i.e. ℓ_y equals the *last* position of x_m in y , whereas ℓ_x equals the *previous* position of x_m in x . With these definitions it can be shown that

$$A_S^+(x, y) = \begin{cases} A_S^+(x^{m-1}, y) & \text{if } x_m \notin y \\ A_S^+(x^{m-1}, y) + A_S^+(x^{m-1}, y^{\ell_y-1}) & \text{if } x_m \in y \text{ and } x_m \notin x^{m-1} \\ A_S^+(x^{m-1}, y) + A_S^+(x^{m-1}, y^{\ell_y-1}) & \\ -A_S^+(x^{\ell_x-1}, y^{\ell_y-1}) & \text{if } x_m \in y \text{ and } x_m \in x^{m-1} \end{cases} \quad (35)$$

TABLE 10. Template trajectories of family formation are shown in the left-most part; the middle part pertains to the birth cohort 1945-49 of Austrian women and the rightmost part pertains to the youngest 1960-64 cohort. In the middle and right part, the left column contains the percentage of the trajectories assigned to the templates on the basis of minimal D_S and the righthand columns show the average similarity s_S per template group. The averages of the similarities for the total cohorts are shown in the headers.

Template	1945-49		1960-64	
	$N = 539$	$\bar{s}_S = .49$	$N = 752$	$\bar{s}_S = .37$
	%	\bar{s}_S	%	\bar{s}_S
S M MC	53.6	.92	23.1	.88
S U M MC	10.4	.79	25.8	.82
S U UC	8.7	.33	17.4	.59
S U S U	1.7	.59	8.2	.43
S M MC SC	16.3	.58	17.6	.44
S	9.3	.84	7.8	.82

The recurrence in (35) implies a dynamic algorithm since we can initialize $A_S^+(x, y^0) = 1 = A_S^+(x^0, y)$ and then use the recurrence to iteratively process the $(m \times n)$ -array $\{x^i, y^j\}$ until we finally obtain $A_S^+(x^m, y^n) = A_S^+(x, y)$.

Since the basic structure of (35) will be encountered in some other algorithms too, it is useful to try to understand why the algorithm correctly counts A_S^+ . To see the correctness of (35), first observe that each line describes what happens to the quantity A_S^+ as a result of elongating x^{m-1} with x_m . Of course, A_S^+ will not change if $x_m \notin y$ since the elongation of x^{m-1} with x_m will not generate any new common subsequences. What will happen to A_S^+ if $x_m \in y$ depends upon whether or not x_m is new to x^{m-1} or not. If x_m is new to x^{m-1} , a new common subsequence can be constructed by taking an arbitrary common subsequence of (x^{m-1}, y^{ℓ_y-1}) and elongating it with x_m . Clearly, there must be $A_S^+(x^{m-1}, y^{\ell_y-1})$ of such common subsequences that can be elongated with x_m .

However, this procedure counts too much in case x_m is not new to x^{m-1} but already occurs at x^{ℓ_x} . For then some of the subsequences that are common to (x^{m-1}, y^{ℓ_y-1}) already end on x_m : those that were already common to $(x^{\ell_x-1}, y^{\ell_y-1})$. Therefore, to compensate and prevent doubly counting, we subtract the quantity $A_S^+(x^{\ell_x-1}, y^{\ell_y-1})$ in the third line of (35).

3.8. De-Standardization and Templates of Family Formation. To further investigate the de-standardization of family formation, Elzinga and Liefbroer (2007) constructed distinct classes or types of family formation trajectories through defining 7 template trajectories and assigning each observed trajectory to the template class with the smallest distance to the observed trajectory. As a distance measure, they used the metric that is explained in the next section and included durations. Here we use D_S and, because we ignore durations, we only use 6 of the proposed 7 templates; without durations, we cannot discern those that enter parenthood early from those that enter parenthood late. Here, we confine ourselves to the oldest and youngest cohorts of Austrian women. The results are shown in Table 10.

Clearly, there is a tendency to cohabit and enter parenthood before entering marriage at the cost of the traditional S M MC. It is remarkable that, even without using information on durations, D_S is capable of unveiling such developments in patterns of family formation.

3.9. NMS: The Number of Matching Subsequences. Consider the pair of labor market entry careers

$$\begin{aligned} x &= \text{E U} \\ y &= \text{E U T E T U E U E} \end{aligned}$$

Clearly, x and y share 3 distinct nonempty subsequences. Also, we observe that x is embedded 6 times in y but this fact is not accounted for in d_S . However, repeated embedding of subsequences mostly points to a socially and/or psychologically relevant fact but such repeats are not taken into account by d_S : with that metric, the sub-career EU is just one distinct common subsequence. So, in some applications it seems relevant to have a metric that does take the number of embeddings of common subsequences into account. In fact, such a metric was proposed by Elzinga (2005) and it is based upon the attribute

$$A_{\mathcal{M}}(x, y) = \sum_{u \in \mathcal{S}(x, y) \setminus \lambda} |x|_u \cdot |y|_u. \quad (36)$$

$A_{\mathcal{M}}$ counts, for each of the nonempty distinct common subsequences of the pair (x, y) , how often each embedding of a subsequence $u \in x$ can be matched by that same subsequence as embedded in y . Therefore, $A_{\mathcal{M}}$ is called the number of matching subsequences of (x, y) . It is not difficult to see that $A_{\mathcal{M}}$ does *not* satisfy the boundary conditions (1) but it still can be proven that

$$d_{\mathcal{M}}(x, y) = A_{\mathcal{M}}(x, x) + A_{\mathcal{M}}(y, y) - 2 \cdot A_{\mathcal{M}}(x, y) \quad (37)$$

is a metric and we can use $A_{\mathcal{M}}$ to construct a similarity measure according to (3). Elzinga (2003, 2005) proposed two algorithms to evaluate $A_{\mathcal{M}}$; we describe one of these. Given a pair of strings (x, y) with $|x| = m$ and $|y| = n$, define a matrix $\mathbf{E}^1 = \{e_{ij}^1\}$ with

$$e_{ij}^1 = \begin{cases} 1 & \text{if } x_i = y_j \\ 0 & \text{if } x_i \neq y_j \end{cases} \quad (38)$$

and recursively calculate, for $1 < k \leq \min\{m, n\} = M$, the matrices $\mathbf{E}^k = \{e_{ij}^k\}$ with

$$e_{ij}^k = \sum_{a>i, b>j} e_{ab}^{k-1}. \quad (39)$$

Then

$$A_{\mathcal{M}}(x, y) = \sum_{k=1}^M S_k \quad (40)$$

with $S_k = \sum_{ij} e_{ij}^k$ for $1 \leq k \leq M$ and this algorithm will evaluate $A_{\mathcal{M}}$ in an amount of time that is proportional to the product Mmn .

Why is this algorithm correct? To explain this, we set $x = abac$ and $y = babc$ and we easily calculate

$$\mathbf{E}^1 = \begin{pmatrix} 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{pmatrix}, \quad \mathbf{E}^2 = \begin{pmatrix} 0 & \mathbf{2} & 0 & 0 \\ \mathbf{2} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{0} \end{pmatrix}.$$

In the diagrams below, we have bullets representing the positive elements of \mathbf{E}^1 as nodes in an (invisible) lattice. In the left diagram, we have one directed path, connecting each node with itself. These 1-paths represent the matching 1-tuples or 1-long matching subsequences of x and y . In the second diagram, we connected each node with all other nodes that are “South-East” of that node. Now \mathbf{E}^2 contains the number of outgoing 2-paths for each node and each 2-path represents a matching 2-tuple or 2-long matching subsequence of x and y . So, the S_k in (40) just contain the sums of the numbers of outgoing k -paths from each of the nodes of the lattice.



3.10. Distance to the unsuccessful labor market entry. On the basis of OM-distances, McVicar and Anyadike-Danes (2002) constructed 5 clusters of labor market entry trajectories and then tried to predict cluster-membership from a number of binary covariates through a logit model. Unfortunately, their results cannot be replicated (Anyadike-Danes, 2005). Here we use the same binary variables to predict, through a standard OLS regression model, the distance to an “unsuccessfulness-template”, the sequence T/20 U/52. In Table 11, we show results for three different metrics. Interestingly, all three metrics pick up the same indicators as contributing significantly, with the exception of being Catholic. According to the “matching subsequences”-metric, being a Catholic (in Northern-Ireland in 1993-1999) negatively contributes to the distance to the template, i.e. is not favoring successful integration into the labor market. The other four relevant indicators are picked up equally by all three metrics. The reader notes that the R^2 -values should not be compared with each other since they refer to proportions of different variances. Of course, we are aware that these results are probably somewhat sensitive to the template and the OM-cost chosen and that we took no measures (e.g. a logistic transformation) to prevent predicted distances to lie outside the $[0,1]$ -interval. However, we feel that the results demonstrate that, on the one hand, there are relevant differences between the metrics and that, on the other hand, the metrics roughly lead to the same, credible conclusions about which factors contribute to successful integration.

TABLE 11. Standardized coefficients in OLS-regression of the distance to the template T/20 U/52 as dependent variable. Three metrics were used: D_S^* and D_M^* include durations according to the MST-principle (see section on durations handling) and the OM-distances were calculated with insert- and delete-cost set to 1 and the cost of substitution set to 2. Boldfaced figures denote that the coefficients are significantly different from 0 ($p < .05$). Belfast, N-East, South and S-East refer to living in particular regions; region West was left out because of redundancy. $\#(\text{GCSE}) > 5$ denotes that the subject has at least obtained 5 General Certificates of Secondary Education which were at least C-graded. Father's occupation =1 if it is of professional or managerial level.

Binary independents (=1)	D_S^*	D_M^*	D_{OM}
Male	-.028	-.018	-.042
Catholic	-.065	-.085	-.066
Belfast	.068	.067	.001
N-East	.004	.020	.056
South	.072	.077	.063
S-East	.100	.113	.148
Finished Grammar	.094	.094	.145
Father unemployed	-.096	-.114	-.112
$\#(\text{GCSE}) > 5$.252	.256	.293
Father's occupation	.030	.027	.035
Live with both parents	.013	.007	.035
R^2	.122	.136	.211

3.11. Weighing the Subsequences. So far, we have seen a few ways of using information on subsequences to define attributes and construct metrics on Σ^* . Of course, there is no limitation to the number and kinds of attributes that could be sensibly defined. However, applying such attributes to actually construct metrics may fail because of the algorithmic complexities of numerically evaluating the attributes defined. One obvious way to generalize the attributes A_S and A_M is to weigh according to the length of the subsequences counted: the longer the common subsequences, the more similar or less distant the strings. This simply amounts to defining the generalized attributes

$$A_{wS}(x, y) = \sum_{u \in \mathcal{S}(x, y) \setminus \lambda} f(|u|) \quad (41)$$

and

$$A_{wM}(x, y) = \sum_{u \in \mathcal{S}(x, y) \setminus \lambda} f(|u|) \cdot |x|_u \cdot |y|_u \quad (42)$$

wherein $f(\cdot)$ is any non-decreasing function on the integers. Choosing $f(k) = k^p$ for a suitably chosen non-negative p will mostly do the job. Practically evaluating $A_{wM}(x, y)$ is easily accomplished by a slight adaptation of the final step (40) of the algorithm that

evaluates $A_{\mathcal{M}}$: we simply change it into

$$A_{w\mathcal{M}}(x, y) = \sum_{k=1}^M f(k) \cdot S_k. \quad (43)$$

However, an equally simple adaptation of the algorithm (35) is not possible: (35) does not discriminate between subsequences of different lengths. So, we need a different algorithm to evaluate the attribute (41) which we discuss here. Let $A_k(x, y)$ denote the number of distinct common k -long subsequences of the pair (x, y) with $|x| = m$, $|y| = n$ and $M = \min\{n, m\}$, i.e.

$$A_k(x, y) = |\{u \in \mathcal{S}(x, y) \setminus \lambda : |u| = k\}|, \quad (44)$$

then

$$A_{w\mathcal{S}}(x, y) = \sum_{k=0}^M f(k) \cdot A_k(x, y). \quad (45)$$

So, it suffices to have an algorithm to numerically evaluate the quantity $A_k(x, y)$. Such an algorithm is implied by the recurrence

$$A_k(x, y) = \begin{cases} A_k(x^{m-1}, y) & \text{if } x_m \notin y \\ A_k(x^{m-1}, y) + A_{k-1}(x^{m-1}, y^{\ell_y-1}) & \text{if } x_m \in y \text{ and } x_m \notin x^{m-1} \\ A_k(x^{m-1}, y) + A_{k-1}(x^{m-1}, y^{\ell_y-1}) \\ \quad - A_{k-1}(x^{\ell_x-1}, y^{\ell_y-1}) & \text{if } x_m \in y \text{ and } x_m \in x^{m-1} \end{cases}. \quad (46)$$

To initialize (46), we set $A_0(x, y) = 1$ and, for $k \geq 1$, we take $A_k(x^i, y^j) = 0$ for $i, j < k$. (46) will take an amount of time that is proportional to the product kmn since for each $1 \leq j \leq k$, we have to evaluate the array A_{j-1} in order to calculate the array A_j .

Evidently, the structures of the algorithms (35) and (46) are identical. Indeed, understanding that (46) is correct, requires reasoning that is almost identical to the argument employed in justifying (35). The only difference is, that we now elongate common subsequences in an ‘‘orderly’’ manner: we confine ourselves to only elongating the $(k-1)$ -tuples to k -tuples and we are punished for this orderliness by having to repeat it k times.

3.12. String-representing vectors. In the previous sections, we have been discussing distances and angles between strings as if these strings have a location in space. However, we never discussed these locations, i.e. we did not discuss how strings can be represented by vectors. In this section, we elaborate and generalize on the way we constructed vectors in the introductory section. Thereto, we start generalizing the ‘‘trick’’ that we used to construct the vectors. This trick consisted of first ranking, i.e. numerically labeling, the characters of the alphabet - $\Sigma = \{E, U, T, S, F, H\} \rightarrow \{1, 2, 3, 4, 5, 6\}$ - and then we used these ranks to point at coordinates of vectors $\mathbf{x} = (\dot{x}_1, \dots, \dot{x}_6)$ through the rule

$$\dot{x}_{r(\sigma)} = \begin{cases} 1 & \text{if } \sigma \in x \\ 0 & \text{if } \sigma \notin x \end{cases}. \quad (47)$$

In this way we obtained $\mathbf{x} = (0, 1, 1, 1, 0, 0)$ representing $x = \text{S U T}$.

Here we generalize this trick of rank numbering by ranking Σ^* instead of Σ . This implies that we numerically label all subsequences u with a nonnegative integer $r(u)$. For example, with $\Sigma = \{a, b, c\}$ and lexicographic ordering, we obtain $r(b) = 2$, $r(ac) = 6$, $r(cab) = 32$ and $r(abac) = 51$. Let us now put this labeling to work for the prefix-based attribute. Thereto, we define vectors with coordinates that are positive only if the pertaining subsequence is a prefix:

$$\dot{x}_{r(u)} = \begin{cases} 1 & \text{if } u \in \mathcal{P}(x, x) \\ 0 & \text{otherwise} \end{cases}. \quad (48)$$

Clearly, since $A_{\mathcal{P}} = |\mathcal{P}(x, y)|$, we have that $A_{\mathcal{P}} = \mathbf{x} \cdot \mathbf{y} = \sum_i \dot{x}_i \dot{y}_i$ with \mathbf{x} and \mathbf{y} constructed according to the above rule.

Since we can easily change a rule like (48), the labeling offers a lot of freedom to construct vectors in different ways. To construct the representing vectors that are associated to the attribute $A_{\mathcal{S}}$, we switch to

$$\dot{x}_{r(u)} = \begin{cases} 1 & \text{if } u \in \mathcal{S}(x, x) \setminus \lambda \\ 0 & \text{otherwise} \end{cases}. \quad (49)$$

which yields

$$\mathbf{x} \cdot \mathbf{y} = A_{\mathcal{S}}(x, y) = \sum_{u \in \mathcal{S}(x, y) \setminus \lambda} 1 \cdot 1. \quad (50)$$

A straightforward modification of the above rule (49) yields the vector that is associated to the number of matching subsequences $A_{\mathcal{M}}$:

$$\dot{x}_{r(u)} = \begin{cases} |u|_x & \text{if } u \in \mathcal{S}(x, x) \setminus \lambda \\ 0 & \text{otherwise} \end{cases}. \quad (51)$$

Constructing the vectors associated with LLCs's is not difficult either; only it requires a small detour. We set

$$\dot{x}_{r(u)} = \begin{cases} \sqrt{|u|} & \text{if } u \in \mathcal{S}(x, x) \setminus \lambda \\ 0 & \text{otherwise} \end{cases}. \quad (52)$$

So, if u is common to x and y , we have $\dot{x}_i \cdot \dot{y}_i = |u|$ and if u is not common, $\dot{x}_i \cdot \dot{y}_i = 0$ and therefore, the above rule yields vectors of which the inner product is associated with a special case of the weighted attribute (41) wherein we set $f(|u|) = |u|$. Because of (52), we now have

$$\mathbf{x} \cdot \mathbf{y} = \sum_i \dot{x}_i \dot{y}_i \quad (53)$$

$$= \sum_{u \in \mathcal{S}(x, y) \setminus \lambda} |u| = A_{w\mathcal{S}}(x, y). \quad (54)$$

If we now replace operator \sum_i in (53) by the operator \max_i , we are back at $A_{\mathcal{L}}$:

$$A_{\mathcal{L}}(x, y) = \max_i \{\dot{x}_i \cdot \dot{y}_i\} \quad (55)$$

and we see that $d_{\mathcal{L}}$ is closely akin to the metric $d_{w\mathcal{S}}$.

Once we are aware that, given alphabet size $|\Sigma| = d$, the number of coordinates that is required to represent all k -long subsequences equals d^k , it is not difficult to see that $d_{w\mathcal{S}}$ implies a linear transform of the space associated with $d_{\mathcal{S}}$ since

$$A_{w\mathcal{S}}(x, y) = \sum_{u \in \mathcal{S}(x, y) \setminus \lambda} f(|u|) = \sum_i b_{ii} \cdot x_i \cdot y_i = \mathbf{B}^{1/2} \mathbf{x} \cdot \mathbf{B}^{1/2} \mathbf{y} \quad (56)$$

wherein $\mathbf{B} = \{b_{ij}\}$ is a diagonal matrix with

$$b_{ii} = f(j) \quad \text{if} \quad \sum_{k=1}^{j-1} d^k + 1 \leq i \leq \sum_{k=1}^j d^k. \quad (57)$$

clearly, the latter condition holds since a weight is a constant for all coordinates that represent subsequences of equal length.

We now also understand that really labeling Σ^* is impossible since it would require a 1-1 map to the nonnegative integers \mathbb{N} . Now one might argue that we do not need vectors of countably infinite dimension since in practice, we will never deal with sequences of such a colossal length that this would be required. But, given $|\Sigma| = d$ and a sequence of length n , we would need vectors that represent all subsequences up to and including length n : vectors with $\sum_{i=1}^n d^i = (d^{n+1} - d) / (d - 1)$ coordinates. With $d = 6$ and $n = 40$, we would need an integer of more than 30 digits to count the coordinates. So, the algorithms that we discussed in previous sections are necessary because we cannot afford the use of the much simpler vector arithmetic; it simply requires too much memory and processing capacity.

4. HANDLING DURATION

Most often, social scientists not only gather categorical time series of the form $x = x_1 \cdots x_n$ but also information on the durations $\mathbf{t}_x = (t_x(1), \cdots, t_x(n))$. So, mostly, our data come in the form of pairs (x, \mathbf{t}_x) and then we are interested in a representation of such pairs or in their distances $d^*(x, y) = d((x, \mathbf{t}_x), (y, \mathbf{t}_y))$.

Basically, durations can be handled in two, quite different ways.

The first, most simple approach is to construct strings that have one state for each and every unit of time covered by the observation period. In the example of the labor market entry careers, this implies that the sequences consist of 72 states, one for each month of the 6-year period covered. Interestingly, these 72-long sequences do not directly reflect the structure of the observations from which they were constructed; the observations are aggregated reconstructions from two face-to-face interviews of two sweeps of the panel, one in 1995 and one in 1999 (McVicar and Anyadike-Danes, 2002, pp. 318). The advantages of this construction are obvious: Durations have vanished since the information has been transformed to a pure string of states. Consequently, any of the metrics discussed can be directly applied since these metrics apply to strings in which the states have no quantified properties. Durations are now inferred from the data. From the string $x = \dots baaaaac \dots$ we infer that there was a spell of state a that lasted for four units of time. The problem is that, in many cases, these are not the sequences that we are trying to discuss. Often we are trying to analyze or classify sequences, trajectories of states, that may show long spells of living cohabitated or periods of unemployment of only short duration. What we have in

mind then are trajectories of states that have the property of duration like e.g in the labor market entry career S/2 T/8 U/15 E/47. For consider the two family formation histories

$$\begin{aligned} & S/14 \ M/12 \ MC/128, \\ & S/12 \ M/18 \ S/12 \ M/8 \ MC/86. \end{aligned}$$

Written as pure strings, we get

$$\begin{array}{c} \underbrace{S \dots S}_{14} \underbrace{M \dots M}_{12} \underbrace{MC \dots MC}_{118}, \\ \underbrace{S \dots S}_{12} \underbrace{M \dots M}_{18} \underbrace{S \dots S}_{20} \underbrace{M \dots M}_{8} \underbrace{MC \dots MC}_{86}. \end{array}$$

Indeed, we now correctly determine the LCP to have a length of 12 and the LCS to have a length of 116. However, it is probably much more sensible to use the (x, \mathbf{t}_x) -format and conclude that the LCP has length 2 and that an LCS, the common backbone, has length 3. The second, far less simple approach, is to take the data in (x, \mathbf{t}_x) -format and try to formulate attributes that incorporate the durations. This is difficult since we have to answer the question of how to evaluate commonness of duration.

The attributes that we have dealt with, evaluate the length of a common subsequence ($A_{\mathcal{P}}$ and $A_{\mathcal{L}}$) or they count a number of common subsequences ($A_{\mathcal{S}}$ and $A_{\mathcal{M}}$). Adapting these attributes to incorporate durations requires that we somehow weigh the length or number of subsequences by their “common duration”. But then we must define the common duration $t(u)$ of a common subsequence u when the duration $t_x(u)$ of u in x is different from the duration $t_y(u)$ of u in y .

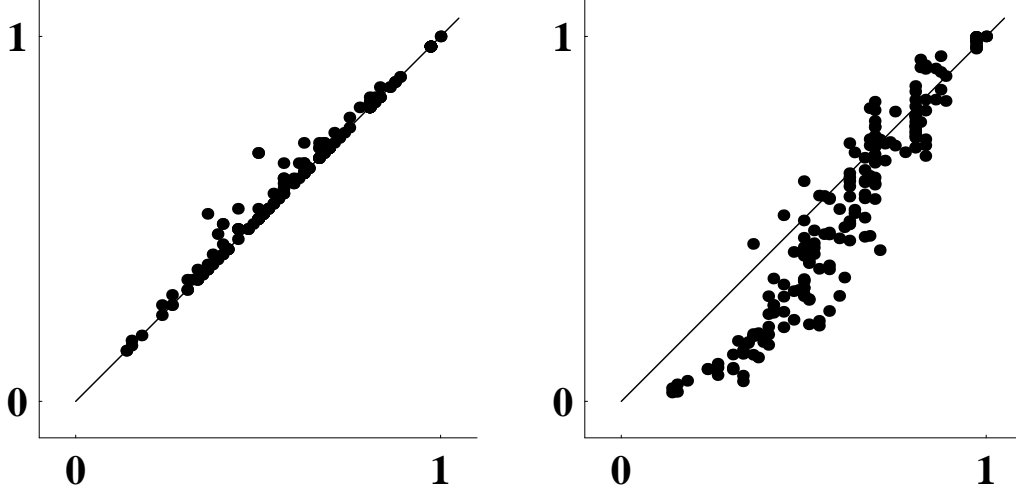
Probably, we agree that $t(u) = f(t_x(u), t_y(u))$ where f is a function that is non-decreasing in either argument. But there is an abundance of such functions; weighed sums and products and maxima and minima and what not. In the next two subsections, we will be concerned with two specific choices of such a function. The first choice tries to account for that part of the duration that is equal in both sequences. For example, suppose that $x_1x_3x_7 = u = y_2y_4y_5$ is a common subsequence of x and y and that

$$t_x(1) = 2, t_x(2) = 8, t_x(3) = 4,$$

$$t_y(2) = 3, t_y(4) = 6, t_y(5) = 5.$$

Then these observations imply that these individuals share having spent at least 2 units of time in state $x_1 = y_2$, share having spent at least 6 units of time in state $x_3 = y_4$, etc. So, it is implied that these individuals shared spending at least $12=2+6+4$ units of time in the common subsequence u : the “common duration”, constructed according to what we call the “Principle of Minimal Shared Time”. The next subsection investigates how we can use this principle to adapt the metrics and the algorithms. In the other subsection, we define common duration as the sum of products of state durations. So, instead of $\min\{2, 3\} + \min\{8, 6\} + \min\{4, 5\} = 12$ units of time, we will then take $2 \cdot 3 + 8 \cdot 6 + 4 \cdot 5 = 74$ squared units of time as the “common duration”, i.e. common duration as a vector product. The most important difference between handling time according to the MST-principle and handling duration as a vector-product is that in the the MST-principle is totally insensitive to the longer of the two durations compared whereas in the vector-product, longer durations “weigh” shorter durations. From the plots in Figure 3, we conclude that choosing a different way of handling

FIGURE 3. Plot of $D_{\mathcal{L}}^{\circ}$ (full-string format, horizontal axis) against $D_{\mathcal{L}}^*$ for distances between 20 labor market entry careers in the left panel. The right panel shows the plot of $D_{\mathcal{L}}^*$ (horizontal axis) against $D_{\mathcal{L}}^{\diamond}$ for the same careers. The straight lines represent the identity function



durations will change the order of the distances; the extend of this change will depend on the data.

4.1. Applying the Principle of Minimal Shared Time. Instead of using the length of the longest common prefix, we might define a new attribute: the minimum amount of shared time spent in the longest common prefix. Formally, this amounts to defining an attribute $A_{\mathcal{P}}^*(x, y)$ as

$$A_{\mathcal{P}}^*(x, y) = \sum_{i=1}^k \min \{t_x(i), t_y(i)\} \quad (58)$$

with $k = \max \{j : x^j = y^j\}$. As $A_{\mathcal{P}}^*(x, y)$ clearly satisfies the boundary conditions (1), we can define the metric $d_{\mathcal{P}}^*$ and an associated similarity measure $s_{\mathcal{P}}^*$ in the usual way.

Let us now try to find out how we can use the concept of “minimal shared time” (MST) to construct an attribute $A_{\mathcal{L}}^*$ that is analogous to $A_{\mathcal{L}}$. The latter attribute measures the *length* of a *longest* common subsequence. Clearly then, $A_{\mathcal{L}}^*$ should measure the maximum MST spent in the common subsequences. However, the MST of a common subsequence may not be well-defined since a subsequence may have several different embeddings in one and the same string. For example, $u = ab$ occurs three times in $x = abab$: as x_1x_2 , as x_1x_4 and as x_3x_4 . Therefore, let us first be more precise about embeddings. An embedding is nothing but an ordered set of indices that indicate where the symbols of a subsequence occur in a string: (1, 2), (1, 4) and (3, 4) are embeddings of $u \in x$. Formally, let u be a k -long subsequence of a string x with $u_1 \cdots u_k = x_{i_1} \cdots x_{i_k}$ then the ordered set of indices $i_x(u) = i_1, \dots, i_k$ is called an embedding of u in x .

Now let us define the set $\mathcal{I}(x, y)$ as the set of all pairs of embeddings of subsequences common

to x and y :

$$\mathcal{I}(x, y) = \{(i_x(u), i_y(u)) : u \in \mathcal{S}(x, y)\}. \quad (59)$$

As we want the attribute $A_{\mathcal{L}}^*$ to maximize MST, we use this set to define

$$A_{\mathcal{L}}^*(x, y) = \max_{(i_x(u), i_y(u)) \in \mathcal{I}(x, y)} \left\{ \sum_{i=1}^{|u|} \min \{t_x(i_i), t_y(j_i)\} \right\} \quad (60)$$

with the understanding that $i_i \in i_x(u)$ and $j_i \in i_y(u)$. The reader has understood that the subsequence u in the above expression is not necessarily a *longest* common subsequence. Although the formal definition of the above attribute is somewhat complicated, it's numerical evaluation is easy with a modification of Sankoff's algorithm: we change (21) into

$$A_{\mathcal{L}}^*(x^i, y^j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ H = \max\{A_{\mathcal{L}}^*(x^{i-1}, y^j), A_{\mathcal{L}}^*(x^i, y^{j-1})\} & \text{if } i, j > 0 \text{ and } x_i \neq y_j \\ \max\{H, A_{\mathcal{L}}^*(x^{i-1}, y^{j-1}) + \min\{t_x(i), t_y(j)\}\} & \text{if } i, j > 0 \text{ and } x_i = y_j \end{cases} \quad (61)$$

Evidently, $A_{\mathcal{L}}^*$ satisfies the boundary conditions (1) so the quantity

$$d_{\mathcal{L}}^*(x, y) = \sum_{i=1}^{|x|} t_x(i) + \sum_{i=1}^{|y|} t_y(i) - 2 \cdot A_{\mathcal{L}}^*(x, y) \quad (62)$$

is a metric. Because of previous remarks OM, it is clear that this metric is an alternative way of handling time in an OM-metric with the substitution cost as specified in (24).

The attribute $A_{\mathcal{M}}$ equals the sum $\sum_u |I_x(u)| \cdot |I_y(u)|$. The analogous attribute $\phi_{\mathcal{M}}^*$ will, for every pair $(I_x(u), I_y(u))$, add the minimal shared times spent in these pairs of embeddings. Therefore, analogous to $A_{\mathcal{M}}(x, y)$, we define the attribute

$$A_{\mathcal{M}}^*(x, y) = \sum_{u \in \mathcal{S}(x, y)} \sum_{i_x(u) \in I_x(u)} \sum_{i_y(u) \in I_y(u)} \left\{ \sum_{i \in i_x(u)} \sum_{j \in i_y(u)} \min \{t_x(i), t_y(j)\} \right\} \quad (63)$$

and it is not difficult to prove that the quantity

$$d_{\mathcal{M}}^*(x, y) = A_{\mathcal{M}}^*(x, x) + A_{\mathcal{M}}^*(y, y) - 2 \cdot A_{\mathcal{M}}^*(x, y) \quad (64)$$

is a metric. Evaluating $A_{\mathcal{M}}^*(x, y)$ requires an extension of the lattice algorithm shown above: we additionally define matrices $\mathbf{T}^k = \{t_{ij}^k\}$ with

$$t_{ij}^1 = e_{ij}^1 \cdot \min\{t_x(i), t_y(j)\}, \quad (65)$$

and calculate, for $1 < k \leq M$,

$$t_{ij}^k = e_{ij}^1 \cdot \left(e_{ij}^k \cdot t_{ij}^1 + \sum_{a>i, b>j} t_{ab}^{k-1} \right). \quad (66)$$

Next, we evaluate, for $1 \leq k \leq M$, $S_k^* = \sum_{ij} t_{ij}^k$ and finally

$$A_{\mathcal{M}}^*(x, y) = \sum_k S_k^*. \quad (67)$$

To demonstrate the logic of the above algorithm, we consider the pair $(x = abac, \mathbf{t}_x = (3, 2, 4, 6))$ and $(y = abc, \mathbf{t}_y = (1, 3, 5, 7))$. The first line (65) changes the lattice with the 1-paths into a lattice with paths that, as shown below, are weighed by the result of $\min\{t_x(i), t_y(j)\}$.



The second line (66) counts how often $(e_{i,j}^k)$ paths should be elongated with a weighed path $t_{i,j}^1$. The righthand diagram shows the result for the 2-paths.

What remains now is to find a way to use the MST-principle to create an analogue to the attribute A_S that counts the number of distinct subsequences $|\mathcal{S}(x, y)|$. This is not so easy since each particular common subsequence may have more than one distinct embedding in either string; so minimum shared time in any distinct subsequence may not be well-defined and maximizing this amount over all embeddings is not an option that leads to a viable algorithm. Therefore, we must turn to a less elegant solution: we estimate or approximate minimal shared time in any subsequence by using an average duration of the states. Using an average state-duration veils the effect of different embeddings: we set

$$\bar{t}_x(i) = \sum_{x_j=x_i} t_x(j) / |x|_{x_i} \tag{68}$$

and then define

$$A_S^*(x, y) = \sum_{u \in \mathcal{S}(x, y)} \sum_{i=1}^{|u|} \min \{ \bar{t}_x(i_i), \bar{t}_y(j_i) \} \tag{69}$$

in which $i_i \in i_x(u)$ and $j_i \in i_y(u)$. Again, we have that A_S^* satisfies the boundary conditions (1) so we can construct a metric d_S^* and its associated similarity measure s_S^* in the usual way. A simple extension of the algorithm (35) evaluates A_S^* numerically: we define $\zeta(i, j) =$

$\min \{\bar{t}_x(i), \bar{t}_y(j)\}$ and use this quantity in

$$A_{\mathcal{S}}^*(x, y) = \begin{cases} A_{\mathcal{S}}^*(x^{m-1}, y) & \text{if } x_m \notin y \\ A_{\mathcal{S}}^*(x^{m-1}, y) + A_{\mathcal{S}}^*(x^{m-1}, y^{\ell_y-1}) \\ \quad + A_{\mathcal{S}}(x^{m-1}, y^{\ell_y-1}) \cdot \zeta(m, \ell_y) & \text{if } x_m \in y \text{ and } x_m \notin x^{m-1} \\ A_{\mathcal{S}}^*(x^{m-1}, y) + A_{\mathcal{S}}^*(x^{m-1}, y^{\ell_y-1}) \\ \quad + A_{\mathcal{S}}(x^{m-1}, y^{\ell_y-1}) \cdot \zeta(m, \ell_y) \\ \quad - A_{\mathcal{S}}^*(x^{\ell_x}, y^{\ell_y}) \\ \quad + A_{\mathcal{S}}^*(x^{\ell_x-1}, y^{\ell_y-1}) & \text{if } x_m \in y \text{ and } x_m \in x^{m-1} \end{cases} \quad (70)$$

Of course, we initialize (70) with $A_{\mathcal{S}}^*(x^0, y) = 0 = A_{\mathcal{S}}^*(x, y^0)$. The logic behind (70) is not difficult to see: the second line follows from the fact that if x^{m-1} is elongated by the “new” x_m , $A_{\mathcal{S}}(x^{m-1}, y^{\ell_y-1})$ “new” common subsequences arise that end on x_m . These new subsequences increase the total minimum amount of shared time with $A_{\mathcal{S}}^*(x^{m-1}, y^{\ell_y-1}) + A_{\mathcal{S}}(x^{m-1}, y^{\ell_y-1}) \cdot \zeta(m, \ell_y)$ units of time. The subtraction in the third line compensates for the fact that the symbol x_m was encountered previously at x_{ℓ_x} .

These remarks conclude our illustrations of how the MST-principle can be used to construct attributes that incorporate duration. Because of lack of space, we do not dwell upon creating analogues to the “weighed” attributes from (41) and (42); this is left as a challenge to the interested reader.

4.2. Durations as vector products. We start modifying the representation (58) by defining the vectors $\mathbf{x} = (\dot{x}_1, \dot{x}_2, \dots)$ with coordinates

$$\dot{x}_{r(u)} = \begin{cases} t_x(|u|) & \text{if } u \in \mathcal{P}(x, x) \\ 0 & \text{otherwise} \end{cases}, \quad (71)$$

which ensures that the inner product has the form

$$A_{\mathcal{P}}^{\diamond}(x, y) = \sum_{i=1}^k t_x(i) \cdot t_y(i) = \mathbf{x} \cdot \mathbf{y} \quad (72)$$

with $k = \max\{i : x^i = y^i\}$. So, only the coordinates that pertain to prefixes are non-zero and their numerical value equals the duration of the *last* state of the prefix. Although $A_{\mathcal{P}}^{\diamond}$ does not satisfy the boundary conditions (1), $d_{\mathcal{P}}^{\diamond}$ is a squared Euclidean distance if constructed in the usual way.

We leave the definition of an LCP-based attribute $A_{\mathcal{L}}^{\diamond}$ to the reader; creating an algorithm to evaluate it amounts to an almost trivial modification in the last line of (61). In the section on the geometrical interpretation of some metrics, we explained that the attribute $A_{\mathcal{S}}(x, y)$, i.e. the number of distinct common subsequences, can be considered as an inner product of two binary-valued vectors \mathbf{x} and \mathbf{y} , the coordinates of which are defined by $\dot{x}_{r(u)} = 1$ if $u \in x$

and $\dot{x}_{r(u)} = 0$ in all other cases. It is now a straightforward matter to weigh the positive coordinates of the x -representing vector \mathbf{x} by the time spent in the subsequences. Only, as was explained in the previous subsection, the time spent in a particular subsequence may not be well-defined, due to the existence of different embeddings of u in x . So, we turn to average state-durations and define vectors $\mathbf{x} = (\dot{x}_1, \dot{x}_2, \dots)$ with coordinates

$$\dot{x}_{r(u)} = \begin{cases} \sum_{i \in i_x(u)} \bar{t}_x(i) & \text{if } u \in x \\ 0 & \text{if } u \notin x \end{cases} \quad (73)$$

wherein the $\bar{t}_x(i)$ are defined as in (68). This weighing implies an attribute of the form

$$\begin{aligned} A_{\mathcal{S}}^{\diamond}(x, y) &= \sum_{u \in \mathcal{S}(x, y)} \left(\sum_{i=1}^{|u|} \bar{t}_x(i) \right) \cdot \left(\sum_{i=1}^{|u|} \bar{t}_y(j_i) \right) \\ &= \sum_{u \in \mathcal{S}(x, y)} \dot{x}_{r(u)} \cdot \dot{y}_{r(u)} = \mathbf{x} \cdot \mathbf{y} \end{aligned} \quad (74)$$

i.e. a count of the common subsequences wherein each distinct subsequence is weighed by a product of estimated times spent.

Evaluating $A_{\mathcal{S}}^{\diamond}$ is not so trivial. From the first line of (74), we note that $A_{\mathcal{S}}^{\diamond}$ is build as a sum of products of sums. Therefore, we start looking at two arbitrary series $\{f_i\}$ and $\{g_i\}$ and their associated series of sums $\{F_m = \sum_{i=1}^m f_i\}$ and $\{G_n = \sum_{i=1}^n g_i\}$. The product of these sums can be written as

$$F_m \cdot G_n = F_{m-1} \cdot G_{n-1} + f_m \cdot G_{n-1} + g_n \cdot F_{m-1} + f_m \cdot g_n. \quad (75)$$

So, the product $F_m \cdot G_n$ can be recursively calculated from F_{m-1} and G_{n-1} and the ‘‘elongations’’ f_m and g_n . We use this fact and therefore start defining the sum $S_x(x, y) = \sum_{u \in (x, y)} x_{r(u)}$, i.e. the sum of all positive coordinates of \mathbf{x} that are positive in \mathbf{y} too. A similar sum $S_y(x, y)$ is defined analogously.

$S_x(x, y)$ consists of two parts: $S_x(x^{n-1}, y)$ and the sum of coordinates that only become positive after elongating x^{n-1} to $x^n = x$. Let us, for the moment, presume that $x_n \notin x^{n-1}$. But then the latter are all coordinates summed in $S_x(x^{n-1}, y)$, augmented by $\bar{t}_x(x_n)$, and there are $A_{\mathcal{S}}(x^{n-1}, y)$ of this kind. So, when $x_n \notin x^{n-1}$, we must have that

$$\begin{aligned} S_x(x, y) &= S_x(x^{n-1}, y) + (S_x(x^{n-1}, y) + A_{\mathcal{S}}(x^{n-1}, y) \cdot \bar{t}_x(x_n)) \\ &= S_x(x^{n-1}, y) + U_x(x^{n-1}, y). \end{aligned} \quad (76)$$

Hence, the quantity $U_x(x^{n-1}, y)$ denotes the difference between $S_x(x, y)$ and its ‘‘predecessor’’ $S_x(x^{n-1}, y)$. Generalizing, we must have that

$$S_x(x, y) = \begin{cases} S_x(x^{n-1}, y) & \text{if } x_n \notin y, \\ S_x(x^{n-1}, y) + U_x(x^{n-1}, y) & \text{if } x_n \in y \text{ and } x_n \notin x^{n-1}, \\ S_x(x^{n-1}, y) + U_x(x^{n-1}, y) - S_x(x^{\ell_x}, y^{\ell_y}) + S_x(x^{\ell_x-1}, y^{\ell_y-1}) & \text{if } x_n \in y \text{ and } x_n \in x^{n-1}. \end{cases} \quad (77)$$

Now we can use the rule on sums of products (75) and define the quantity

$$\begin{aligned}
 U_{xy}(x^{n-1}, y^{\ell_y-1}) = A_S^\diamond(x^{n-1}, y^{\ell_y-1}) &+ S_y(x^{n-1}, y^{\ell_y-1}) \cdot \bar{t}_x(x_n) \\
 &+ S_x(x^{n-1}, y^{\ell_y-1}) \cdot \bar{t}_y(y_{\ell_x}) \\
 &+ A_S(x^{n-1}, y^{\ell_y-1}) \cdot \bar{t}_x(x_n) \cdot \bar{t}_y(y_{\ell_x}),
 \end{aligned} \tag{78}$$

which allows for writing the dynamic algorithm as

$$A_S^\diamond(x, y) = \begin{cases} A_S^\diamond(x^{n-1}, y) & \text{if } x_n \notin y, \\ A_S^\diamond(x^{n-1}, y) + U_{xy}(x^{n-1}, y^{\ell_y-1}) & \text{if } x_n \in y \text{ and } x_n \notin x^{n-1}, \\ A_S^\diamond(x^{n-1}, y) + U_{xy}(x^{n-1}, y^{\ell_y-1}) \\ \quad - A_S^\diamond(x^{\ell_x}, y^{\ell_y}) \\ \quad + A_S^\diamond(x^{\ell_x-1}, y^{\ell_y-1}) & \text{if } x_n \in y \text{ and } x_n \in x^{n-1}. \end{cases} \tag{79}$$

The time complexity of this algorithm is proportional to mn as it simultaneously evaluates 4 arrays of order mn : $\phi(x^i, y^j)$, $S_x(x^i, y^j)$, $S_y(x^i, y^j)$ and the target array $A_S^\diamond(x^i, y^j)$. The variables ℓ_x and ℓ_y can be evaluated while iterating through the array $\phi(x^i, y^j)$.

In the same spirit, not even relying on averaged state durations, we can generalize the attribute ϕ_M that counts the number of matching subsequences to a version that weighs each match with the product of times spent:

$$\phi_M^\diamond(x, y) = \sum_{u \in \mathcal{S}(x, y)} \sum_{i_x(u) \in I_x(u)} \sum_{i_y(u) \in I_y(u)} \left(\sum_{i \in i_x(u)} t_x(i) \right) \cdot \left(\sum_{j \in i_y(u)} t_y(j) \right). \tag{80}$$

For lack of space, we do not present an algorithm to evaluate ϕ_M^\diamond . Elzinga (2005) provides for such an algorithm; a much faster one can be developed by modifying the graph-algorithm that evaluates A_M with the product-of-sums rule (75). We leave this as an exercise to the reader. Of course, associated metrics d_S^\diamond and d_M^\diamond can be constructed in the usual way.

5. CONCLUDING REMARKS

As announced in the introduction to this paper, we have presented a range of alternatives to OM and we have demonstrated how such metrics can be put to work; both by presenting feasible algorithms and by applying them to real data sets in the context of relevant demographical problems. Each of the metrics maps the categorical time series to points in a Euclidean space wherein it is natural to define similarity as an angle between these points. Furthermore, we have demonstrated how to adapt the metrics to handling durations other than by turning to a full-string representation of the underlying observations.

As announced too, we did not attempt to compare the metrics with respect to their general appropriateness, simply because such a comparison is impossible. There is no external criterion to evaluate the representations other than substantive theory and/or the sensibility of the results obtained. However, this does not make sequence analysis arbitrary; sequence analysis provides us with an opportunity to investigate different aspects of the same time series by using different metrics. We illustrate this by looking again at the family formation histories, only now by employing an attribute that is totally different from what we have seen so far. The metrics that we have discussed all rely on an attribute that considers one

TABLE 12. Average similarities $\bar{s}_{\mathcal{H}}$ between family formation histories, represented in full-string format, of four cohorts of Austrian women. Characteristic sequences are those that have the highest average similarity within the cohort.

Cohort	Size	$\bar{s}_{\mathcal{H}}$	sd	Char. Sequence
1945-49	539	.46	.13	S/54 M/1 MC/89
1950-54	558	.41	.12	S/48 M/2 MC/94
1955-59	650	.37	.10	S/57 M/3 MC/84
1960-64	752	.34	.10	S/63 M/2 MC/79

or more bigger “chunks of time” and then use the commonness of order within such chunks of time. Precisely because of this property, it is sensible to use an (x, \mathbf{t}_x) -format and not so sensible to use a string-format.

However, instead of comparing sequences by comparing orders of events, we may also compare them by primarily and directly looking at timing of events. But then it is only natural to represent the observations in a string-format since the string format explicitly and simultaneously shows the time piece and what was timed. Therefore, we discuss the next metric while assuming that the time series have been represented in string-format. We do so, because the metric focusses on timing by counting how often sequences show different states at the same position, i.e. at comparable moments in time. Thereto, we define the attribute

$$A_{\mathcal{H}}(x, y) = |\{i : x_i = y_i\}| \quad (81)$$

for sequences x and y with $|x| = m$ and $|y| = n$. So, $A_{\mathcal{H}}$ counts the positions on which x and y have identical states. Clearly, $0 \leq A_{\mathcal{H}}(x, y) \leq \min\{A_{\mathcal{H}}(x, x), A_{\mathcal{H}}(y, y)\}$ so

$$d_{\mathcal{H}}(x, y) = m + n - 2A_{\mathcal{H}}(x, y) \quad (82)$$

is a squared Euclidean distance and normalizing it yields

$$D_{\mathcal{H}}(x, y) = 1 - \frac{A_{\mathcal{H}}(x, y)}{\sqrt{m \cdot n}} \quad (83)$$

$$= \text{Prob}(x_i \neq y_i) \text{ if } |x| = |y| \quad (84)$$

So, we now have, presuming the sequences are equally long, that distance is equivalent with the probability that the states are unequal at any time. We use the subscript \mathcal{H} since $d_{\mathcal{H}}$ is an extension of the classical Hamming distance (Hamming, 1950) that simply counts the number of positions on which two equally long strings have different symbols. $d_{\mathcal{H}}$ is an extension since it does not require the sequences to be equally long. Also geometrically, the metric is quite different from the order-based metrics. For we construct the vectors by first ranking the elements $\sigma_i \in \Sigma$ as $r(\sigma_i) = i$ and then defining $m(i) = \lceil i / |\Sigma| \rceil$. Then, for $1 \leq i \leq |x| \cdot |\Sigma|$, we calculate the coordinates of the representing vector $\mathbf{x} = (\dot{x}_1, \dots)$ by

$$\dot{x}_i = \begin{cases} 1 & \text{if } i = (m(i) - 1) \cdot |\Sigma| + r(x_{m(i)}) \\ 0 & \text{otherwise} \end{cases} . \quad (85)$$

For the sequence $x = abac$ from $\Sigma = \{a, b, c\}$, this construction yields the vector

$$\mathbf{x} = (\underbrace{1, 0, 0}_{x_1=a}, \underbrace{0, 1, 0}_{x_2=b}, \underbrace{1, 0, 0}_{x_3=a}, \underbrace{0, 0, 1}_{x_4=c}),$$

i.e. for each position in x we have $|\Sigma|$ coordinates, one of which equals 1. In Table 8, we studied de-standardization of family formation through an LCS-based metric. The analysis revealed that the typical backbone-structure of family formation had changed over time and that average similarity decreased. Here we repeat this analysis, only now using $s_{\mathcal{H}}$ and $D_{\mathcal{H}}$, and we show the results in Table 12. First, we observe that the average $\bar{s}_{\mathcal{H}}$ decreases, just like $\bar{s}_{\mathcal{L}}^*$ in Table 8. Only, $\bar{s}_{\mathcal{H}}$ decreases more. So, when we look at timing, there seems to be more de-standardization than when we look at backbone structures. Furthermore, Table 12 shows that the characteristic sequence changes over cohorts, not by an insert of cohabitation as a new state U like in Table 8, but by living single for longer spells and a postponement of entering parenthood by almost 10 months.

The above example clearly demonstrates that sequence analysis does not become arbitrary or unscientific because we have many different metrics at our disposal. On the contrary, by changing metric we change perspective on the same time series and unveil a different structure. We might even confine ourselves to just comparing the temporal structures of time series by considering sequences like $x = abbabc$ and $y = ppqqqr$ as equivalent since their transitions occur simultaneously. The sum of conditional entropies $E(x|y) + E(y|x)$ would provide for a metric (Cover and Thomas, 1991, Chap. 2, Probl. 15) that exactly does just that.

We conclude this paper by drawing the reader's attention to two technical problems that may arise when applying $D_{\mathcal{S}}$ or $D_{\mathcal{M}}$ to sequences in string format. For consider the 72-months labor market entry career

$$x = \underbrace{\text{TT} \dots \text{T}}_{20} \underbrace{\text{UU} \dots \text{U}}_{12} \underbrace{\text{EE} \dots \text{E}}_{40}.$$

One 9-long subsequence of this career is $u = \text{TTUUUEEE}$. Clearly, there are $\binom{20}{2} \binom{12}{4} \binom{40}{3} = 929214000 = |x|_u$ embeddings of this one u in x . Now imagine that we have another career y that is not too different from x . Not unlikely then, we will find that $|y|_u = |x|_u$ and therefore we should calculate that $|y|_u \cdot |x|_u = 863438657796000000$. Only the latter figure has far too many digits to be representable in a normal 32-bits machine without taking special provisos. So, using $D_{\mathcal{S}}$ or $D_{\mathcal{M}}$ may cause problems because of lack of computational precision.

The second problem is also directly related to the lengths of the sequences. On the average, written in the (x, \mathbf{t}_x) -format, the family formation histories of the 2499 Austrian women have a length of 3.12 states and calculating the full distance matrix $\{d_{\mathcal{S}}(\cdot, \cdot)\}$ takes, on standard PC, about 178 seconds. We know that the processing time that is required is proportional to the product of the sequence length. So, switching from the (x, \mathbf{t}_x) -format to sequences consisting of 144 symbols each, will increase the length of the sequences by a factor of, roughly, 48 hence the processing time will increase with a factor $48^2 = 2304$. Therefore, the expected processing time will raise to approximately 409000 seconds, i.e. to over 100 hours! Admittedly, such technical problems will disappear in the future but for now, using $D_{\mathcal{S}}$ or $D_{\mathcal{M}}$ with sequences in string format is to be avoided.

REFERENCES

- Abbott, Andrew and John Forrest, 1986. "Optimal Matching Methods for Historical Sequences." *Journal of Interdisciplinary History* 15:471–491.
- Abbott, Andrew and Angela Tsay, 2000. "Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect." *Sociological Methods & Research* 29:3–33.
- Anyadike-Danes, Michael, 2005. Personal communication.
- Batagelj, Vladimir and Matevž Bren, 1995. "Comparing Resemblance Measures." *Journal of Classification* 12:73–90.
- Billari, Francesco C. and Rafaella Piccarreta, 2005. "Analysing Demographic Life Courses through Sequence Analysis." *Mathematical Population Studies* 12:81–106.
- Bookstein, Abraham, Vladimir A. Kulyukin, and Timo Raita, 2002. "Generalized Hamming Distance." *Information Retrieval* 5:353–375.
- Brückner, Hannah and Karl U. Mayer, 2005. "De-Standardization of the Life Course: What might it Mean? And if it means anything, Whether it Actually Took Place?" In *The Structure of the Life Course: Standardized? Individualized?*, edited by Ross Macmillan, pp. 27–53. Elsevier, Amsterdam.
- Brüderl, Josef, 2004. "Die Pluralisierung partnerschaftlicher Lebensformen in Westdeutschland und Europa." *Aus Politik und Zeitgeschichte*, 19:3–10.
- Brüderl, Josef and Stefani Scherer, 2005. "Methoden zur Analyse von Sequenzdaten." *Kölner Zeitschrift für Sociologie und SocialPsychologie* 44:330–347.
- Brzinsky-Fay, Christian, 2006. "Lost in Transition: Labour Market Entry Sequences of School Leavers in Europe." *WZB Discussion Paper* .
- Clote, Peter and Rolf Backofen, 2000. *Computational Molecular Biology: An Introduction*. Wiley, New York.
- Corijn, Martine and Erik Klijsing, 2001. *Transitions to Adulthood in Europe*. Kluwer, Dordrecht.
- Cover, Thomas M. and Joy A. Thomas, 1991. *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley, New York.
- Dijkstra, Wil and Toon Taris, 1995. "Measuring the Agreement Between Sequences." *Sociological Methods & Research* 24:214–231.
- Elzinga, Cees H., 2003. "Sequence Similarity - A Non-Aligning Technique." *Sociological Methods & Research* 31:3–29.
- , 2005. "Combinatorial Representation of Token Sequences." *Journal of Classification* 22:87–118.
- , 2007. "Turbulence in Categorical Time Series." , *unpublished* .
- Elzinga, Cees H. and Aart C. Liefbroer, 2007. "De-Standardization and Differentiation of Family Life Trajectories of Young Adults: A Cross-National Comparison." *European Journal of Population* , forthcoming.
- Eppstein, David, Zvi Galil, Rafaele Giancarlo, and Giuseppe F. Italiano, 1992a. "Sparse Dynamic Programming I: Linear Cost Functions." *Journal of the ACM* 39:519–545.
- , 1992b. "Sparse Dynamic Programming II: Convex and Concave Cost Functions." *Journal of the ACM* 39:546–567.
- 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.
- Fussell, Elisabeth and Anne H. Gauthier, 2005. "American Women's Transition to Adulthood in Comparative Perspective." In *On the Frontier of Adulthood: Theory, Research and Public Policy*, edited by R.A. Settersten, F.F. Furstenberg, and R.G. Rumbaut, pp. 76–109. University of Chicago Press, Chicago.
- Gower, John C., 1971. "A general coefficient of similarity and some of its properties." *Biometrics* 27:857–871.
- Gower, John C. and Pierre Legendre, 1986. "Metric and Euclidean properties of dissimilarity coefficients." *Journal of Classification* 3:5–48.
- Gusfield, Daniel M., 1997. *Algorithms on Strings, Trees and Sequences*. Computer Science and Computational Biology. Cambridge University Press, Cambridge.
- Hamming, Richard W., 1950. "Error-Detecting and Error-Correcting Codes." *Bell System Technical Journal* 26:147–160.
- Hartigan, John, 1975. *Clustering Algorithms*. Wiley, New York.
- Holliday, John D., C-Y. Hu, and Peter Willett, 2002. "Grouping of Coefficients for the Calculation of Inter-Molecular Similarity and Dissimilarity using 2D-Fragment Bit-Strings." *Combinatorial Chemistry and High-Throughput Screening* 5:155–166.
- Jaccard, Paul, 1912. "The Distribution of the Flora in the Alpine Zone." *New Phytologist* 11:35–50.
- Levenshtein, Vladimir I., 1966. "Binary codes capable of correcting deletions, insertions and reversals." *Soviet Physics Doklady* 10:707–710.
- Levine, Joel H., 2000. "But what have you done for us lately?" *Sociological Methods & Research* 29:34–40.
- Liefbroer, Aart C. and Frances Goldscheider, 2007. "Transitions to Adulthood: How Unique is Sweden in the European Context?" In *Immigration, Gender and Family Transitions to Adulthood in Sweden*, edited by Eva M. Berhardt, Calvin Goldscheider, Frances Goldscheider, and Gunilla Bjéren, pp. 203–227. University Press of America, Lanham, MD.
- 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 A* 165:317–334.
- Rick, Claus, 2000. "Simple and Fast Linear Space Computation of Longest Common Subsequences." *Information Processing Letters* 75:275–281.
- Rozenberg, Grzegorz and Arto Salomaa, editors, 1997. *Handbook of Formal Languages*. Springer, New York.
- Sankoff, David, 1974. "Matching Sequences under Deletion/Insertion Constraints." *Proceedings of the National Academy of Science USA* 69:4–6.
- Sankoff, David and Joseph B. Kruskal, editors, 1983. *Time warps, string edits and macromolecules. The Theory and Practice of String Comparison*. Reading: Addison-Wesley.
- Shanahan, Michael J., 2000. "Pathways to Adulthood: Variability and Mechanisms in Life Course Perspective." *Annual Review of Sociology* 26:667–692.
- Stovel, Katherine, 2001. "Local Sequential Patterns: The Structure of Lynching in the Deep South, 1882-1930." *Social Forces* 79:843–880.

- Stovel, Katherine and Mark Bolan, 2004. "Residential Trajectories: The Use of Sequence Analysis in the Study of Residential Mobility." *Sociological Methods & Research* 32:559–598.
- Tanimoto, T. T., 1957. "IBM Internal Report." Technical report, IBM.
- Wang, Hui, 2006. "Nearest Neighbors by Neighborhood Counting." *IEEE Transactions on Pattern Learning and Machine Intelligence* 28:1–12.
- Widmer, Eric, Rene Levy, Alexandre Pollien, Raphael Hammer, and Jacques-Antoine Gauthier, 2003. "Entre Standardisation, Individualisation et Sexuation: une Analyse des Trajectoires Personelles en Suisse." *Swiss Journal of Sociology* 29:35–67.
- Wilson, Clark, 2006. "Reliability of Sequence Alignment Analysis of Social Processes: Monte Carlo Tests of ClustalG Software." *Environment and Planning* 38:187–204.
- Wu, Lawrence L., 2000. "Some Comments on "Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect"." *Sociological Methods & Research* 29:41–64.
- Yianilos, Peter, 2002. "Normailized Forms of Two Common Metrics." Technical Report 91-082-3-9027-1 Rev 7/7/2002, NEC Research Institute.