Sequencing of Categorical Time Series

Christian Richter, Martin Luboschik, Martin Röhlig, Heidrun Schumann [*]

Abstract

Exploring and comparing categorical time series and finding temporal patterns are complex tasks in the field of time series data mining. Although different analysis approaches exist, these tasks remain challenging, especially when numerous time series are considered at once. We propose a visual analysis approach that supports exploring such data by ordering time series in meaningful ways. We provide interaction techniques to steer the automated arrangement and to allow users to investigate patterns in detail.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI) I.5.3 [Clustering]: Algorithm—Similarity measures

1 Problem Description

Categorical time series consist of nominal data values. In many domains, such time series result from some underlying configurable process, e.g., machine learning algorithms for activity recognition [4]. In this case, different time series emerge from alternative parametrizations of the algorithms. To evaluate the algorithms and the influence of the available parameters, typically a large amount of time series needs to be analyzed and compared to find patterns and relate them to the examined parametrizations. Although several analysis approaches in the fields of time series data mining and hierarchical dimension ordering address this problem (cf. [5]), it still remains challenging to support users in such analysis tasks.

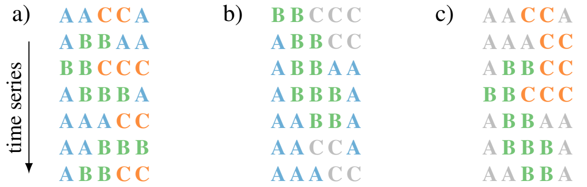
To ease direct comparison of multiple time series, they are commonly visualized in a row-wise alignment (Fig. 1). Esling et al. [2] compare such visual analysis with the extraction of knowledge from the data's shape and refer to the humans' natural perception capacity to perform such tasks through visual skills. Nevertheless, this is only possible if the time series are sorted in a way that certain patterns become visible. To facilitate the recognition of temporal patterns, it is required to arrange the time series in appropriate sequences. Ideally, such sequences are groupings of similar time series in regard to specific aspects (e.g., co-occurrences). To obtain these sequences, three major challenges have to be addressed.

The first challenge is to select appropriate similarity metrics. Well known numerical metrics, e.g., L_p norms, can not directly be applied since they do not fit the characteristics of categorical data. Simply applying them would imply representing the different categories with numerical values which introduces an artificial order. On the other hand, edit distances as categorical metrics are better suited for our time series data.

The second challenge is to find algorithms that generate suitably ordered sequences of multiple time series, so that immanent patterns are unveiled. To bring similar time series close together, it makes sense to group time series by their similarity and then determine a good order of the groups. Thus, a similarity based sequencing typically results in a hierarchy similar to hierarchical clustering. Hence, the success of a sequencing algorithm – in regard to unveiling patterns – basically relies on building such a hierarchy and finding an appropriate traversal order.

The third challenge is dealing with the large number of possible sequences. A set of n categorical time series results in $n!$ possible permutations, each of them potentially reveals noteworthy patterns. Even reducing the number of sequences using a fixed set of different similarity metrics and deterministic sequencing algorithms, it remains challenging to find sequences which match specific analysis goals. Hence, manual inspecting and analyzing several different sequences, even when they are tailored to some specific application requirements, remains necessary.

Due to these three challenges it is obvious that there is *no* single sequence that reveals *all* relevant patterns (Fig. 1). Bertin [1] states: *"a graphic is constructed and reconstructed until it reveals all the relations in the data"*. On this account, we present a visual analysis approach that supports users in generating meaningfully ordered sequences of categorical time series using automated and interactive means and thus, facilitates the visual detection of temporal patterns.



a)	b)	c)
AACCA	BBCCC	AACCA
ABBAA	ABBCC	AAACC
BBCCC	ABBAA	ABBCC
ABBBA	ABBBA	BBCCC
AAACC	AABBA	ABBAA
AABBB	AACCA	ABBBA
ABBCC	AAACC	AABBA

Figure 1: Different sequences of categorical time series. The random sequence a) does not reveal patterns. The sequence b) allows for comparing occurrences of categories A and B but not of C. The sequence c) shows patterns of categories B and C but not of A.

2 Approach

Our visual analysis approach consists of two main aspects: First, we extend the edit distance. Second, we propose a novel automated sequencing algorithm. Further, we integrate interaction techniques that allow for steering the automated sequencing, enable exploring subsets of time series, and support manual adjustments for finding global and local patterns.

2.1 Similarity Metric

A well known metric applicable to categorical time series is the edit distance [3]. The edit distance between two time series is defined by the minimal number of the operations substitute, insert, and delete, that are necessary to convert one time series into the other. Although this metric adequately rates similar time series, its sensitivity decreases with larger numbers of differences. Moreover, temporal aspects are neglected as it only registers the fact of deviation (e.g., at which time points differences are located). We address this problem by applying a weighting scheme to the time points. Time points that are temporally far apart are weighted differently than time points that are closer together. This way, we derive a temporally weighted edit distance that produces different similarity values for edit operations at different time points.

2.2 Sequencing Algorithm

To support the recognition of patterns within the set of all possible sequences, we follow two criteria:

(a) group similar time series and
(b) form transitions between groups.

[*] University of Rostock, E-mail: {christian.richter, martin.luboschik, martin.roehlig, heidrun.schumann}@uni-rostock.de

According to this, our sequencing algorithm comprises three steps. The first step works as a hierarchical clustering to find groups. The next step is to determine a suitably ordered sequence of the groups generated in the first step. To this end, it is necessary to define a good starting point for traversing the clustering hierarchy. In the last step the final sequence is extracted. The complete procedure is detailed in the following.

1. Creating Groups Each time series is considered as a node. All nodes are connected with each other by weighted edges that reflect the edit distance between the nodes (Fig. 2a). We start building the hierarchy by selecting a random node as root and from there on construct a minimum spanning tree (MST, Fig. 2b). For this purpose, we use a modified version of Prim's algorithm. Instead of allowing an arbitrary connection of equidistant nodes, we select one node to serve as the parent of the others. Without this modification, the MST may comprise connections that introduce unnecessary breaks in the MST-traversal. The MST clusters similar time series to address criterion (a). Criterion (b) is considered as the MST connects different groups by a path of minimal edges and consequently forms a transition between the groups by the nodes along this path.

2. Finding a Starting Point The randomly selected root node in the previous step is often not an appropriate starting point for the tree traversal in the next algorithm step. If the node has multiple large subtrees (Fig. 2c), the node would cause multiple large breaks in the resulting sequence due to the tree traversal. To reduce these breaks, we search for a proper node that simplifies the hierarchy tree by rearranging the tree along this node (the yellow node in Fig. 2c). This node can be found by different heuristics (e.g. the deepest node). In many cases the rearrangement decreases the width of the tree (Fig. 2d) and consequently results in a simpler tree structure that causes fewer breaks in the resulting sequence of the next step.

3. Sequence Extraction To finally extract an ordered sequence of the categorical time series from the MST, the tree is traversed using a depth-first search. To further meet criterion (b), we developed a decision function that determines the traversal order in case of multiple branches. It uses a density measure based on the subtrees and traverses compact subtrees in favor of loosely-packed ones. By doing so, diverging time series located in loosely-packed subtrees are rather located at the end of the final sequence.

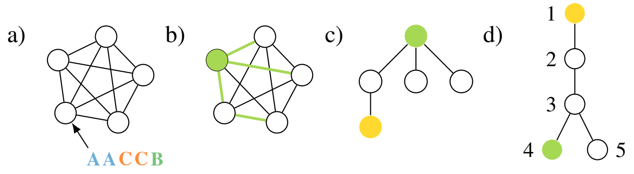


Figure 2: Time series (nodes) and their similarities (edges) are represented as a graph (a). A random root node (green) is selected and similar time series are grouped to create a hierarchy (b). The deepest node (yellow) is selected as the new root (c). The hierarchy is rearranged and traversed from the root to extract a sequence (d).

2.3 Interaction

The resulting sequences of the above algorithm provide an appropriate basis for further explorations. To address user needs and interests, we integrate several interactions to refine the sequences. First, the user can choose between alternative weighting schemes for the edit distance to focus on different temporal aspect. Additionally, it is possible to build ordered sequences based on selected time intervals or categories. An interactive reordering of time series within a sequence is allowed by drag and drop.

3 Discussion and Future Work

The presented visual analysis approach extends prior work in the field of sequencing categorical time series data. Our approach is generally independent from the concrete visualization and has been developed to support the finding and recognition of patterns in aligned categorical time series. For example, it can be applied to pixel-based visualizations [4] or to glyph-based visualizations (Fig. 1). Figure 3 shows its application to color-coded categorical time series. The default (arbitrary) order of the time series (Fig. 3a) suggest a random distribution. Figures 3b and c illustrate how the order influences what can be seen from the visualization. The sequence in Fig. 3b, generated with the original edit distance, groups the co-occurrence of categories all at once. In contrast, the sequence in Fig. 3c has been generated using an edit distance with an ascending time point weighting. This sequence emphasizes time series that are more similar in the last quarter of the data. As we see, our weighted metric and new sequencing approach are a promising method to deal with categorical time-evolving data.

While examining different data sets, it became apparent that an appropriate description of user interests and their translation to corresponding metrics remains the most challenging part. So far, there is no solution to automatically map a user interest to a proper selection of similarity metrics and parametrization of the sequencing algorithm. Thus, it remains tedious to explore the data for potentially noteworthy sequences.

For future work we plan to extend our approach to make it more flexible and to further integrate the user's interests and the properties of the data. To support the user in the analysis of different sequences and to make them comparable, we intend to calculate multiple metrics of information content (e.g. entropy, signal-to-noise ratio) and to evaluate to what extend these metrics address the user's interests. This way, it is possibly to generate multiple sequences and to suggest those with a high information content. Our approach has been implemented in a interactive prototype and is available for demonstration.

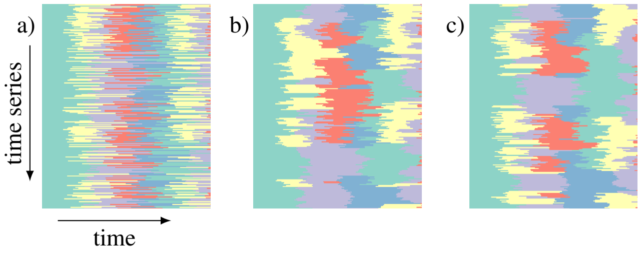


Figure 3: Sequencing results of a set of 216 categorical time series with 500 time points. The arbitrary sequence (a) results in a noisy visualization with no apparent patterns. The sequence (b) was generated using the edit distance and reveals groups of similar time series. The sequence (c) was produced using a temporally weighted edit distance and shows a stronger grouping towards the end.

References

[1] J. Bertin. *Graphics and graphic information processing*. Walter de Gruyter, 1981.

[2] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.

[3] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33:2001, 1999.

[4] M. Röhlig, M. Luboschik, M. Bögl, F. Krüger, B. Alsallakh, T. Kirste, S. Miksch, and H. Schumann. Supporting Activity Recognition by Visual Analytics. In *Proc. of IEEE VAST*, 2015.

[5] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proc. of IEEE INFOVIS*, 2003.