Divisive Property-Based and Fuzzy Clustering for Sequence Analysis

Matthias Studer

1 Introduction

In this paper, we introduce property-based and *fuzzy* clustering and discuss their usefulness in the context of sequence analysis. We also present some tools available in R code by which to conduct these analyses. These two clustering methods aim to overcome some of the limitations of the more "traditional" ones, such as partitioning around medoids or agglomerative clustering.

Most of the clustering methods used in sequence analysis are polythetic, with the notable exception of model-based clustering. This means that cluster memberships are defined according to a broad set of properties and by comparing a given sequence to all the other ones. For this reason, the rules that define which sequences belong to which cluster are implicit. In other words, we do not know exactly on which grounds a sequence is assigned to a given cluster. Having a typology based on implicit rules of cluster membership has two disadvantages.

First, the resulting clustering is sample-dependent, which means it cannot be compared to another typology that is created in a subsample or another sample, for instance. Even comparing two typologies that appear to be similar might be problematic, since their underlying implicit clustering rules might differ.[1] On the other hand, having explicit rules would allow one to validate a typology in other

[1] We cannot be sure that the clustering rules are sufficiently similar, because we do not know them. They are only implicit.

M. Studer (✉)
NCCR LIVES and Geneva School of Social Sciences, University of Geneva, Geneva, Switzerland
e-mail: matthias.studer@unige.ch

G. Ritschard, M. Studer (eds.), *Sequence Analysis and Related Approaches*,
Life Course Research and Social Policies 10,
https://doi.org/10.1007/978-3-319-95420-2_13

samples and reproduce a given (validated) typology in other studies. Implicit rules therefore hinder the reproducibility of life-course research, as well as the possibility of undertaking a large-scale literature review.

Second, implicit rules make the interpretation of clustering more difficult. The first step after creating a typology is usually to try to interpret and recover these implicit rules by using different kinds of graphics and analyses. Explicit clustering rules would make the interpretation of clustering results much easier.

Monothetic clustering—which here we call "property-based clustering"—aims to create a sequence typology defined by explicit classification rules. In this paper, we introduce a method that is based on the "DIVCLUS-T" algorithm proposed by Chavent et al. (2007). Following the work of Piccarreta and Billari (2007), we also discuss its use in sequence analysis and extend their work by proposing for consideration new sets of state sequence features. Finally, we make the analytical results broadly available in the `WeightedCluster` package.

In this paper, we also discuss the use of *fuzzy* clustering, which aims to overcome another limitation of "traditional clustering." In sequence analysis, we usually use *crisp* clustering (i.e., each sequence is assigned to only one sequence type). In *fuzzy* clustering, however, each sequence belongs to one or more clusters, with a certain degree or strength (D'Urso 2016).

The *fuzzy* approach has several advantages over the more usual *crisp* one (D'Urso 2016). First, sometimes some sequences are between two sequence types. In *crisp* clustering, these sequences would be assigned to one of the two types; in *fuzzy* clustering, however, these sequences would be considered a *hybrid-type* or a *mixture* of the two types (D'Urso 2016). From a statistical viewpoint, *fuzzy* clustering might lead to better results if some sequences are between two (or more) sequence types. This case might occur frequently, according to Warren et al. (2015), who argue that exact cluster membership should not be trusted. From a sociological perspective, this approach is of special interest when the trajectories are not strongly structured into types, and when we can think that some individuals can be influenced by several sequence types.

Second, in *fuzzy* clustering, membership is thought to be *gradual*. Some sequences are more central (typical) of a given type than are others. This is also an interesting property from a sociological viewpoint. From the Weberian *ideal–typical* perspective, nobody is the perfect incarnation of an ideal type, but some are closer to it than others. This sociological perspective is similar to the gradual membership approach inherent in *fuzzy* clustering.

For all these reasons, the use of the *fuzzy* clustering approach is promising in life-course research and sequence analysis. However, to the best of our knowledge, it has been only seldom used in sequence analysis. One of the likely reasons for this paucity is the lack of proper tools by which to analyze sequences in conjunction with a membership matrix, instead of a categorical covariate (as in *crisp* clustering). For instance, Salem et al. (2016) used *fuzzy* clustering, but they ultimately assigned each sequence to the cluster with the highest membership in all subsequent analyses, thus turning in fact back to *crisp* clustering. In this paper, we propose different tools to fill this gap and make use of the full information of the membership matrix.

This paper is organized as follows. It starts by presenting property-based clustering and the set of sequence properties whose consideration we propose. We then turn to the presentation of *fuzzy* clustering and introduce several ways of representing the results, before interpreting them. We also discuss how to properly analyze cluster membership according to some explanatory covariates. Before concluding, we briefly show how to run the proposed analysis in `R`, using our `WeightedCluster` library.

2 Sample Issue

The usefulness of the proposed methodology is illustrated by using the data and the research question from McVicar and Anyadike-Danes (2002), who studied school-to-work transition in Northern Ireland. Their analysis was undertaken in two steps. They started by identifying ideal-typical trajectories, before explaining clustering membership by using information such as qualification at the end of compulsory schooling, family background, and demographic characteristics. Their aim was to "identify the 'at-risk' young people at age 16 years and to characterize their post-school career trajectories" (p. 317). To build our clustering, we use optimal matching with constant cost.

3 Property-Based Clustering

The aim in using property-based clustering is to build a sequence typology by identifying well-defined clustering rules that are based on the most relevant properties of the analyzed object. In the literature, these clustering methods are called *monothetic divisive* clustering methods (Chavent et al. 2007), and they were first introduced in sequence analysis by Piccarreta and Billari (2007). We propose here a conceptual presentation of the "DIVCLUS-T" algorithm (a detailed presentation can be found in Chavent et al. 2007). The "DIVCLUS-T" algorithm is very similar to one proposed by Piccarreta and Billari (2007). Our choice between the two is based on availability in `R`. Here, we mainly extend the work of Piccarreta and Billari (2007), by proposing new sets of sequence features for consideration.

3.1 Principle

Property-based clustering uses two kinds of information. First, the sequence properties are used to build the rules. Second, it uses a dissimilarity matrix, which is used to measure variation in the sequence and how much of this variation can be explained by a given property; this is in line with the discrepancy analysis framework (Studer et al. 2011).

The method then works in two steps: tree building and splits ordering. In the tree building phase, all sequences are grouped in an initial node. Then, this node is split according to one of the object properties, into two subnodes or clusters. This property and the associated split are chosen in such a way that the split "explains" the biggest share of the sequence discrepancy (Studer et al. 2011; Chavent et al. 2007; Piccarreta and Billari 2007). The process is then repeated on each new node until a stopping criterion is found. First, the algorithm might stop because there are no further relevant properties by which to make a split. Second, nodes with only one observation are obviously not split. This first step is roughly equivalent to the procedure that Piccarreta and Billari (2007) propose.

As in Piccarreta and Billari (2007), our implementation of the "DIVCLUS-T" algorithm also makes it possible to specify a minimum number of observations per node. Splits that would lead to at least one node with fewer than this minimum number of observations are discarded. This is a useful extension, if we want to ensure that all clusters represent at least a given percentage (for instance, 5%) of the sequences. One might also restrict to "significant" splits by using permutation tests, as in discrepancy analysis (Studer et al. 2011). However, the usefulness of the latter approach is subject to discussion, as the concept of significance is not very well defined in cluster analysis. This first step of the procedure can be seen as a decision tree, where the splits are chosen according to the explanatory power of the considered properties. In fact, our implementation is based on the tree-structured discrepancy analysis.

Once the whole tree is built, the splits are ordered according to their overall "relevance." More precisely, this "relevance" is measured by calculating the increase in the share of the overall discrepancy that is explained by adding a split. This procedure has the advantage of maximizing a global criterion. Ultimately, the result of this procedure is a series of nested partitions ranging from one group to a number of groups, any of which depend on the stopping criteria or a maximal number of groups to consider.[2] This second step is the major difference from the procedure proposed by Piccarreta and Billari (2007), where the final clustering and the stopping criteria depend on a pruning procedure.

Figure 1 graphically represents the procedure, using our illustrative example of school-to-work transition in Northern Ireland, for the first nine splits. The order of the splits is presented on the right, with the associated number of clusters. We start at the top of the tree with a single cluster. At this stage, the most relevant feature in splitting this top node into two is to have spent more (or less) than 17 months in higher education (property "duration.HE"). Stopping at this level would result in a cluster in two groups (presented on the right of Fig. 1). The clustering in three groups is obtained by splitting the node "less than or equal to 17 months in higher education" in two groups: having spent more (or less) than 33 months in employment (criterion "duration.EM"). This last split was preferred to the solution of splitting the node "more than 17 months in higher education," because it has greater explanatory power regarding sequence variation at a global level. The procedure then continues until some stopping criteria are met.

[2] In other words, new groups are added by dividing one of the existing groups into two subgroups.

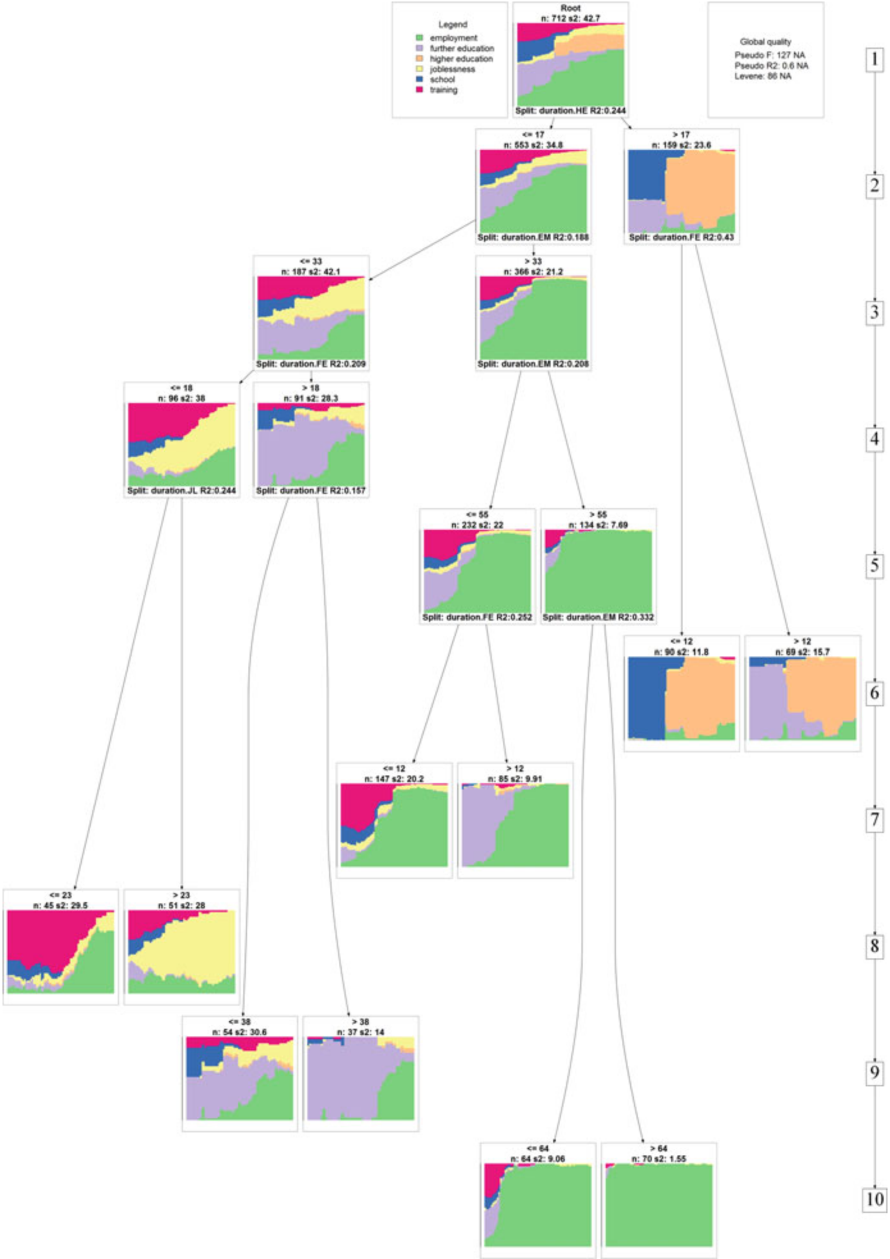

Fig. 1 Clustering tree

As with any agglomerative or divisive clustering, the choice of the number of groups can be grounded on cluster quality measures (see Studer 2013, for a review). Here, various clustering quality measures agree on choosing either the solutions in four ($ASW = 0.41$) or nine ($ASW = 0.38$) groups as the best clustering. According to the empirical evaluation of Chavent and Lechevallier (2006), for a small number of groups, the "DIVCLUS-T" algorithm tends to produce better clustering (measured on the basis of statistical criteria) than Ward clustering. However, the reverse becomes true as the number of groups increases.

3.2 Property Extraction

The results of property-based clustering are highly dependent on the properties of the object to be clustered. In the empirical evaluation made by Chavent and Lechevallier (2006), having a large number of meaningful properties was one of the key elements that led to good-quality clustering. In this section, we propose a set of properties worthy of consideration. In our implementation of the algorithm, these properties are automatically extracted.

Within the life course paradigm, three main dimensions of the trajectories are of central interest: the timing, the duration, and the sequencing of the states (Studer and Ritschard 2016). We propose the automatic extraction of various properties that measure each of these dimensions.

To measure the timing of the state, we consider the state at each time position t. If we consider the sequence of length ℓ, we therefore end with ℓ categorical covariates that measure the situation over time. Piccarreta and Billari (2007) propose another way of measuring the timing, by considering the spells that form the sequences. They generate one property $A_{s,k}$ that stores the age at the beginning of the kth spell in state s. If the property is not observed (i.e., there is no kth spell in state s), they propose setting it to $\ell + 1$, where ℓ is the length of the sequence. However, we take here a slightly different strategy: we set it to a missing value. Missing values are then treated as a special case when defining a split. Although we use the numeric information, when available, to define the split by using an inequality, the missing values are attributed to one of the two groups—whichever gives the best result.

We propose the use of two sets of properties to measure duration. First, we consider the duration in the successive spells. This is achieved by generating one property $D_{s,k}$ that stores the duration of the kth spell in state s. Second, we consider the overall time spent in each state that results in one property per state.

We use frequent subsequence mining to extract the properties that measure the state sequencing. With this method, the aim is to uncover frequent subsequences within a set of sequences (Studer et al. 2010; Agrawal and Srikant 1995; Zaki 2001). A subsequence s is defined as a subsequence of x if all the states of s occur in the same order as in x. For instance, the sequence $A - C$ is a subsequence of sequence $A - B - C$ because A and C occur in the same order. A subsequence is said to be frequent if it is found in more than a predefined percentage of sequences. Using this framework, several sets of sequence properties can be extracted. First, we look for frequent (in our case, 1%) subsequences in the sequence of distinct successive states (DSS). This step generates one property (i.e., variable) per identified frequent subsequence, and stores the number of times that the subsequence is found in each sequence. For instance, the subsequence $A-C$ occurs twice in the sequence $A-B-C-B-C$. We also consider the age at the first occurrence of the pattern (i.e., when the pattern starts). Second, we look for frequent subsequences within the transition sequences. This is achieved by specifying a distinct state for each transition. For instance, the DSS $A-B-C$ will be recoded as $A-AB-BC$ before running the analysis. Here again, the number of occurrences and the age at the first occurrence are stored as properties.

Finally, the user can consider and add other properties. The algorithm computes different sequence complexity measures. Piccarreta and Billari (2007) suggest adding information about covariates such as education level. Application-specific sequence properties could also be of interest. For instance, one might have an interest in adding the time spent in a state of joblessness within the last 12 months of our sequences, if professional integration at the end of the sequence is of primary concern.

Although all these properties are automatically extracted, they need to be carefully chosen according to the issue under investigation. For instance, for a study that mainly concerns itself with timing differences, we suggest restricting attention to timing-related properties. For this reason, in our implementation of the algorithm, one can specify the sets of properties to be considered. Table 1 summarizes the properties considered in this study. The first column contains the name of the property used in our R implementation, and the second provides brief descriptions of the sets of properties.

Table 1 Sequence properties considered in the clustering algorithms

Name	Description
`state`	The state in which an individual is found, at each time position t
`spell.age`	The age at the beginning of each spell of a given type
`spell.dur`	The duration of each of the spells presented above
`duration`	The total time spent in each state
`pattern`	Count of the frequent subsequences of states in the DSS
`AFpattern`	Age at the first occurrence of the above frequent subsequence
`transition`	Count of the frequent subsequence of events in each sequence, where each transition is considered another event
`AFtransition`	Age at the first occurrence of the above frequent subsequence
`Complexity`	Complexity index, number of transitions, turbulence

3.3 Running the Analysis in R

Property-based clustering can be run using the `seqpropclust` function available in the `WeightedCluster` package. Aside from the state sequence object `myseq`, one needs to specify the distance matrix (argument `diss`), the properties (by default, all properties are computed), and the maximum number of clusters under consideration.

```
R>  ## Clustering using properties "state" and "duration"
R>  pclust <- seqpropclust(myseq, diss=diss, maxcluster=5,
       properties=c("state", "duration"))
R>  ## Displaying the resulting clustering
R>  seqtreedisplay(pclust, type="d", border=NA,  showdepth=TRUE)
R>  ## Computing clustering quality and cluster membership
R>  pclustqual <- as.clustrange(pclust, diss=diss, ncluster=5)
```

The clustering membership can be extracted by using the `as.clustrange` function, which also computes various cluster quality measures. See Studer (2013) for a detailed presentation of this procedure.

4 Fuzzy Clustering

In *crisp* clustering, each sequence is assigned to exactly one sequence type. The result is a categorical covariate that summarizes the typology. In *fuzzy* clustering, each sequence can belong to more than one cluster; this is achieved by computing the degree or strength of membership of each sequence to each identified sequence type (D’Urso 2016). This is of central interest when the sequences are not thought to be strongly structured, or when some sequences could have been influenced by more than one type.

More precisely, the result of *fuzzy* clustering is a membership matrix **u** comprising one row per individual and one column per cluster. Each value u_{iv} of this matrix measures the membership strength of an individual i to each cluster v. These membership degrees, which usually sum to 1, are also called “probabilities,” and they lead to two slightly different interpretations. The concept of membership “strength” or “degree” refers to the closeness of each sequence to each type. The notion of “membership probabilities” refers to the chances that the underlying sequence was generated according to one of the types.

In this section, we start by presenting the algorithm used herein. We then propose different approaches to describing and visualizing the results of *fuzzy* clustering, using the membership matrix. Finally, we discuss possible strategies by which to analyze how explanatory covariates are linked to cluster membership; again, this is based on the membership matrix. We hope that the availability of these tools will lead to more widespread use of *fuzzy* clustering in sequence analysis.

4.1 Fanny Algorithm

We use the *Fanny* algorithm proposed by Kaufman and Rousseeuw (1990) and later adapted by Maechler et al. (2005). This algorithm aims to minimize the following function:

$$\sum_{v=1}^{k} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} u_{iv}^{r} u_{jv}^{r} d(i, j)}{2 \sum_{j=1}^{n} u_{jv}^{r}} \quad , \tag{1}$$

where n is the number of observations, k a predefined number of groups, u_{iv} the membership value of individual i to cluster v, and $d(i, j)$ the distance between sequences i and j. The exponent r is a *fuzziness* parameter that needs to be set by the user. A value of 2 is often used, but values between 1.5 and 2.5 are usually recommended (D'Urso 2016). The standard procedure is to start with $r = 2$ and use a smaller value if the algorithm does not converge.

Using our sample data, we used a value of 1.5 for the fuzziness parameter. We kept a solution in seven groups, based on the interpretability of the resulting clustering and the aim of the study.

4.2 Plotting and Describing a Fuzzy Typology

Once the clustering has been computed, typically, the first step is to describe each cluster and give it a first interpretation. In this section, we propose several approaches to doing so by using the membership matrix.

4.2.1 Most Typical Members

A first way to label the clusters and interpret them is to identify typical sequences. In "traditional sequence analysis clustering," this can be done by identifying the medoid or a representative sequence based on other criteria, such as neighborhood density (Gabadinho et al. 2011). A natural way to do it with *fuzzy* clustering is to choose the sequence with the highest membership strength in each cluster. The first row of Table 2 presents this information. Using this strategy, we can have a first look at our clustering membership matrix. We found a first cluster related to full employment, then three patterns of education (i.e., training or further education) followed by employment: two patterns leading to higher education, and one pattern of training followed by joblessness.

Table 2 also presents descriptive statistics of membership strength for each cluster. As one will recall, in *fuzzy* clustering, each sequence has a measure of its membership strength in each cluster. Hence, it is not possible to compute a percentage of sequences belonging to each cluster, as we would do for *crisp*

Table 2 Descriptive statistics of the cluster membership matrix

	Mean	Min.	Max.	SD	Sum
(EM,70)	0.20	0.00	0.99	0.31	142.98
(TR,23)-(EM,47)	0.17	0.00	0.94	0.25	123.27
(FE,22)-(EM,48)	0.17	0.00	0.94	0.23	119.09
(FE,46)-(EM,24)	0.12	0.00	0.88	0.19	88.22
(FE,25)-(HE,45)	0.10	0.00	0.96	0.23	73.96
(SC,25)-(HE,45)	0.14	0.00	0.98	0.29	97.21
(TR,22)-(JL,48)	0.09	0.00	0.80	0.16	67.27

Table 3 Example of an augmented dataset

Sequence	Weight	Cluster
s_1	u_{11}	1
s_1	u_{12}	2
s_1	u_{13}	3

clustering. However, average cluster membership provides similar information: it can be interpreted as the relative frequency of each cluster, if sequences are weighted according to their membership strength.

The maximal value is also interesting, as it provides an estimation of the quality of the chosen representative. The higher the membership, the better the representative (i.e., a value of 1 would identify a sequence that fully belongs to that cluster). In some clusters, the maximum is quite low, if we consider that the maximal possible value is 1. For instance, in the training–joblessness cluster, the maximum equals 0.8. Hence, our representative is also close to other clusters—perhaps cluster 2. To describe the clusters, we therefore need to take into account more information than just the sequences with the highest membership.

4.2.2 Weight-Based Presentation

Our second proposition in analyzing the *fuzzy* cluster is to weigh the sequences according to their membership strength or probabilities. We augment the dataset by repeating the sequence s_i of individual i k times (i.e., once per cluster). We therefore have k sequences for individual i, denoted as $s_{i1} \cdots s_{ik}$. We weight these sequences according to their membership degree $u_{i1} \cdots u_{ik}$. Hence, even if the same sequence were repeated k times, its weights will sum to 1. We then create a new categorical covariate in this augmented dataset, and it specifies the cluster (ranging from 1 to k) of the associated membership degree.

Table 3 presents a small example, to make the presentation clearer. Suppose we have three clusters named 1, 2, and 3. For individual 1 with the sequence s_1, we have three observations (i.e., one per cluster). The first observation is weighted according to the strength of membership to the first cluster, and the associated cluster covariate is set to 1. We then repeat the same procedure for each observation.

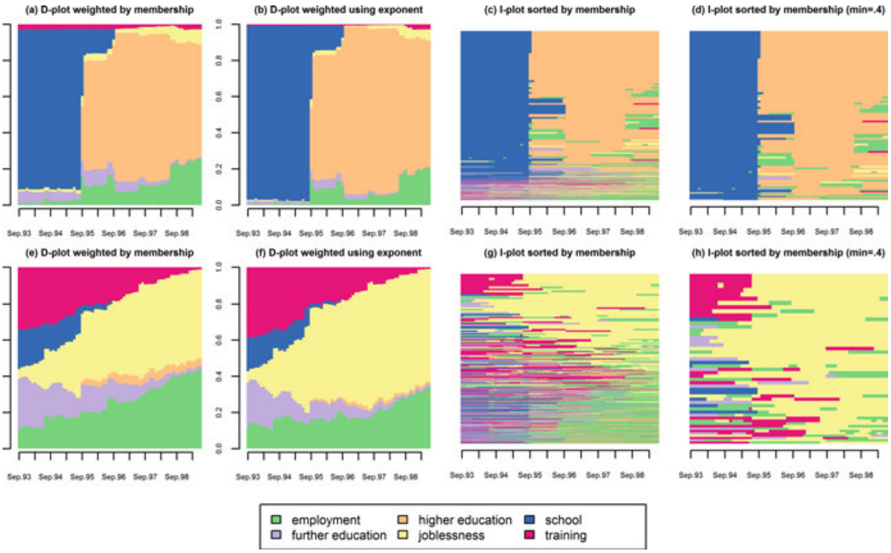

Fig. 2 Membership-weighted plots of the sequence for clusters “(SC,25)-(HE,45)” and “(TR,22)-(JL,48)”

This weighting strategy allows us to use any tools available for weighted sequence data. For instance, we can use a sequence distribution plot. Figure 2 proposes several plots of these membership-weighted sequence data for the clusters “School–Higher Education” and “Training–Joblessness.” Subfigures (a) and (e) present sequence distribution plots for these clusters. If the cluster “School–Higher Education” seems to be quite well defined, the “Training–Joblessness” one shows more discrepancy, as we already noted.

This weighting strategy is also supported from a more statistical perspective. Minimizing Eq. 1 is equivalent to minimizing a residual sum of squares in a discrepancy analysis of this augmented dataset (Studer et al. 2011). More precisely, it minimizes the residual sums of squares of this augmented dataset, where each sequence is weighted $u^r_{i1} \cdots u^r_{ik}$ and the explanatory categorical covariate would be the cluster $1 \dots k$.

Following this reasoning, we can weigh the sequences by using the exponent (here, $r = 1.5$). The result might be closer to the underlying algorithm. However, the interpretation is more difficult and, as we will see, it is also interesting in the following analysis to rely on the membership strength. For this reason, this approach should be used mostly to describe the clusters. The result of this strategy is shown using a d-plot in subfigures (b) and (f); in both cases, the cluster seems to be better defined.

By using index plots, we can take a closer look at the longitudinal patterns. In this case, we additionally suggest ordering the sequences according to membership degree. The result is shown in subfigures (c) and (g). The most typical sequence lies at the top of the subfigures, with a high membership degree; meanwhile, the bottom shows less-characteristic patterns. Interpretation should be made with caution, as it depends on the maximal membership degree. In the cluster “School–Higher Education,” this maximum is close to 1, while in the other one it reaches only 0.8.

It can be interesting to focus on the sequences with the highest membership. In subfigures (d) and (h), we discarded sequences with a membership degree below 0.4. Interestingly, among the sequences with the highest membership in the “Training–Joblessness” cluster, we find sequences starting in “Further Education” or “Employment,” for instance.

We propose several methods by which to visualize and describe a *fuzzy* typology; these methods allow us to properly interpret this typology. However, most sequence analysis applications go beyond the typology description by studying the factors that influence the kinds of trajectories being followed. We now turn to this kind of analysis for *fuzzy* clustering.

4.3 Analyzing Cluster Membership Using Dirichlet Regression

In typical sequence analysis, one often relies on multinomial regression to explain cluster membership (Abbott and Tsay 2000). The aim is to identify how covariates explain the trajectory type that is followed. This cannot be done with *fuzzy* clustering, because our typology is described by a membership matrix and not by a categorical variable. Assigning each sequence to the cluster with the highest membership strength is not a solution either, for in doing so, we would lose all the added value inherent in *fuzzy* clustering.

Several models are available to analyze membership matrices, which can be seen as “compositional data” (Morais et al. 2016; Pawlowsky-Glahn and Buccianti 2011). Here, for two reasons, we suggest relying on Dirichlet regressions (Maier 2014), which are extensions of beta regression (Ferrari and Cribari-Neto 2004). First, interpretations of them are very similar to those of multinomial models, if we use the so-called alternative parametrization. Second, good performance is reported in Maier (2014) and in Morais et al. (2016), even under some violations of the statistical assumptions. In the current model, one of the categories (i.e., clusters) is chosen as the reference; we then estimate how explanatory factors influence the likelihood of being fully classified in a category, rather than in the reference.

Interpretations of the coefficients are very similar, then, to the multinomial ones, and they can be interpreted in the usual log-odds scale. Their exponents can therefore also be interpreted as “odds-ratio” values on cluster membership. In a Dirichlet regression, one can also estimate the effect of covariates on a “precision” parameter that measures the precision of estimation. (This parameter is named “precision” because it takes a high value when the residual variance of the dependent variable tends to be lower.) This can be used to take into account possible heteroscedasticity.

Table 4 presents the coefficients of the Dirichlet regression. We used the employment cluster as the reference. To simplify our presentation, we included only three covariates: gcseq5eq (the qualifications gained by the end of compulsory

Table 4 Dirichlet regression of cluster membership

	(TR,23)-(EM,47)	(FE,22)-(EM,48)	(FE,46)-(EM,24)	(FE,25)-(HE,45)	(SC,25)-(HE,45)	(TR,22)-(JL,48)
(Intercept)	-0.14^{*}	-0.14^{*}	-0.40^{***}	-0.77^{***}	-0.92^{***}	-0.53^{***}
	(0.07)	(0.07)	(0.07)	(0.07)	(0.07)	(0.07)
Grammaryes	0.09	0.03	0.05	0.18	0.82^{***}	0.16
	(0.13)	(0.13)	(0.13)	(0.13)	(0.12)	(0.13)
gcse5eqyes	0.13	0.37^{***}	0.55^{***}	0.98^{***}	1.06^{***}	0.42^{***}
	(0.10)	(0.10)	(0.10)	(0.10)	(0.10)	(0.10)
funempyes	0.10	0.06	0.13	0.09	0.04	0.26^{*}
	(0.13)	(0.13)	(0.13)	(0.13)	(0.13)	(0.13)

Log likelihood = 6402.14; Num. obs. = 712; Precision= 1.22^{***} (0.02); $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

education: five or more GCSEs at Grades A to C, or equivalent), Grammar (grammar school secondary education), and funemp (father unemployed at the time of the survey).

Let us interpret the coefficient of our covariate on the membership degree (adequacy with) or chance (if we think about probabilities) to follow the more "at-risk" pattern "(TR,22)-(JL,48)," as opposed to employment. We observe no significant difference between those having had a grammar school education and those who had not. However, individuals with unemployed fathers did tend to have a higher membership in this cluster (significant positive coefficient) than in the employment cluster—or, if we take the "probability" interpretation, they have a lower chance of following this pattern than the reference (employment). The same applies to those who had the five-grade qualification (variable `gcse5eq`)—probably because they are very unlikely to follow the reference employment trajectory.

Additionally, it is often useful to understand the distinctive features of each cluster. For *crisp* clustering, this can be achieved by running a logistic regression on a dummy variable that measures cluster membership. Here, we can make use of beta regression, which aims to model a dependent variable that lies in the [0, 1] interval (Ferrari and Cribari-Neto 2004).[3] The interpretation of the coefficient is similar to that of the Dirichlet regression. The exponent of the coefficients can be interpreted as an "odds-ratio" on cluster membership. Here, again, a "precision" parameter allows us to take into account over- or under-dispersion. The results lead to similar conclusions but further highlight that those who had the five-grade qualification (variable gcse5eq) are very unlikely to follow the employment trajectory type of sequence.

[3] Unlike logistic (binomial) regression, beta regression does not assume that the dependent variable is a proportion (i.e., the result of a count of 0 and 1). Furthermore, it can cope with over- or under-dispersion.

4.4 Running the Analysis in R

The Fanny algorithm is available in the `cluster` package, through the `fanny` function. Aside from the distance matrix `diss`, one needs to specify the number of groups (argument `k=7`) and set the argument `diss=TRUE` to specify that we provided a distance matrix and not a dataset. Finally, the value of the fuzziness parameter *r* can be set through the `memb.exp` argument (default value of 2). The returned object provides the membership matrix (`fclust$membership`) and additional information such as quality measures or related *crisp* clustering.

```
R>  ## Fuzzy clustering in 7 groups using r=1.5
R>  fclust <- fanny(diss, k=7, diss=TRUE, memb.exp=1.5)
R>  ## Display the resulting clustering with membership
     threshold of 0.4
R>  fuzzyseqplot(myseq, group=fclust$membership, type="I",
          membership.threshold=0.4, sortv="membership")
R>  ##Estimation of Dirichlet Regression
R>  ##Dependent variable formatting
R>  fmember <- DR_data(fclust$membership)
R>  ## Estimation
R>  bdirig <- DirichReg(fmember~var1+var2|1,
          data=mydata, model="alternative")
R>  ## Displaying results of Dirichlet regression
R>  summary(bdirig)
R>  ## Estimation of beta regression
R>  breg1 <- betareg(fclust$membership[, 1]~var1+var2, data=mydata)
R>  ## Displaying results
R>  summary(breg1)
```

All the visualizations proposed here are available in the `WeightedCluster` package, through the `fuzzyseqplot` function. The function works in the same ways as the usual `seqplot` function available in `TraMineR`, except that the `group` argument should be a membership matrix or a `fanny` object. Furthermore, one can specify a membership threshold (for instance, 0.4) and whether graphics should be weighted by membership strength. If one wants to weights the sequences using the fuzziness parameter, one should set `memb.exp` to the correct value. By default, the fuzziness parameter is not used; hence, the `memb.exp=1`. When using index plots (`type="I"`), one can additionally set `sortv="membership"` to sort the sequences in each plot according to their membership strength.

Dirichlet regression can be estimated using the `DirichReg` function in the `DirichletReg` package (Maier 2014), while the beta regression can be computed with the function `betareg` available in the `betareg` package (Cribari-Neto and Zeileis 2010). For the former, the dependent variable should first be formatted using the function `DR_data` before estimating the model using `DirichReg`. For beta regressions, a separate regression should be estimated for each cluster. One needs to specify the cluster membership strength on the right-hand side of the R formula, while adding covariates on the left-hand side as usual. In both cases, one can set a data frame where covariates should be found, using the usual data argument.

5 Conclusion

In this paper, we introduced two alternative clustering methods, each of which has its own strengths. We believe that property-based clustering is a very promising sequence analysis tool. Having clustering membership rules allows one to reproduce and validate a typology; furthermore, it significantly simplifies the interpretation of the clustering.

Property-based clustering is also useful in understanding the underlying criteria used by a dissimilarity measure to compare trajectories. For instance, in our example application, all splits were made according to the overall time spent in different states. This prevalence of duration illustrates once again that optimal matching tends to favor duration while comparing sequences. The use of other distance measures such as those reviewed in Studer and Ritschard (2016) or those introduced in this bundle in Collas (2018) or Bison and Scalcon (2018) would lead to the selection of other properties. For instance, sequence pattern properties would probably be selected by using a distance sensitive to sequencing, such as SVRspell (Elzinga and Studer 2015).

On the other hand, *fuzzy* clustering has been seldom used in sequence analysis. Nonetheless, the method should be useful in many situations. First, in many cases, exact cluster membership is doubtful (Warren et al. 2015). *Fuzzy* clustering allows one to relax the assumption that cluster memberships have been correctly retrieved by the cluster analysis; it does so by allowing multiple cluster memberships. This is also an interesting perspective from a sociological viewpoint, as trajectories might be influenced by several trajectory types. Second, in *fuzzy* clustering, membership is thought to be gradual; this too is interesting from a social science perspective. Some trajectories might be more typical of a type than others.

The aim of this study was to develop tools by which to facilitate the use, interpretation, and analysis of both clustering methods. However, further application of these methods is still needed to fully assess their strengths and weaknesses with regards to sequence analysis. We believe that this study is a first step in that direction.

Acknowledgements The author warmly thanks the two reviewers for their insightful comments on an earlier version of the manuscript. This publication benefited from the support of the Swiss National Centre of Competence in Research LIVES Overcoming vulnerability: Life course perspectives (NCCR LIVES), which is financed by the Swiss National Science Foundation (grant number: 51NF40–160590). The author is grateful to the Swiss National Science Foundation for its financial assistance.

References

Abbott, A., & Tsay, A. (2000). Sequence analysis and optimal matching methods in sociology, Review and prospect. *Sociological Methods and Research*, *29*(1), 3–33. (With discussion, pp. 34–76).

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In P. S. Yu & A. L. P. Chen (Eds.), *Proceedings of the International Conference on Data Engeneering (ICDE), Taiwan* (pp. 487–499). IEEE Computer Society.

Bison, I., & Scalcon, A. (2018). From 07.00 to 22.00: A dual-earner couple's typical day in Italy. Old questions and new evidence from social sequence analysis. In G. Ritschard & M. Studer (Eds.), *Sequence Analysis and Related Approaches: Innovative Methods and Applications*. Cham: Springer (this volume).

Chavent, M., & Lechevallier, Y. (2006). Empirical comparison of a monothetic divisive clustering method with the ward and the k-means clustering methods. In V. Batagelj, H.-H. Bock, A. Ferligoj, & A. Žiberna (Eds.), *Data science and classification* (pp. 83–90). Berlin/Heidelberg: Springer.

Chavent, M., Lechevallier, Y., & Briant, O. (2007). DIVCLUS-T: A monothetic divisive hierarchical clustering method. *Computational Statistics & Data Analysis, 52*(2), 687–701.

Collas, T. (2018). Multiphase sequence analysis. In G. Ritschard & M. Studer (Eds.), *Sequence analysis and related approaches: Innovative methods and applications*. Cham: Springer (this volume).

Cribari-Neto, F., & Zeileis, A. (2010). Beta regression in R. *Journal of Statistical Software, 34*(2), 1–24.

D'Urso, P. (2016). Fuzzy clustering. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 545–573). New York: Chapman & Hall.

Elzinga, C. H., & Studer, M. (2015). Spell sequences, state proximities and distance metrics. *Sociological Methods and Research, 44*(1), 3–47.

Ferrari, S., & Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics, 31*(7), 799–815.

Gabadinho, A., Ritschard, G., Studer, M., & Müller, N. S. (2011). Extracting and rendering representative sequences. In A. Fred, J. L. G. Dietz, K. Liu, & J. Filipe (Eds.), *Knowledge discovery, knowledge engineering and knowledge management* (Communications in computer and information science (CCIS), Vol. 128, pp. 94–106). Berlin/Heidelberg: Springer.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data. An introduction to cluster analysis*. New York: Wiley.

Maechler, M., Rousseeuw, P., Struyf, A., & Hubert, M. (2005). Cluster analysis basics and extensions. Rousseeuw et al. provided the S original which has been ported to R by Kurt Hornik and has since been enhanced by Martin Maechler: Speed improvements, silhouette() functionality, bug fixes, etc. See the 'Changelog' file (in the package source).

Maier, M. J. (2014). Dirichletreg: Dirichlet regression for compositional data in R. Research Report Series/Department of Statistics and Mathematics 125. WU Vienna University of Economics and Business, Vienna.

McVicar, D., & Anyadike-Danes, M. (2002). Predicting successful and unsuccessful transitions from school to work using sequence methods. *Journal of the Royal Statistical Society A, 165*(2), 317–334.

Morais, J., Thomas-Agnan, C., & Simioni, M. (2016). A tour of regression models for explaining shares. Working Paper 16–742, Toulouse School of Economics.

Pawlowsky-Glahn, V., & Buccianti, A. (Eds.) (2011). *Compositional data analysis: Theory and applications*. Chichester: Wiley.

Piccarreta, R., & Billari, F. C. (2007). Clustering work and family trajectories by using a divisive algorithm. *Journal of the Royal Statistical Society: Series A (Statistics in Society), 170*(4), 1061–1078.

Salem, L., Crocker, A. G., Charette, Y., Earls, C. M., Nicholls, T. L., & Seto, M. C. (2016). Housing trajectories of forensic psychiatric patients. *Behavioral Sciences & The Law, 34*(2–3), 352–365.

Studer, M. (2013). WeightedCluster library manual: A practical guide to creating typologies of trajectories in the social sciences with R. LIVES Working Papers 24, NCCR LIVES, Switzerland.

Studer, M., & Ritschard, G. (2016). What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures. *Journal of the Royal Statistical Society, Series A, 179*(2), 481–511.

Studer, M., Müller, N. S., Ritschard, G., & Gabadinho, A. (2010). Classer, discriminer et visualiser des séquences d'événements. *Revue des nouvelles technologies de l'information RNTI, E-19*, 37–48.

Studer, M., Ritschard, G., Gabadinho, A., & Müller, N. S. (2011). Discrepancy analysis of state sequences. *Sociological Methods and Research, 40*(3), 471–510.

Warren, J. R., Luo, L., Halpern-Manners, A., Raymo, J. M., & Palloni, A. (2015). Do different methods for modeling age-graded trajectories yield consistent and valid results? *American Journal of Sociology, 120*(6), 1809–1856.

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning, 42*(1/2), 31–60.