Computational Statistics and Data Analysis 177 (2023) 107583

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

www.elsevier.com/locate/csda

Nonparametric bagging clustering methods to identify latent structures from a sequence of dependent categorical data

Konrad Abramowicz [a], Sara Sjöstedt de Luna [a], Johan Strandberg [b,*]

[a] Department of Mathematics and Mathematical Statistics, Umeå University, Umeå, Sweden
[b] Department of Statistics, Umeå School of Business, Economics and Statistics, Umeå University, Umeå, Sweden

ARTICLE INFO

Article history:
Received 29 June 2021
Received in revised form 31 January 2022
Accepted 25 July 2022
Available online 2 August 2022

Keywords:
Categorical dependent data
Clustering
Bagging methods
Entropy

ABSTRACT

Nonparametric bagging clustering methods are studied and compared to identify latent structures from a sequence of dependent categorical data observed along a one-dimensional (discrete) time domain. The frequency of the observed categories is assumed to be generated by a (slowly varying) latent signal, according to latent state-specific probability distributions. The bagging clustering methods use random tessellations (partitions) of the time domain and clustering of the category frequencies of the observed data in the tessellation cells to recover the latent signal, within a bagging framework. New and existing ways of generating the tessellations and clustering are discussed and combined into different bagging clustering methods. Edge tessellations and adaptive tessellations are the new proposed ways of forming partitions. Composite methods are also introduced, that are using (automated) decision rules based on entropy measures to choose among the proposed bagging clustering methods. The performance of all the methods is compared in a simulation study. From the simulation study it can be concluded that local and global entropy measures are powerful tools in improving the recovery of the latent signal, both via the adaptive tessellation strategies (local entropy) and in designing composite methods (global entropy). The composite methods are robust and overall improve performance, in particular the composite method using adaptive (edge) tessellations.

1. Introduction

Clustering methods typically aim at partitioning a given set of observations into homogeneous groups, such that high similarity of objects within clusters and low in-between them is attained. When data consists of independent observations, there exist a well studied battery of tools for clustering, including, k-means, hierarchical clustering and parametric mixture models, see, e.g., Gan et al. (2007) for an exhaustive summary of clustering methods. In this paper, we focus on clustering methods for dependent categorical data observed on a one dimensional grid. We assume that the observed categorical data has been generated by a sequence of hidden states. Given the hidden state, the observed category for a specific time point is generated from a state-specific probability distribution over d labels. The goal is to recover the hidden states based on the observed categorical data.

A prominent and well-studied example of a model generating such data is the discrete-time, discrete-valued Hidden Markov Model (HMM). Due to the Markov assumption on the hidden states' dependence structure, the sequence exhibits

* Corresponding author.
E-mail address: johan.strandberg@umu.se (J. Strandberg).

https://doi.org/10.1016/j.csda.2022.107583

specific distributional properties. Likelihood-based methods, such as the EM algorithm, can be used to estimate the underlying distributional parameters. Once the model is estimated, the most likely hidden state sequence can be predicted using, e.g., the Viterbi decoding algorithm. We refer to Zucchini et al. (2017) for an extensive summary of HMM models and its estimation algorithms. Some deterministic non-parametric clustering methods for detection of latent state sequences include a modification of the k-means method, the so called embedded segmental (ES) k-means method. It has been proposed and used in the context of speech recognition by Kamper et al. (2017b) and Bhati et al. (2018) and is a modification of a Bayesian approach developed in Kamper et al. (2017a). In ES k-means, the data set is first partitioned into segments of consecutive observations, each segment mapped (via an embedding function) to a constant length vector (local representative) and then clustered using weighed k-means (see, e.g., Tseng, 2007) with weights being determined by the length of the segments. The optimisation algorithm iteratively updates the segmentation and the clustering until an optimal solution is found. ES k-means is sensitive to the initial segmentation, which is suggested to be randomly chosen or chosen by some segmentation algorithm that groups similar neighbouring observations into the same segments. The latter is rarely suitable for the types of situations we consider, since neighbouring observations do not need to be similar (to belong to the same latent state/segment).

In our paper, we propose and study a class of random non-parametric bagging clustering methods to recover the sequence of hidden states from the sequence of observed categorical data. The methods combine the categorical data type characteristics, the ordered nature of the data, random tessellations and the bagging framework. This paper extends and augments the studies presented in Abramowicz et al. (2019) by considering new non-adaptive and adaptive random tessellation strategies, modified clustering methods as well as creating composite methods which have the ability to improve the overall classification accuracy. Abramowicz et al. (2019) discuss an application that is relevant to our paper, related to reconstruction of past climate based on annually laminated varved lake sediment from Northern Sweden, covering more than 6000 years. The seasonal pattern of each varve of the lake sediment contains information about the weather the year the varve was produced. In climatology, it is of interest to study climate, being defined as frequencies of weather types over longer time periods, rather than fluctuating annual weather. The focus is thus on how to reconstruct past climate regimes from the identified annual weather types. Here the weather types would correspond to the categorical data and the states of the latent signal be the different climate regimes. It is worth underlining, however, that the problem of recovering the hidden regimes, using both parametric and non-parametric methods, has been studied in one and multiple dimensions within many other applied fields, including but not limited to medicine (Andreao et al., 2006), marketing (Lemmens et al., 2012) and speech recognition (Bhati et al., 2018).

In our work, similarly to the ES k-means, we partition the data and cluster the corresponding segments representatives. In our work segments are represented by the frequency vectors of the observed categories within the segment and rather than explicitly looking for the optimal solution along partitions, we consider multiple random tessellations (bootstrap samples). Since such an approach incorporates randomness, we obtain access to additional assessments tools, that are not available for deterministic methods. An example of such a tool is entropy, which can be used for validation and further improvements of existing techniques. The clustering methods we study here use the randomness inherent in the bootstrap samples of the bagging strategies, and utilise the Shannon entropy to extract information about the result's reliability. Both global and local measures of the final clustering stability are formed. The global normalised average entropy is used to select the parameters of the clustering method. The local entropy, computed for each site, gives information about cluster assignment stability/certainty at individual sites. We incorporate the local information in novel adaptive sampling strategies that lead to adjusted tessellations and improved overall performance. We also use entropy to create a framework for combining single clustering methods into composite ones. Such compositions select the most stable clustering mechanisms to suit a specific data set.

Even though we here apply and evaluate bagging clustering methods on dependent categorical (label) data and use frequency vectors as representatives for each segment, the framework is general, and can be applied to complex dependent data. Methods utilizing the bagging framework for dependent data have already been applied to cluster (see, e.g., Secchi et al., 2013; Abramowicz et al., 2017), and to perform inference, e.g., in kriging and regression problems (Menafoglio and Secchi, 2017).

The rest of this paper is organized as follows. In Section 2 we describe the bagging clustering framework and present four clustering methods (within the framework) to recover the hidden layer. In Section 2.3 we discuss parameter selection and the (Shannon) entropy as a tool to aid some parameter choices. The entropy, which measures the stability of the clustering, is further used to adapt the initial clustering methods which lead to four additional adaptive clustering methods, see Section 2.4. Section 3 presents a simulation study which compares the performance of the introduced eight clustering methods. We also introduce four composite methods from the above eight clustering methods and investigate their performance in Section 4. Comparisons with the Viterbi algorithm are summarized in Section 4.2. In Section 5 we discuss and present our conclusions. Finally, Appendix A provides additional figures, and Appendix B and Appendix C details and present results from an extended simulation study.

2. Methods

In this section we formalise the clustering problem and introduce the bagging clustering methodology. The methodology relies on multiple random tessellations (partitions) of the domain and clustering the local representatives of the tessellation

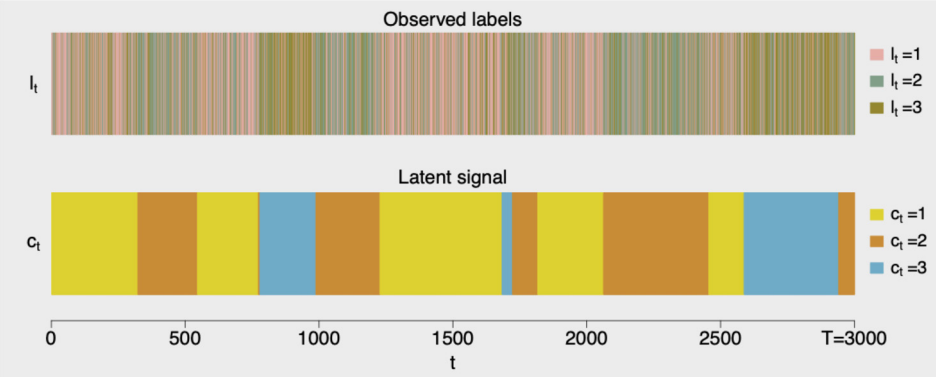

Fig. 1. A simulated example illustrating the dynamics of the observed labels (top) and the states of the latent signal (bottom).

cells. We present several new and existing ways of forming random tessellations and clustering the local representatives, which combined leads to eight bagging clustering methods.

2.1. Problem formulation

Consider a time series (sequence) of dependent observed categorical data, represented by the labels

$$l_1, l_2, \ldots, l_T, \qquad T \in \mathbb{N}^+,$$

where each label takes a value from the set $\{1, 2, \ldots, d\}$, $d \in \mathbb{N}^+$. We assume that the labels have been generated by a random mechanism that depends on an underlying unobservable (hidden) slowly varying latent signal

$$c_1, c_2, \ldots, c_T,$$

taking values in the hidden state space $\{1, 2, \ldots, K\}$, $K << T$. Each of the K hidden states of the latent signal is associated with a state-specific probability distribution. Given the state c_t, the observable label l_t is generated from the corresponding state-specific probability distribution. Connecting to the climate reconstruction example mentioned in the introduction, the l_t's correspond to the observable d annual weather types and the latent signal to the (slowly varying) dynamics of the climate, with K different climatic regimes. Further, for natural reasons, we want to use relatively small K, as we look for stable and significantly different climatic regime.

The objective of our work is to construct clustering methods, within the bagging clustering framework, that are suitable for recovering the hidden states (clusters), based on the information given by the observed labels (the l_t's) without any assumptions about the probability distributions generating the labels and the nature of the time dynamics for the latent signal. Hence, the resulting clusters should correspond to the states of the latent signal.

In the special case when the dynamics of the latent variable $\{c_t\}$ follows a Markov Chain, a Hidden Markov Model is generating our data. Fig. 1 shows an example of a generated sequence of observed labels and states of the latent signal when $d = 3$, $K = 3$, and the sample size is $T = 3000$.

2.2. The bagging clustering framework

The bagging (clustering) methods for dependent data consist of two phases: a bootstrap and an aggregation phase. In the bootstrap phase, each bootstrap replicate is typically composed by three steps.

Bootstrap phase. For $b = 1, \ldots, B$, we repeat the following steps:

Step 1. Generate a random tessellation (partition) of the domain {1,2,...,T} into n cells (subsequent sub-intervals) $\{V_i^b\}_{i=1}^n$.

Step 2. Compute the local representative in each cell. For $i = 1, \ldots, n$, the local representative $\mathbf{x}_i^b$ should summarise the information carried by the elements within the cell V_i^b. (In our work $\mathbf{x}_i^b$ are d-dimensional frequency vectors of the observed data labels).

Step 3. Cluster the local representatives $\{\mathbf{x}_1^b, \ldots, \mathbf{x}_n^b\}$ into K clusters using a suitable clustering method. Assign the cluster label obtained for each representative $\mathbf{x}_i^b$, to all sites (indicies) belonging to the corresponding cell.

For each of the B replicates in the Bootstrap phase, we save the cluster labels (states) for each of the T sites.

Aggregation phase. After the bootstrap phase, the aggregation phase provides the final clustering. First, a relabelling procedure, e.g., according to Stephens (2000), is applied in order to match the classifications across the bootstrap replicates. This step is crucial to assure that the possible permutations of the cluster numbering do not corrupt the final result. Let C_t^b be the label of site t in the b-th bootstrap replicate (after the relabelling), for $t = 1, \ldots, T$ and $b = 1, \ldots, B$. Then, for each site $t = 1, ..., T$, the relative frequency vector of cluster assignments over the bootstrap replicates is computed as

$$\hat{\pi}_t^q = |\{b \in \{1, \ldots, B\} : C_t^b = q\}|/B, \qquad \forall\, q = 1, \ldots, K, \tag{1}$$

where $|A|$ denotes the number of elements in set A. The final cluster assignment, for each site, is now obtained by a majority vote with respect to the K clusters, i.e.,

$$\hat{c}_t = \arg \max_{q=1,\ldots,K} \hat{\pi}_t^q, \qquad t = 1, \ldots, T.$$

The sequence $\hat{c}_1, \ldots, \hat{c}_T$ represents the recovered latent signal, resulting in non-contiguous intervals of clusters labels.

2.2.1. Domain partition

In this subsection, we present two ways of forming random partitioning of the domain in Step 1 of the bootstrap phase. The two methods lead to different distributions of the resulting cell lengths, and as such may affect the final clustering. The first partition mechanism is the discrete version of a Voronoi tessellation, and the second a modification that we introduce and call edge tessellation, both described below. In Section 2.4 additional ways of forming random partitions are introduced, the so called adaptive tessellations.

Voronoi tessellation. Starting from the set of n nuclei $\{Z_1^b, \ldots, Z_n^b\}$, obtained by equiprobable sampling of n items without replacement from the set of integers (sites) $I = \{1, \ldots, T\}$, the set of Voronoi cells $\{V_i^b\}_{i=1}^n$ is obtained by assigning each site index $t = 1, \ldots, T$, to the nearest nucleus Z_i^b. When a site index is at equal distance to two nuclei, we here decide to randomly assign it to one of the two.

Edge tessellation. A set of $n-1$ edges $\{E_1^b, \ldots, E_{n-1}^b\}$ are uniformly picked without replacement from the set of integers $I = \{1, \ldots, T-1\}$. The edges are then arranged in increasing order $\{E_{(1)}^b, \ldots, E_{(n-1)}^b\}$, and the set of random intervals $\{V_i^b\}_{i=1}^n$ are obtained by letting $V_1^b = [1, ..., E_{(1)}^b]$, $V_i^b = [E_{(i-1)}^b + 1, ..., E_{(i)}^b]$ for $i = 2, ..., n-1$ and $V_n^b = [E_{(n-1)}^b + 1, ..., T]$.

Distributional properties of the cell lengths in both tessellation schemes can be inferred from the properties of (continuous-time) homogeneous Poisson process (see, e.g., Ross, 2012), for which the inter-event times follow an exponential distribution. For such processes, given the number of events in an interval, the events are uniformly distributed within the interval - which we use in our simulation mechanisms. Applying the continuous time reasoning to the tessellation obtained, it follows that the distribution of the cell lengths for edge tessellation corresponds to a discrete version of the exponential distribution (i.e. the geometric distribution) while, for the Voronoi tessellation, it becomes a discrete version of a gamma distribution (the sum of two exponential distributions). To visualise the difference between the distributions of the cell lengths generated by Voronoi and edge tessellations, we present two histograms of the cell lengths in Fig. 2, when a domain of length $T = 3000$ is divided into 30 cells (of average length 100). The histograms are based on aggregated data obtained from 1000 independent simulations. Fig. 2 illustrates the larger variance of the cell length distribution for edge tessellations, compared to Voronoi tessellations. The edge tessellations are thus more likely to result in very small or large cells, while the Voronoi partitions yield cells of less variable size.

2.2.2. Local representatives

In this paper, we form the local representatives as the relative frequency vectors of the observed categorical data in the tessellation cells since we assume that the states of the latent structure cause different frequency patterns in the observed categorical data. For $i = 1, \ldots, n$, the local representative $\mathbf{x}_i^b$, corresponding to the i-th cell V_i^b of the b-th partition, is thus computed as the d-dimensional relative frequency vector

$$\mathbf{x}_i^b = \left[\frac{|\{m \in I : m \in V_i^b \text{ and } l_m = 1\}|}{|V_i^b|}, \cdots, \frac{|\{m \in I : m \in V_i^b \text{ and } l_m = d\}|}{|V_i^b|} \right].$$

Even though we have chosen to work with the relative frequencies, the methodology is flexible with respect to the selection of local representatives depending on the nature of the data and the inferential problem at hand.

2.2.3. Clustering local representatives

In Step 3 of the bootstrap phase of the bagging clustering algorithm, the local representatives of the cells are clustered. This can be done in many different ways. Here we present the multivariate K-means method as well as one of its modifications that takes into account the lengths of the individual cells.

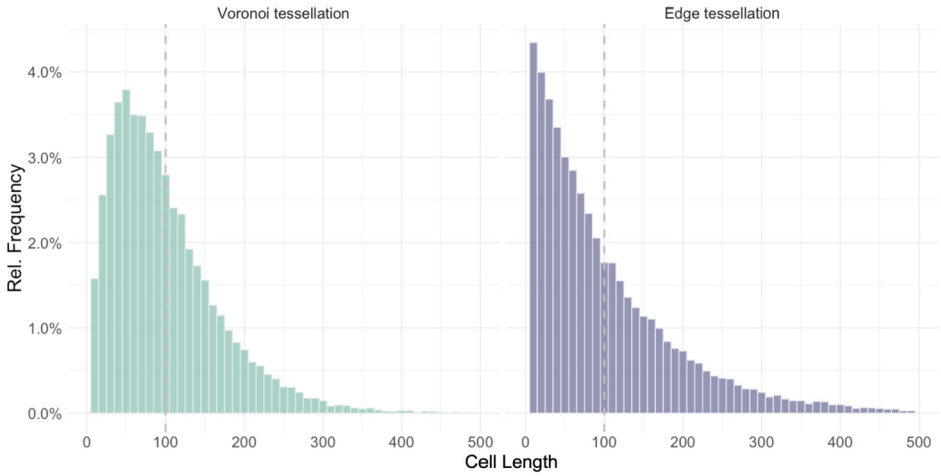

Fig. 2. Distributions of individual cell lengths when a domain of length $T = 3000$ is divided into $n = 30$ cells. The left and right panels show Voronoi and edge tessellations, respectively. The dashed vertical lines correspond to the average length of cells in each tessellation, i.e. 100.

K-means. The multivariate K-means method (KM) aims to cluster the d-dimensional local representatives $\mathbf{x}_1^b, \ldots, \mathbf{x}_n^b$, of the b-th bootstrap replicate into K homogeneous groups, say $\{C_1^b, \ldots, C_K^b\}$. More specifically, using the Euclidean norm $\|\cdot\|_2$, the objective of KM is to find the partitioning that minimizes the sum of squared Euclidean distances

$$\min_{C_1^b,\ldots,C_K^b} \sum_{k=1}^{K} \sum_{\mathbf{x}_i^b \in C_k} \|\mathbf{x}_i^b - \hat{\boldsymbol{\mu}}_k^b\|_2^2,$$

where

$$\hat{\boldsymbol{\mu}}_k^b = \frac{1}{|C_k^b|} \sum_{\mathbf{x}_i^b \in C_k^b} \mathbf{x}_i^b,$$

is the mean value of the local representatives belonging to cluster C_k^b. The Euclidean norm is just one example of how to measure the distance between relative frequency vectors, and other measures, such as the Aitchison distance (see, e.g., Egozcue et al., 2003) or entropy-based measures (see, e.g., Lin, 1991) can also be considered.

Weighted K-means. The weighted multivariate K-means method (wKM) (see, e.g., Kamper et al., 2017b; Tseng, 2007) allows the local representatives to have unequal importance in the clustering process, by taking into account different assigned weights to different representatives. More specifically, using the Euclidean norm, the objective of wKM is to find the partitioning that minimizes

$$\min_{C_1^b,\ldots,C_K^b} \sum_{k=1}^{K} \sum_{\mathbf{x}_i^b \in C_k^b} w(\mathbf{x}_i^b) \|\mathbf{x}_i^b - \hat{\boldsymbol{\mu}}_k^b\|_2^2,$$

where

$$\hat{\boldsymbol{\mu}}_k^b = \frac{1}{\sum_{\mathbf{x}_i^b \in C_k^b} w(\mathbf{x}_i^b)} \sum_{\mathbf{x}_i^b \in C_k^b} w(\mathbf{x}_i^b) \mathbf{x}_i^b,$$

and $w(\mathbf{x}_i^b)$ is the weight assigned to the local representative $\mathbf{x}_i^b$. In this paper, the weights are equal to the length of the cell of which $\mathbf{x}_i^b$ is based upon, i.e., $w(\mathbf{x}_i^b) = |V_i^b|$. The reason for that is motivated by the fact that if all elements within an interval belong to the same cluster, then the longer the interval, the better its representative would depict the structure of its true cluster mean.

2.2.4. Some bagging clustering methods

We initially study four bagging clustering methods, applied to the observed categorical data, to recover the latent signal. By combining the two tessellation generating mechanisms and the two clustering techniques we obtain the four bagging clustering methods presented in Table 1. The methods differ in their construction and have the potential to capture different features of the data. We compare their performance through a simulation study presented in Section 3 and Appendices A-B.

Table 1
Four bagging clustering methods obtained by combining the two tessellation generating mechanisms and the two clustering techniques.

	Clustering	
tessellation	k-means	weighted k-means
Voronoi	**BVKM**	**BVwKM**
edge	**BEKM**	**BEwKM**

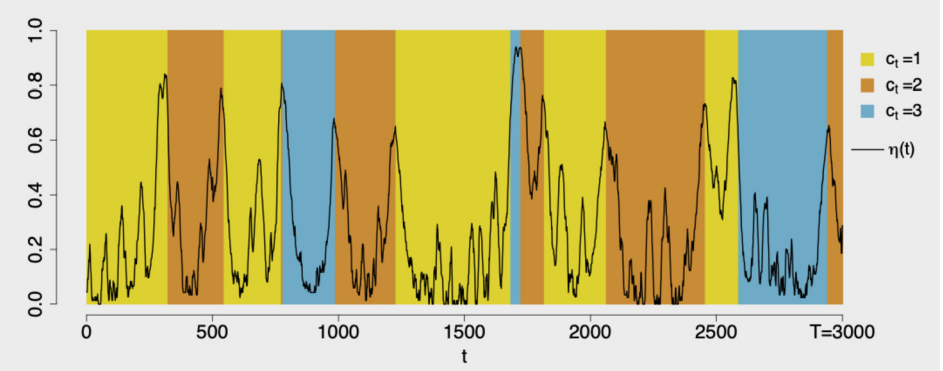

Fig. 3. The latent signal of Fig. 1 and the corresponding computed normalized entropy function $\eta(i)$ (solid line) of the relative frequencies of assignment for each site $t = 1, ..., 3000$ when the bagging clustering method *BEKM* is applied to the simulated observed data of Fig. 1 with $K = 3$ and $n = 75$.

2.3. Parameter selection and entropy

For all of the introduced methods, we need to select two parameters, K and n. Parameter K quantifies the number of different states of the latent signal, which is often specified from the problem at hand and expert knowledge. Suggestions on how to choose it based on the observed data are addressed, e.g., by Secchi et al. (2013) and Abramowicz et al. (2017, 2019), and relate to careful analysis of the structure of the final clustering. A natural strategy is to start with a smaller value of K and observe the change in the final clustering when increasing the value of K. We continue to increase K as long as the new emerging clusters provide new distinct (and significant) clusters. However, once the optimal K is reached, additional clusters arising will have a nature (the relative frequencies of the observed label) that is indistinct from the ones obtained with optimal K.

Parameter n is the number of cells used in the tessellation, and is related to the unknown strength of the underlying dependence structure. If there is strong dependence in the latent signal, implying large sojourn times for the different states, it would typically be good to have large cells in the tessellations (and thus a rather small n). If the latent signal exhibits a rather short dependency, the sojourn times for the states is typically small. In this case large cells should be avoided (and thus a large n be chosen) since they would typically cover several states and thus not capture the variation in states over time. A way forward to find a solution to the choice of n lies in the random nature of the bagging clustering methods. Observe that for fixed parameters K and n, we perform multiple bootstrap replicates, which provides information about the stability of the final assignments.

After B bootstrap replicates for each site t the relative frequency vector of cluster assignments $(\hat{\pi}_t^1, \ldots, \hat{\pi}_t^K)$, $t = 1, ..., T$, is computed, according to equation (1). For each t, the normalized Shannon entropy

$$\eta(t) = \eta(t, K, n) := -\frac{1}{\log(K)} \sum_{q=1}^{K} \hat{\pi}_t^q \log(\hat{\pi}_t^q), \tag{2}$$

taking values between 0 and 1, indicates how stable the cluster assignment is at site t. A stable assignment for site t corresponds to one large value in the relative frequency vector and the rest close to zero, which makes $\eta(t)$ close to zero. On the other hand if all K states have been assigned about the same number of times for site t then all $\hat{\pi}_t^q$'s are of the same magnitude and the value of (2) close to its maximum value of 1. An illustration of the entropy function $\eta(t)$, $t = 1, \ldots, T$, is given in Fig. 3 (solid black line) for the *BEKM* method applied to the observed data in Fig. 1, with $T = 3000$, $K = 3$ and $n = 75$. As expected, the entropy function is larger close to state-changes in the latent signal. Further, following Secchi et al. (2013) and Abramowicz et al. (2017, 2019) we use the average normalized Shannon entropy

$$\bar{\eta} = \bar{\eta}(K, n) = \frac{1}{T} \sum_{t=1}^{T} \eta(t, K, n), \tag{3}$$

to quantify and compare the overall uncertainty of the clustering method. To minimise the average uncertainty, the value of the parameter n is chosen to be the one that minimizes $\bar{\eta}$, for a given K.

Finally, a technical parameter to be chosen for the algorithm is the number of bootstrap replicates B. It is naturally recommended to choose it as high as possible, to minimise the error in the estimates due to Monte Carlo error. However, in practice, the minimal suitable number of bootstrap replicates can be chosen by pilot studies, which indicate the minimum value necessary to attain stability of the final results.

2.4. Adaptive tessellation

Having run a bagging clustering method once, yields a normalized entropy function, e.g., like in Fig. 3, illustrating the local uncertainty variation in cluster assignments. As already mentioned in Section 2.3, the entropy function typically has larger values close to state-changes of the latent signal and lower values further away from transitions of states. If the borders of the cells in a tessellation could be chosen with a higher degree close to the state-changes instead of just randomly as in the edge and the Voronoi tessellation, we would expect to gain stability and improved performance of the bagging clustering methods.

Utilising the information in the normalized entropy function (2) from an already run bagging clustering method, we construct new partition methods that results in shorter cells where the uncertainty is higher, i.e., there a large variation in the clustering result between the bootstrap replicates is obtained. Analogously, we allow for longer cells where we have lower uncertainty, hence more stable result.

To attain this, we introduce a modified edge and a modified Voronoi tessellation strategy, called adaptive edge and adaptive Voronoi tessellation, respectively. We use the estimated entropy function as the (unnormalized) density from which the random set of points are sampled. These points then become edges or nuclei, depending if edge or Voronoi tessellation strategy is applied. With this adaptive aspect incorporated in the strategies, we tend to obtain a higher number of cells where the entropy is large, which often occurs around transitions of the states of the latent signal.

The sampling process of the points is again related to the theory of Poisson processes. Consider a non-homogeneous Poisson process on the interval $[0, T]$ with intensity function $\lambda(\cdot)$. The distribution of the events in the interval, given that exactly M events occurred, corresponds to a distribution with density proportional to $\lambda(\cdot)$. Using the acceptance-rejection method, we can simulate numbers from the variable with the desired density without having to determine the proportionality constant. For a detailed review of the theory and simulation of non-homogeneous Poisson processes, we refer to Ross (2012). In our case, the discretized version of the algorithm is employed to simulate M sites. For a given

Algorithm 1 Entropy-proportional site sampling.

Input: Total number of sites T; number of points to sample M; estimated entropy function $\eta(t), t = 1, \ldots, T$;
Output: Set $\mathcal{J}_M$ of ordered points

Initialize $k = 0$, $\mathcal{J}_M = \emptyset$, $\mathcal{I} = \{1, ..., T\}$
while $k < n$ **do**
  Simulate $j \sim \mathcal{U}(\{\mathcal{I}/\mathcal{J}_M\})$ and $U \sim \mathcal{U}(0, 1)$, where $\mathcal{U}$ denotes a uniform distribution and / is a set difference operator.
  if $U \leq \eta(j) / \max_{j \in \mathcal{I}/\mathcal{J}_M} \eta(j)$ **then**
    $\mathcal{J}_M = \mathcal{J}_M \cup \{j\}$
    $k = k + 1$
  end if
end while
Sort and return $\mathcal{J}_M$

entropy function $\eta(t)$, $i = 1, \ldots, T$, we generate the set of M sites $\mathcal{J}_M = \{j_1, \ldots, j_M\}$ using entropy proportional sampling. The sampling strategy is described in detail in Algorithm 1. The Voronoi and edge tessellations of size n can then be created utilising the above algorithm, by simulating $M = n$ nuclei or $M = n - 1$ edges, accordingly. Due to the cell construction mechanism, for the adaptive edge tessellation we exclude site T as possible outcome in Algorithm 1.

Some adaptive bagging clustering methods. The adaptive edge/Voronoi tessellation strategy (based on the normalized entropy function of an already run bagging clustering method) can be combined with any of the clustering methods for the local representatives, and in our case leads to four additional bagging clustering methods presented in Table 2. Observe that each method utilizes the entropy function from its non-adaptive counterpart. We employ the entropy function corresponding to the n that leads to the lowest average normalized entropy. We include these four adaptive bagging clustering methods in the simulation study and compare their performance to the other four previously introduced bagging clustering methods in Table 1.

3. A simulation study

In this section we present a simulation study to compare the performance of the introduced eight bagging clustering methods in Tables 1 and 2. The performance is evaluated by the coincidence between the true and estimated hidden state sequences of the latent signal. The simulated categorical data sets are generated using different types of discrete

Table 2
Four adaptive bagging clustering methods obtained by combining the two adaptive tessellation generating mechanisms and the two clustering techniques.

	Clustering	
tessellation	k-means	weighted k-means
adaptive Voronoi	**aBVKM**	**aBVwKM**
adaptive edge	**aBEKM**	**aBEwKM**

time Hidden Markov Models (HMM), both time-homogeneous and non-homogeneous time models. The simulation study presented in this section is part of a larger study where several additional simulated cases with variants of the latent and observed structures have been considered as well as different sample sizes. For brevity, those extended results are presented in the Appendices.

3.1. Simulation setup

The data for the study are obtained by first simulating a sequence of $T = 3000$ states of the latent signal, modelled by a discrete-time Markov chain $\{c_t, t = 1, 2, ..., T\}$ with $K = 3$ states. Given the states of the Markov chain, the (observable) data, being a sequence of categorical variables $l_t \in \{1, 2, 3\}$, $t = 1, 2, ..., T$ is obtained on the basis of specific state-dependent probability distributions. The transition probabilities between the hidden states of the latent signal in two consecutive time steps, are described by transition matrix $\mathbb{P}(t) = [p_{ij}(t)]$, where

$$p_{ij}(t) = P(c_{t+1} = j | c_t = i), \qquad i, j = 1, 2, 3, \quad t = 1, \ldots, T-1.$$

For sake of clarity, we suppress argument (t) in the transition matrix expression if the probabilities do not change over time. For each time step t, given the state c_t of the latent signal, a categorical variable l_t is generated according to a probability mass function determined by a specific row of the emission matrix $\mathbb{E} = [E_{rk}]$. The elements of the emission matrix correspond to conditional emission probabilities, i.e.,

$$E_{rk} = P(l_t = k | c_t = r), \qquad r, k = 1, 2, 3, \quad t = 1, \ldots, T-1.$$

In our study, we consider 27 different examples of HMMs, combining the following:

- nine types of Markov chains (six time-homogeneous and three non-homogeneous time) with distinct transition matrices describing the probabilities of transition between hidden states
- three different types of emission probability matrices

The six homogeneous Markov chains reflect various intensities of the temporal dynamics. The first three examples are based on the symmetric transition matrices constant over time:

$$\mathbb{P}_1 = \frac{1}{100}\begin{bmatrix} 98 & 1 & 1 \\ 1 & 98 & 1 \\ 1 & 1 & 98 \end{bmatrix}, \quad \mathbb{P}_2 = \frac{1}{400}\begin{bmatrix} 398 & 1 & 1 \\ 1 & 398 & 1 \\ 1 & 1 & 398 \end{bmatrix}, \quad \mathbb{P}_3 = \frac{1}{1000}\begin{bmatrix} 998 & 1 & 1 \\ 1 & 998 & 1 \\ 1 & 1 & 998 \end{bmatrix}.$$

The corresponding Markov chains defined by $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ are generating latent sequences where all states have mean sojourn times 50, 200, and 500, respectively. In this way we can compare slowly and rapidly changing transitions of states, by keeping uniform chances of transitions among alternative states. Next, we consider the following three non-symmetric matrices:

$$\mathbb{P}_4 = \frac{1}{2000}\begin{bmatrix} 1960 & 20 & 20 \\ 5 & 1990 & 5 \\ 2 & 2 & 1996 \end{bmatrix}, \quad \mathbb{P}_5 = \frac{1}{1200}\begin{bmatrix} 1188 & 6 & 6 \\ 3 & 1194 & 3 \\ 2 & 2 & 1196 \end{bmatrix},$$

$$\mathbb{P}_6 = \frac{1}{1500}\begin{bmatrix} 1440 & 48 & 12 \\ 10 & 1490 & 0 \\ 5 & 0 & 1495 \end{bmatrix}.$$

The Markov chains defined by $\mathbb{P}_4$, $\mathbb{P}_5$ and $\mathbb{P}_6$ represents cases where the mean sojourn times between the states differ and are given by 50, 200 and 500; 100, 200 and 300; 25, 150 and 300, respectively. The first two models incorporate equally likely transitions to alternative states, while the last one favours transitions to states with lower sojourn times.

Finally, the three non-homogeneous Markov chains are defined by time dependent transition matrices. For each $t \in [1, 2, ..., T]$, we define

Table 3
List of the 27 different types of simulated HMMs with their corresponding transition and emission matrices.

Case	Transition matrix	Emission matrix	Case	Transition matrix	Emission	Case	Transition matrix	Emission matrix
S1.L	$\mathbb{P}_1$	$\mathbb{E}_1$	*S2.L*	$\mathbb{P}_2$	$\mathbb{E}_1$	*S3.L*	$\mathbb{P}_3$	$\mathbb{E}_1$
S1.M	$\mathbb{P}_1$	$\mathbb{E}_2$	*S2.M*	$\mathbb{P}_2$	$\mathbb{E}_2$	*S3.M*	$\mathbb{P}_3$	$\mathbb{E}_2$
S1.H	$\mathbb{P}_1$	$\mathbb{E}_3$	*S2.H*	$\mathbb{P}_2$	$\mathbb{E}_3$	*S3.H*	$\mathbb{P}_3$	$\mathbb{E}_3$
A1.L	$\mathbb{P}_4$	$\mathbb{E}_1$	*A2.L*	$\mathbb{P}_5$	$\mathbb{E}_1$	*A3.L*	$\mathbb{P}_6$	$\mathbb{E}_1$
A1.M	$\mathbb{P}_4$	$\mathbb{E}_2$	*A2.M*	$\mathbb{P}_5$	$\mathbb{E}_2$	*A3.M*	$\mathbb{P}_6$	$\mathbb{E}_2$
A1.H	$\mathbb{P}_4$	$\mathbb{E}_3$	*A2.H*	$\mathbb{P}_5$	$\mathbb{E}_3$	*A3.H*	$\mathbb{P}_6$	$\mathbb{E}_3$
N1.L	$\mathbb{P}_7(t)$	$\mathbb{E}_1$	*N2.L*	$\mathbb{P}_8(t)$	$\mathbb{E}_1$	*N3.L*	$\mathbb{P}_9(t)$	$\mathbb{E}_1$
N1.M	$\mathbb{P}_7(t)$	$\mathbb{E}_2$	*N2.M*	$\mathbb{P}_8(t)$	$\mathbb{E}_2$	*N3.M*	$\mathbb{P}_9(t)$	$\mathbb{E}_2$
N1.H	$\mathbb{P}_7(t)$	$\mathbb{E}_3$	*N2.H*	$\mathbb{P}_8(t)$	$\mathbb{E}_3$	*N3.H*	$\mathbb{P}_9(t)$	$\mathbb{E}_3$

$$\mathbb{P}_7(t) = \frac{T-t}{T-1}\mathbb{P}_3 + \frac{t-1}{T-1}\mathbb{P}_1,$$
$$\mathbb{P}_8(t) = \mathbf{1}_{\{t \le T/2\}}\mathbb{P}_3 + \mathbf{1}_{\{t > T/2\}}\mathbb{P}_1,$$
$$\mathbb{P}_9(t) = \mathbf{1}_{\{t \le T/2\}}\mathbb{P}_4 + \mathbf{1}_{\{t > T/2\}}\mathbb{J}\,\mathbb{P}_4\mathbb{J},$$

where $\mathbb{J}$ is a 3x3 backward identity matrix (exchange matrix) and $\mathbf{1}_{\{\cdot\}}$ is the indicator function. Note that $\mathbb{P}_7$ represents the case where the mean sojourn time of the states starts at 500 when $t = 1$ and then linearly decreases to 50 at the time point $t = T$. The transition matrix $\mathbb{P}_8$ corresponds to the case where the mean sojourn time is 500 for the first half of the time and then abruptly switches to 50 for the second half. Finally, the transition matrix $\mathbb{P}_9$ represents a case where the mean sojourn times for the three states are 500, 200 and 50, respectively, for the first half and then switches to 50, 200, and 500, respectively, for the second half.

We now introduce three types of emission scenarios for the observable categorical labels, designed to have different degrees of separability between the $K = 3$ latent states. The emission matrices for the three types are:

$$\mathbb{E}_1 = \begin{bmatrix} 0.450 & 0.275 & 0.275 \\ 0.275 & 0.450 & 0.275 \\ 0.275 & 0.275 & 0.450 \end{bmatrix}, \quad \mathbb{E}_2 = \begin{bmatrix} 0.50 & 0.25 & 0.25 \\ 0.25 & 0.50 & 0.25 \\ 0.25 & 0.25 & 0.50 \end{bmatrix}, \quad \mathbb{E}_3 = \begin{bmatrix} 0.550 & 0.225 & 0.225 \\ 0.225 & 0.550 & 0.225 \\ 0.225 & 0.225 & 0.550 \end{bmatrix}.$$

The i:th row and j:th column of each matrix gives the probability of categorical label j occurring given the latent variable is in state i. The separability between the states is lowest for $\mathbb{E}_1$, moderate for $\mathbb{E}_2$ and highest for $\mathbb{E}_3$. The resulting variation in distinction of the observed labels expression has a natural effect on performance of any latent state recovery method. Additional emission matrices are considered in Appendix B where more extreme separability scenarios are investigated.

A summary of all simulation scenarios, obtained by combinations of Markov chains and emission probability distributions is presented in Table 3. We introduce the intuitive case names in the first column of Table 3 to facilitate the readability and discussion of the results. Cases "S" and "A" correspond to data generated by time homogeneous HMMs, while cases "N" correspond to data generated by non-homogeneous HMMs. The homogeneous cases "S" and "A" refers to the type of transition matrices used, "S" representing the symmetric cases (equal sojourn times in the latent states) whereas "A" represents the asymmetric cases (unequal sojourn times in the latent states). Furthermore, the letters "L" (low), "M" (moderate) and "H" (high) represent the degree of separability between the states and are coupled to the emission matrices $\mathbb{E}_1$, $\mathbb{E}_2$ and $\mathbb{E}_3$ used, respectively.

3.2. Computational details

We generate 100 data sets (of length $T = 3000$) for each of the 27 HMM cases in Table 3. To ensure that each of the data sets include all three latent states for a non-negligible amount of time, we replace data sets if a state appears less than 1% of the time sequence, i.e., less than 30 times. This happened in cases S3, A1, and N3 where less than 10 data sets had to be regenerated in order to obtain 100 sets. All eight proposed bagging clustering methods are then applied to the simulated data sets using the true number of hidden states, $K = 3$.

For the four methods *BVKM*, *BVwKM*, *BEKM* and *BEwKM*, we consider the following values of n (number of cells in the tessellations): 15, 17, 20, 24, 30, 33, 37, 43, 50, 60, 75, 100, 120, 150, 200, 300 and 400. For each value of n, we use $B = 500$ bootstrap replicates. For each generated data set, and for each of the four methods, the recovered latent signal is then saved for the n that minimizes the average entropy function $\bar{\eta}$. The four adaptive methods *aBVKM*, *aBVwKM*, *aBEKM* and *aBEwKM* are applied with the same n as found optimal for the methods *BVKM*, *BVwKM*, *BEKM* and *BEwKM*, respectively.

All numerical studies are performed using the R programming language (R Core Team, 2019). The average computational time per data set (standard deviation in parentheses) for recovery of a latent signal when run on a 3.5 GHz Intel Core i7 processor with 32 GB ram memory (using one core) for the eight methods *BVKM*, *BVwKM*, *BEKM*, *BEwKM*, *aBVKM*, *aBVwKM*,

Table 4
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Markov processes), is presented for different methods. The number in bold shows for each case the method that has the highest percentage of correctly classified observations. The numbers in parentheses present the standard errors (of the accuracies) of the methods.

Case	*BVKM*	*BEKM*	*BVwKM*	*BEwKM*	*aBVKM*	*aBEKM*	*aBVwKM*	*aBEwKM*
S1.L	71.0(0.62)	46.8(1.01)	72.8(0.55)	72.8(0.54)	71.1(0.59)	47.5(0.98)	**72.8**(0.50)	72.7(0.52)
S1.M	**83.2**(0.33)	74.5(1.64)	82.7(0.40)	82.8(0.40)	82.7(0.31)	73.8(1.59)	82.4(0.38)	82.5(0.37)
S1.H	88.6(0.21)	88.7(0.20)	88.6(0.20)	**88.7**(0.21)	88.2(0.21)	88.2(0.23)	88.2(0.19)	88.3(0.21)
S2.L	87.3(0.67)	76.4(1.08)	87.5(0.63)	87.8(0.65)	87.2(0.58)	77.5(1.04)	87.8(0.52)	**88.0**(0.53)
S2.M	**93.8**(0.29)	93.3(0.29)	93.5(0.31)	93.7(0.29)	93.5(0.26)	92.9(0.27)	93.4(0.27)	93.4(0.26)
S2.H	96.2(0.16)	**96.5**(0.13)	96.2(0.19)	96.2(0.19)	96.3(0.14)	96.2(0.12)	96.1(0.15)	96.1(0.15)
S3.L	93.4(0.64)	87.8(0.66)	92.3(0.70)	92.4(0.61)	**93.4**(0.58)	88.6(0.70)	92.4(0.66)	92.5(0.56)
S3.M	97.0(0.21)	96.3(0.29)	96.4(0.42)	96.6(0.37)	**97.0**(0.17)	96.7(0.19)	96.5(0.35)	96.5(0.33)
S3.H	98.1(0.18)	98.2(0.13)	97.8(0.40)	98.0(0.26)	98.2(0.12)	**98.3**(0.09)	97.8(0.37)	98.0(0.21)
A1.L	84.0(1.12)	88.1(0.50)	75.4(1.13)	76.6(1.07)	85.3(1.04)	**88.7**(0.47)	77.5(1.05)	78.1(1.01)
A1.M	89.7(0.74)	91.8(0.35)	85.0(0.99)	86.1(0.94)	91.7(0.59)	**93.4**(0.31)	87.4(0.89)	87.6(0.84)
A1.H	93.6(0.64)	95.1(0.22)	91.3(0.84)	91.6(0.82)	94.9(0.46)	**96.4**(0.15)	92.7(0.77)	92.9(0.73)
A2.L	85.0(0.90)	79.5(0.87)	83.2(1.01)	84.3(0.89)	**85.6**(0.85)	80.8(0.88)	84.1(0.93)	84.6(0.86)
A2.M	92.3(0.58)	91.5(0.40)	91.8(0.61)	92.4(0.54)	92.5(0.52)	**92.7**(0.33)	92.0(0.53)	92.3(0.49)
A2.H	95.6(0.36)	96.0(0.16)	95.5(0.42)	95.5(0.48)	95.9(0.26)	**96.3**(0.13)	95.6(0.38)	95.5(0.41)
A3.L	80.6(1.13)	84.4(0.45)	76.0(0.98)	77.8(0.92)	81.8(1.09)	**85.2**(0.40)	77.8(0.98)	79.1(0.93)
A3.M	87.7(0.63)	86.9(0.45)	85.2(0.68)	86.1(0.65)	**89.3**(0.56)	89.1(0.37)	86.8(0.61)	87.2(0.60)
A3.H	91.2(0.48)	91.0(0.40)	90.6(0.52)	90.9(0.55)	**92.5**(0.38)	92.5(0.30)	92.0(0.43)	91.9(0.42)
N1.L	78.7(0.66)	62.0(1.20)	79.8(0.56)	80.2(0.52)	79.0(0.63)	63.0(1.14)	80.1(0.52)	**80.3**(0.49)
N1.M	**89.0**(0.29)	85.9(0.90)	88.6(0.36)	88.6(0.34)	88.9(0.29)	85.8(0.80)	88.5(0.33)	88.6(0.28)
N1.H	92.9(0.18)	**93.1**(0.17)	92.8(0.18)	92.9(0.17)	92.9(0.17)	92.8(0.17)	92.9(0.17)	92.9(0.17)
N2.L	78.6(0.71)	69.6(0.86)	78.5(0.76)	78.4(0.80)	79.7(0.62)	70.7(0.89)	**79.9**(0.68)	79.6(0.75)
N2.M	87.5(0.45)	84.3(0.86)	86.3(0.62)	86.1(0.63)	**88.5**(0.36)	85.6(0.73)	87.5(0.51)	87.4(0.55)
N2.H	91.8(0.24)	92.1(0.34)	90.9(0.47)	91.0(0.47)	92.6(0.19)	**92.8**(0.25)	91.9(0.36)	92.0(0.41)
N3.L	89.5(0.53)	80.8(0.93)	88.7(0.57)	89.1(0.54)	**89.9**(0.53)	81.9(0.95)	89.9(0.51)	89.8(0.50)
N3.M	94.3(0.25)	93.0(0.54)	93.7(0.33)	93.9(0.30)	**94.9**(0.23)	93.6(0.41)	94.5(0.25)	94.6(0.23)
N3.H	96.3(0.18)	96.2(0.20)	96.1(0.21)	96.1(0.21)	**96.9**(0.13)	96.8(0.15)	96.7(0.16)	96.7(0.16)

aBEKM and *aBEwKM* is approximately 6 (0.4), 15 (0.6), 6 (0.4), 16 (0.6), 6 (0.5), 16 (0.7), 7 (0.4) and 17 (0.7) minutes, respectively.

3.3. Results

The performance of the eight proposed bagging clustering methods is quantified by the clustering accuracy, here quantified as the percentage of correctly identified states of the latent signal. The clustering accuracy, averaged over the 100 data sets is presented in Table 4, for each of the eight methods. The data sets are the same for each of the methods, and hence pairwise comparisons of accuracy can be made as a complement to average performance measures.

In general, we see that all methods perform better when the separability between the states increases. In the homogeneous symmetric cases (S), we also note that the performance of the methods increases whenever the mean sojourn time increases. For the asymmetric cases (A) the methods typically identify the latent sequences more easily when differences in mean sojourn times of the states are small (A2). Among the non-homogeneous cases (N) the latent sequences of case N3 are more correctly recovered compared to cases N1 and N2.

From Table 4 it can also be concluded that the adaptive methods generally improve the classification accuracy compared to their non-adapted versions. Fig. 4 visualizes how each adaptive bagging clustering method compares to its non-adaptive counterpart with respect to percentage of correctly classified observations. More specifically, for each of the 100 data sets of a simulated scenario, we calculated the pairwise differences in classification accuracy between the adaptive and its corresponding non-adaptive bagging clustering method, and present in Fig. 4 the average difference together with its corresponding 95% confidence interval. The asymmetric and non-homogeneous cases gain most from using adaptive methods, while the non-adaptive methods tend to work slightly better for the symmetric cases when the separability between states is high.

There are small differences in performance between the edge and Voronoi tessellation strategies when weighted clustering (*wKM*) is used, whereas substantial differences are noted using the unweighted clustering *KM* (cf. Table 4 and Fig. A.7

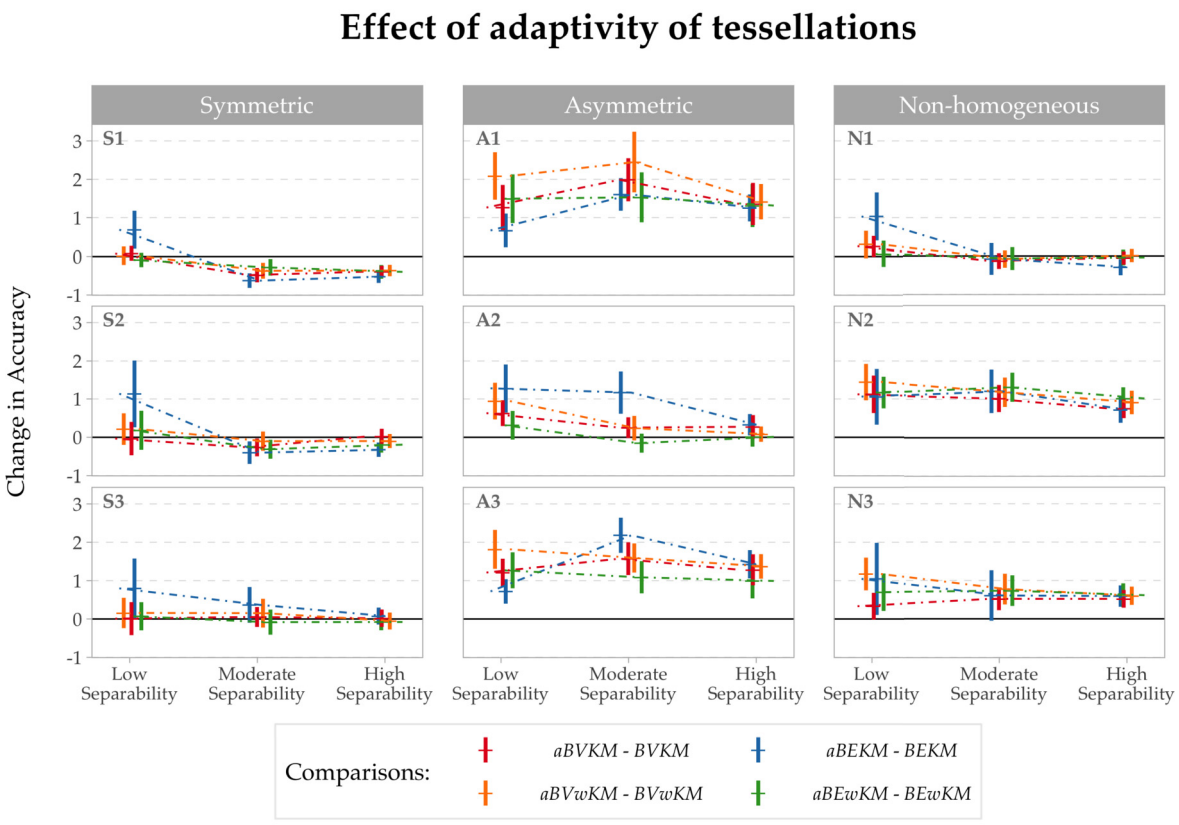

Fig. 4. Average pairwise differences in classification accuracy for the bagging clustering methods with and without adaptively formed tessellations. Each panel corresponds to different scenarios. Horizontal markers present the average difference where the average is calculated over the 100 replicates. Vertical bars indicate 95% confidence intervals for the mean difference.

in Appendix A), especially when the separation between states is low. In particular, the edge tessellation methods with *KM* work better in cases where there are large differences between the mean sojourn times of the states, e.g., in cases A1.L and A3.L. In cases where the mean sojourn times of the states are more equal, the Voronoi tessellation methods tend to work better, see, e.g., cases S1.L and N1.L. A reason could be that the edge strategy, which allows a larger variation of the simulated cell lengths, makes it more likely to capture states with both long and short sojourn times, while the Voronoi strategy is more suitable when the sojourn times are more equal.

To see how incorporating weighted clustering affects the performance of bagging clustering methods, we illustrate in Fig. 5 how each *wKM* method compares to its non-weighted counterpart in terms of percentage of correctly classified observations. The construction of the figure is made in the same way as Fig. 4. The edge tessellations methods, as opposed to the Voronoi tessellation methods, are substantially influenced by the choice of clustering. The edge tessellations methods benefits from using *wKM* in the symmetric and non-homogeneous cases whereas it is preferable to use *KM* in the asymmetric cases, especially in the cases with low separability. The lower performance of *wKM* in the asymmetric cases may be related to the fact that it emphasises states with longer sojourn times in the clustering procedure, and therefore may fail to capture the underlying structure of the states with shorter sojourn times.

To summarise, there is no method that always performs better than all the others. The adaptive methods tend to work better for asymmetric and non-homogeneous cases. For symmetric and non-homogeneous cases when the separability between states is low the Voronoi tessellations tend to work better than edge tessellations while the usage of a weighted k-means strategy tends to work better than its non-weighted counterpart.

4. Composite methods

As seen in the previous section, there is no unanimous method that in all cases outperforms the others. It depends on the underlying data generating mechanism which for a given data set at hand usually is unknown. Therefore, it would for the user be helpful if an automated decision rule could pick the best of the methods to use for any given case. We here propose four such data driven decision rules based on the previous eight bagging clustering methods and their average normalized entropies. We construct the four composite methods in such a way that they select the reconstructed signal from one of the two underlying methods (weighted and non-weighted clustering methods) with lowest average normalized entropy. The introduced methods are:

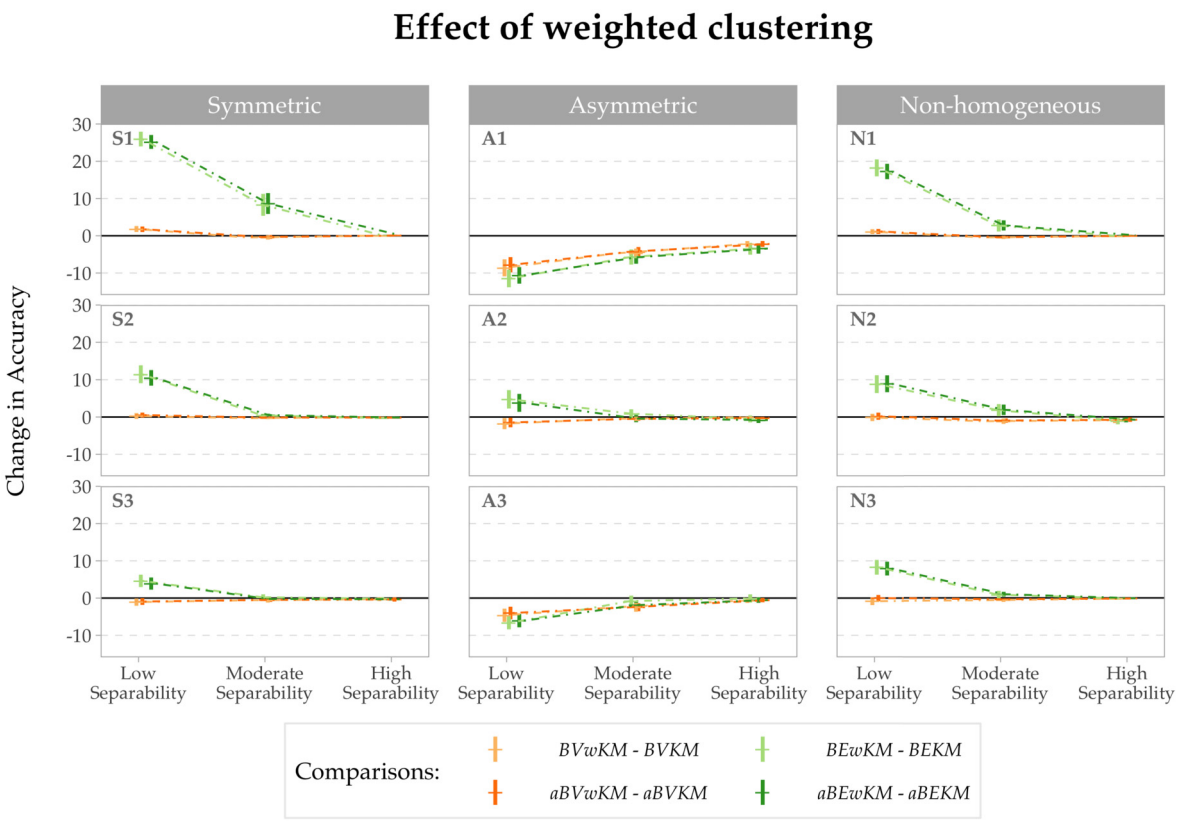

Fig. 5. Average pairwise differences in classification accuracy for the bagging clustering methods with and without weighting incorporated in the clustering of the local representatives. Each panel corresponds to different scenarios. Horizontal markers present the average difference where the average is calculated over the 100 replicates. Vertical bars indicate 95% confidence intervals for the mean difference.

$\boldsymbol{C\{BV\}}$ selecting the best of *BVKM* and *BVwKM*
$\boldsymbol{C\{BE\}}$ selecting the best of *BEKM* and *BEwKM*
$\boldsymbol{C\{aBV\}}$ selecting the best of *aBVKM* and *aBVwKM*
$\boldsymbol{C\{aBE\}}$ selecting the best of *aBEKM* and *aBEwKM*

Note that each of the four composite methods makes a choice only between the methods that use the same random tessellation mechanism. Each tessellation method generates different random variability (cf. Section 2.2.1), which is captured in the average normalized entropy together with the uncertainty that is associated with the latent signal. Since we cannot separate the two sources of variability in the entropy measure and the underlying cell sampling mechanisms differ in variability between the *BV*, *BE*, *aBV* and *aBE* tessellation methods, we choose not to combine the methods across tessellation schemes.

4.1. Performance of the composite methods

The performance of the composite methods are evaluated on the same simulated data sets as in Section 3, by investigating their (average) clustering accuracy. They are also compared to the worst and best (in average) uncombined bagging clustering methods (cf. Table 4) for each of the 27 cases.

The average clustering accuracy for all 27 cases is presented in Table 5. For each case and method, the accuracy marked with an asterisk indicates where a significant difference (at the 0.05 level) is obtained when testing equality of the average clustering accuracy between the method and the best uncombined bagging clustering method, using a two-tailed paired t-test. Table 5 shows that the composite methods based on the adaptive methods work at least as good as their corresponding non-adaptive versions and generally better in the asymmetric and non-homogeneous cases (see also Fig. A.8 in Appendix A for a graphical comparison).

From Table 5 we further conclude that all composite methods work equally well or (much) better than the worst uncombined method. Moreover, $C\{aBE\}$ is on par with the best uncombined method for all cases except cases S1.L and N1.L, where it has substantially lower percentage of correctly classified observations. $C\{aBV\}$ is on par with the best uncombined

Table 5
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Markov processes), is presented for the four composite methods, the worst and the best uncombined method from Table 4 as well as the Viterbi algorithm applied on the resulting HMM models trained by the EM algorithm with both a random and optimum (only applied to the homogeneous cases) initialization strategy. The number in bold shows for each case the method, among the composite methods, that has the highest classification accuracy. The asterisk (*) indicates the methods where a significance difference (on 0.05 level) is detected when comparing the methods to the best uncombined method, using a two-tailed paired t-test.

Case	$C\{BV\}$	$C\{BE\}$	$C\{aBV\}$	$C\{aBE\}$	Worst uncombined method	Best uncombined method	*Viterbi* (random)	*Viterbi* (optimum)
S1.L	72.8	61.6*	**72.8**	65.1*	46.8*	72.8	62.7*	63.8*
S1.M	82.7	**82.8**	82.4*	82.5*	73.8*	83.2	79.7*	79.6*
S1.H	88.6*	**88.7**	88.2*	88.3*	88.2*	88.7	87.5*	87.6*
S2.L	87.9	**88.7***	88.0	88.5	76.4*	88.0	84.2*	87.1*
S2.M	93.5*	**93.8**	93.5*	93.6	92.9*	93.8	94.1	94.1
S2.H	96.2*	**96.3***	96.2*	96.2*	96.1*	96.5	96.8*	96.8*
S3.L	92.9	93.3	93.0	**93.4**	87.8*	93.4	84.0*	94.6
S3.M	96.4	96.8	**96.9**	96.8*	96.3*	97.0	94.2*	97.6*
S3.H	97.8	98.2	98.1	**98.2**	97.8	98.3	96.2*	98.8*
A1.L	77.7*	88.1	80.9*	**88.3**	75.4*	88.7	80.2*	87.3
A1.M	86.1*	91.2*	89.3*	**92.5**	85.0*	93.4	92.5	94.7*
A1.H	91.3*	94.3*	93.4*	**96.0**	91.3*	96.4	96.8	97.3*
A2.L	83.9*	86.0	85.5	**86.2**	79.5*	85.6	83.6	87.6*
A2.M	91.9	93.0	92.3	**93.2***	91.5*	92.7	94.0*	94.2*
A2.H	95.5	96.0*	95.8*	**96.2**	95.5	96.3	96.9*	96.8*
A3.L	77.9*	83.8*	80.7*	**84.3***	76.0*	85.2	80.8*	84.7
A3.M	86.0*	88.0*	88.4*	**89.1**	85.2*	89.3	90.4	91.4*
A3.H	90.6*	91.6*	**92.5**	92.5	90.6*	92.5	94.4*	94.4*
N1.L	79.8	78.0*	**79.9**	78.9*	62.0*	80.3	74.5*	
N1.M	88.6	88.6*	**88.7**	88.6*	85.8*	89.0	88.0*	
N1.H	92.8*	**92.9**	92.9*	92.9	92.8*	93.1	92.7*	
N2.L	78.4*	79.2	80.1	**80.2**	69.6*	79.9	75.2*	
N2.M	86.3*	86.2*	87.8*	**88.0**	84.3*	88.5	87.9*	
N2.H	90.9*	91.0*	**92.5**	92.3	90.9*	92.8	92.7	
N3.L	88.9	89.4	**90.3**	89.9	81.0*	89.9	80.5*	
N3.M	93.7*	93.9*	**94.7**	94.7	93.1*	94.9	94.6	
N3.H	96.0*	96.2*	**96.8**	96.8	96.0*	96.9	96.9	

method except for most of the asymmetric cases where it works less well. Hence, the overall performance of both adaptive composition methods is good.

The effect of the tessellation type (edge or Voronoi) is illustrated in Fig. 6. The clustering accuracy is higher for the edge strategy in most of the asymmetric cases, especially so for the low separability cases. The Voronoi strategy works substantially better for the symmetric case with low separability (S1.L). For all the remaining cases the effect of tessellation strategy is negligible.

4.2. Comparison with Viterbi algorithm

As a reference, we also compare the results of the composite methods to the standard HMM methods, using an EM algorithm and the Viterbi algorithm to recover the latent signal (for the time homogeneous HMMs), due to the nature of the simulated data sets. The Viterbi algorithm is applied to the estimated HMM found by a version of an EM algorithm (both implemented in the R package `hmm.discnp`: Turner, 2020). We consider two ways of initializing the iterative EM algorithm with starting values of the parameter estimates. The first uses 300 random starts (called random), and the second starts with the true HMM parameters of the data generating mechanism (called optimum), which in practice typically is unknown. In the former case (following the default setting of `hmm.discnp` package) for each random start, the transition matrix is initiated as a row normalised matrix filled with uniform random numbers on the unit interval. The emission probabilities are initiated using normalised inverse logit transformations applied to randomly selected standard normal numbers. For each of the random starts, applied to the EM algorithm, the likelihood of the observed sequence is calculated and the model

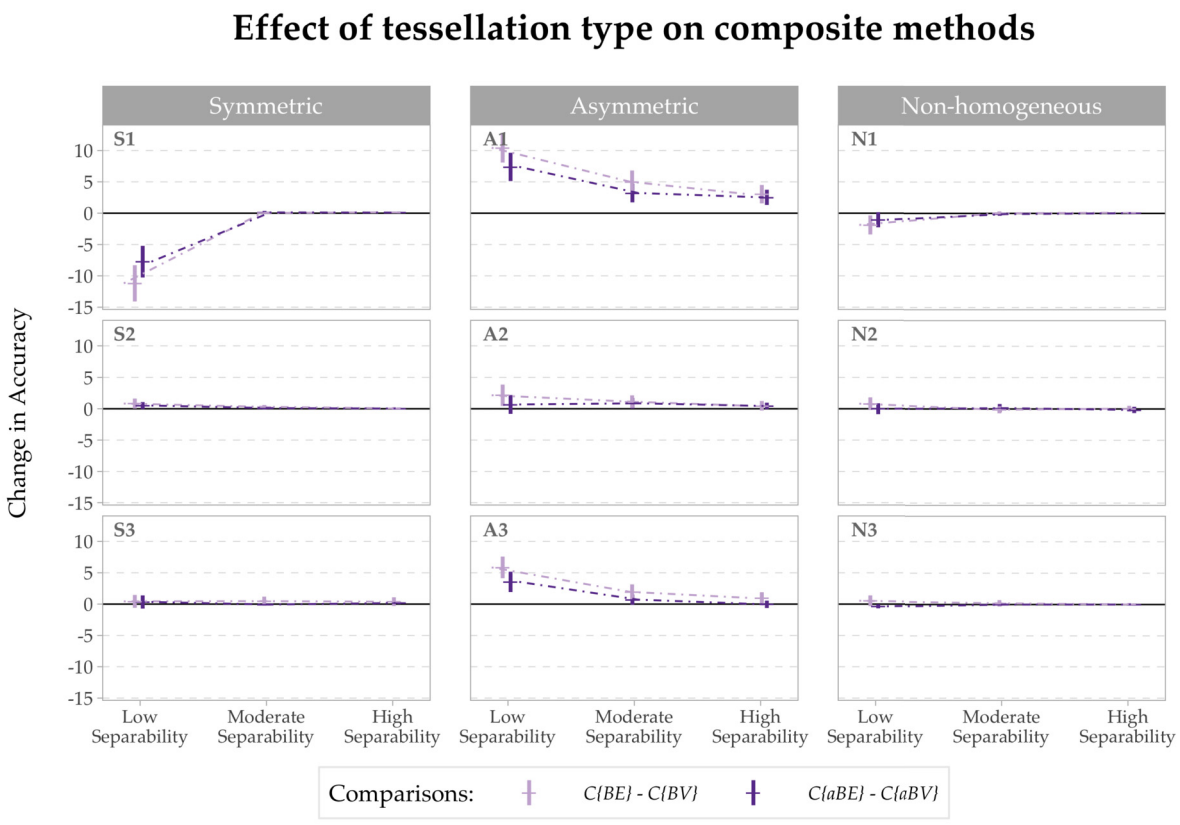

Fig. 6. Pairwise differences in accuracy for the composite methods with respect to tessellation type incorporated. Each panel corresponds to different scenarios. Horizontal markers present the average difference where the average is calculated along all 100 replicates created. Vertical bars indicate 95% confidence intervals for the mean.

that maximises the likelihood chosen as the final model. The optimum initialisation alternative is added as a reference to facilitate understanding the convergence of the EM algorithm to the (global) optimum and the sufficiency of 300 random starts to reach it. For the non-homogeneous HMM cases the EM algorithm that we use is not originally designed to handle such cases and therefore we cannot provide the optimum initialization.

The average number of iterations per data set (standard deviation in parentheses) required for the EM algorithm to converge using optimum initialization (only computed over the homogeneous cases) and 300 random starts is approximately 20 (30) and 400 (370), respectively. Moreover, the average computational time per data set (standard deviation in parentheses) for recovery of a latent signal when run on a 3.5 GHz Intel Core i7 processor with 32 GB ram memory (using one core) with use of the optimum and random approach is approximately < 1 (< 0.1) and 11 (5.5) minutes, respectively (cf. computational times of uncombined methods in Section 3.2).

The Viterbi algorithm (and the EM algorithm), utilizes knowledge of the true parametric model of the transition mechanism, i.e., the fact that the latent states follow a Markov Chain. It can therefore be expected to yield higher classification accuracy than the non-parametric composition methods for the homogeneous HMMs. It is confirmed in Table 5 and Appendix B where, in general, the Viterbi (optimum) algorithm works equally well or better than the non-parametric composite methods for the homogeneous HMMs. Interestingly, the Viterbi (optimum) algorithm performs worse than the composite methods for cases where the separability of the states is low. For the non-homogeneous HMMs we see that in general the combined methods tend to outperform the Viterbi (random) algorithm, which is not surprising given the incorrect model assumptions made. The Viterbi algorithm with 300 random starts performs most of the times worse than the best uncombined method. Notwithstanding, even if the Viterbi (random) performs poorly in some cases, the performance of the Viterbi (optimal) shall stand as a measure of how high accuracy we potentially can achieve with the Viterbi (random) for time homogeneous HMM, by using, e.g., more random starts or some other initialisation strategy. In practice, this would be the only remedy, as the true underlying generating mechanism is unknown, and the Viterbi (optimal) can not be applied.

5. Discussion and conclusions

We have studied a non-parametric bagging clustering framework to reconstruct the (slowly varying) latent signal, from observed dependent discrete random variables generated by latent state-specific probability distributions. The data are observed along a one-dimensional discrete time domain. New and existing bagging clustering methods are compared through a

simulation study presented in Sections 3-4, Appendix B and Appendix C. In particular we propose new ways of constructing the bootstrap samples via edge tessellations/partitions and adaptive tessellations of the time domain and compare with the commonly used methods based on Voronoi tessellations. We also construct composite methods that through an automated entropy-based rule selects the "best" among some (uncombined) bagging clustering methods.

The simulation study illustrates that none of the uncombined methods is superior to the others all the time. It depends on the data generating mechanism which for the user typically is unknown. Moreover, methods based on adaptive data-driven tessellations tend to work better than their non-adaptive versions. Overall, the composite method based on adaptive edge tessellations, $C\{aBE\}$, is robust to different data generating mechanisms and generally works equally well or better than the other methods. The exceptions occur for the "symmetric cases" where the mean sojourn times of the latent states are equal combined with low separability of the latent states. For these cases the composite method based on adaptive Voronoi tessellations $C\{aBV\}$ performs better. Note that the degree of separability of the latent states is given not only by how distinct the state-specific probability distributions are but also by the size of the mean sojourn times, see further discussion in Appendix B.4. For the "asymmetric cases" where the mean sojourn times of the states differ, $C\{aBE\}$ works substantially better. In the light of the above, we would therefore recommend to use the composite method based on adaptive edge tessellations, $C\{aBE\}$, unless there is prior knowledge about the latent states, having similar size of their mean sojourn times and low separability. In this case the composite method based on adaptive Voronoi tessellations would be more appropriate.

A comparison with classical parametric methods for HMM, using the Viterbi and EM algorithms, shows that it often yields significantly higher clustering accuracy compared to the non-parametric composite methods for the homogeneous HMMs. This is to be expected since it utilizes knowledge of the true parametric model, which in practice is rarely known. Surprisingly though, the Viterbi algorithm has significantly lower clustering accuracy for the symmetric cases with short mean sojourn times, especially when the separability between states is low. These comparisons have been made using the true parameter values of the data generating mechanism to initialize the algorithms, which cannot be applied in practice. When the algorithms instead were initialized using 300 random starts, performance was in general worse than the best uncombined bagging clustering method, including the non-homogeneous HMM cases where incorrect model assumptions are made for the Viterbi and EM algorithms. The results illustrate the potential of the Viterbi algorithm and the corresponding practical challenges, in the light of the non-parametric bagging clustering methods.

In the simulation study, for the adaptive methods, we used the same number of segments n to form the adaptive tessellations as was found optimal for their non-adaptive counterpart. That was due to computational reasons but also due to the fact that preliminary analyses showed good results (close to optimal) using the same n. In principle slight improvements might be possible by searching for an optimal n also for the adaptive tessellations, but it comes with additional computational cost. Moreover, the adaptive tessellations might, instead of being based on the local entropy directly, profit from being formed based on some transformation of the local entropy, to either accentuate or reduce the importance of the local extreme behaviour.

In our work, we have focused on methods for data observed along a one-dimensional (discrete) time domain. However, the methods based on Voronoi tessellations can be straightforwardly generalized to higher dimensions, which is not the case for the methods based on edge tessellations. Hence, extensions of our methods to more complex domains are an interesting continuation of current work. Further, the k-means clustering method applied to frequency vectors, used in this paper is one of multiple strategies possible, and our methodology can be extended beyond this limitation. For example, in Abramowicz et al. (2019) we have studied the bagging Voronoi methodology as part of a two step procedure, where the labels carry additional information that could be used in the clustering of representatives.

In summary, our study elucidates that non-parametric bagging clustering methods show great potential to recover latent signals from observed dependent categorical data, for many different data generating mechanisms. Further, the introduced novel mechanisms can also be generalised to other inferential problems and more complex data types.

Acknowledgements

This work was supported by the Swedish Research Council (Project id 340-2013-5203).

Appendix A. Additional figures

Additional figures, discussed in Sections 3 and 4, are given below.

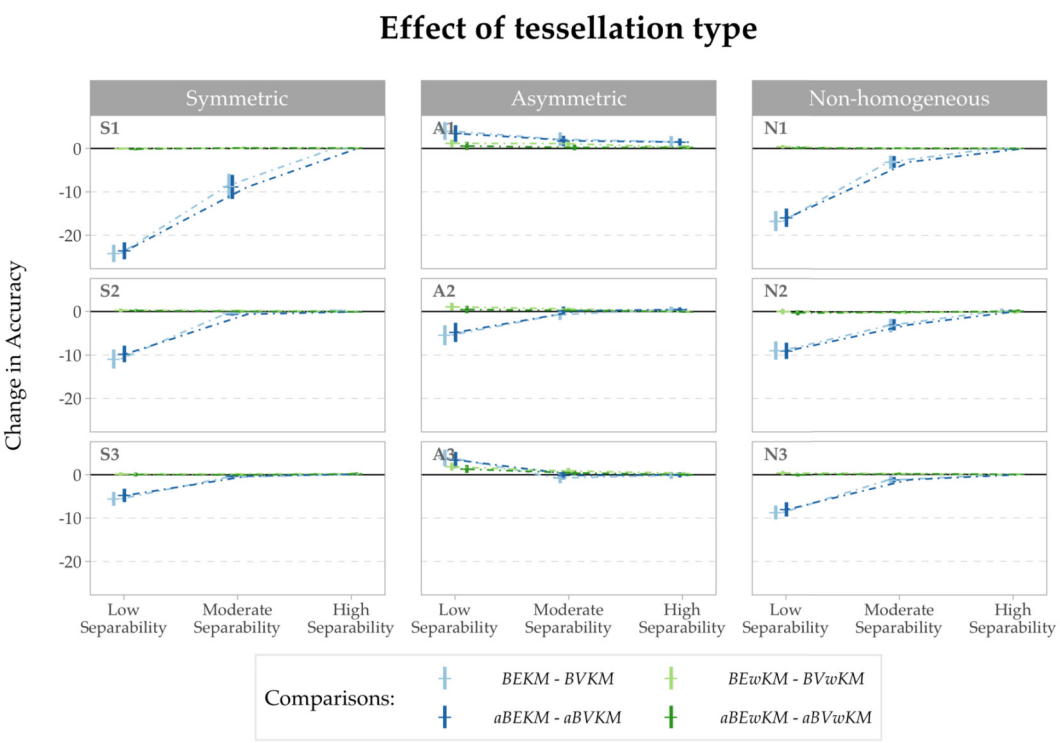

Fig. A.7. Pairwise differences in accuracy for the clustering methods with respect to tessellation type incorporated. Each panel corresponds to different scenarios. Horizontal markers present the average difference where the average is calculated along all 100 replicates created. Vertical bars indicate 95% confidence intervals for the mean.

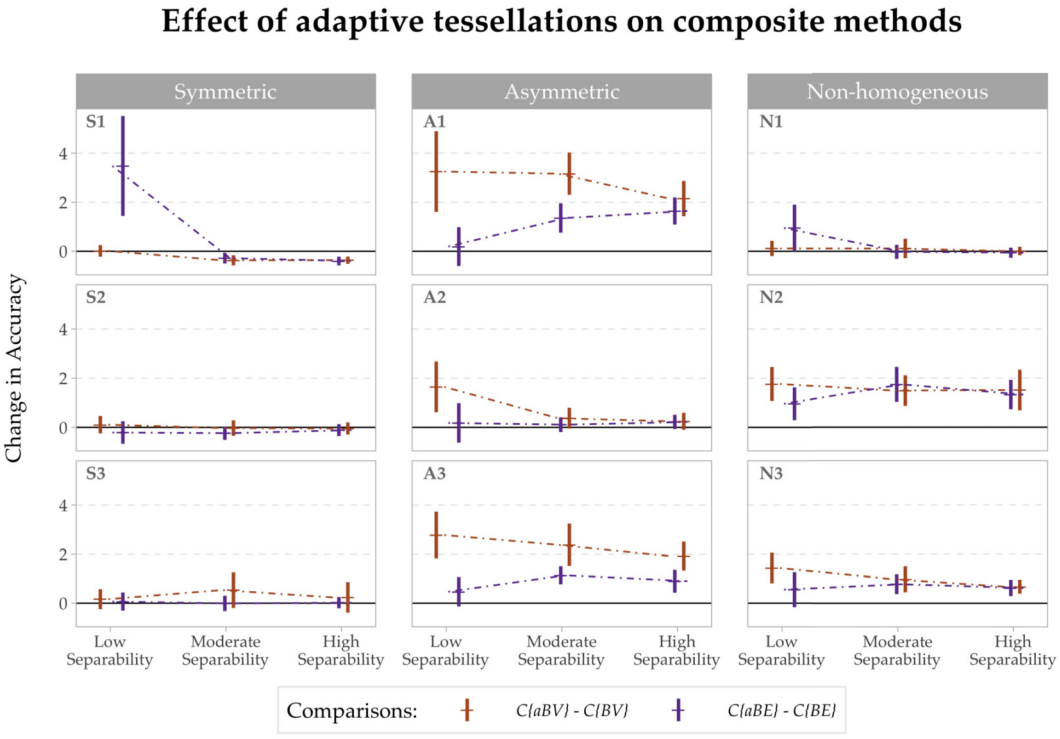

Fig. A.8. Pairwise differences in accuracy for the composite methods with and without adaptivity of tessellation incorporated. Each panel corresponds to different scenarios. Horizontal markers present the average difference where the average is calculated along all 100 replicates created. Vertical bars indicate 95% confidence intervals for the mean.

Appendix B. Additional analyses of an extended simulation study

In this section we present the setup, results and conclusions of an extended simulation study, complementing the simulation study presented in Sections 3-4. In particular, besides studying the performance of the introduced methods for different types of homogeneous and non-homogeneous discrete time Hidden Markov Models, we here also study the methods' performance for different sojourn time distributions, so called Hidden Semi-Markov Models, (Appendix B.1), sample sizes (Appendix B.2), and emission probabilities having both different number of emission labels (Appendix B.3) and different types of separability between states (Appendix B.4). All results are based on 100 realisations of length $T = 3000$ for each case if not otherwise specified. In general, the results and conclusions of the extended study do not significantly deviate from the ones presented in Section 3 and 4. The composite method based on adaptive edge tessellations, $C\{aBE\}$, continues to be a robust choice that in most cases performs in par with or better than the other bagging clustering methods. Below we focus attention on performance of the composite methods and present performance of the uncombined methods in Appendix C.

B.1. Different sojourn time distributions

In addition to the basic assumption of discrete-time Markov chains that the sojourn time in each state is geometrically distributed, we here also consider discrete-time semi-Markov chains with sojourn times that are Poisson- (Po) and lognormal distributed (logN) (rounded to the nearest integer). For a given state, the sojourn time is set such that all three distribution have the same mean (μ) while the variances are given by $(\mu - 1)\mu$, μ and $(exp(1) - 1)\mu^2$ for the geometric-, Poisson- and lognormal distribution, respectively. (Note that, for all $\mu > 2$, the lognormal distribution has the largest variance while the Poisson distribution has the smallest variance.)

The performance of the composite methods applied to the three cases $S2.M$, $A1.M$ and $N2.M$ having different sojourn time distributions are presented in Table B.6 (see Table C.10 in Appendix C for performance of uncombined methods). The results of the composite methods show us that the general trends remain the same regardless of the sojourn times distribution, that is, that an (adaptive) edge tessellation strategy is to prefer in the asymmetric cases, that the adaptive methods perform better than their non-adaptive counterparts in the non-homogeneous cases, and that there is a little difference between the methods in the symmetric cases with the exception that the non-adaptive methods seems to perform slightly better in the case of Poisson distributed sojourn times. Moreover, we also note that all methods, in the non-homogeneous cases, have a substantially lower average classification accuracy when sojourn times are Poisson distributed and that the Voronoi based strategies, in the asymmetric cases, performs worse when sojourn times are lognormally distributed. A comparison with the Viterbi algorithm (random) shows that it performs in par with the best uncombined method in all cases except in the non-homogeneous cases and in the asymmetric case with lognormally distributed sojourn times where it performs significantly worse.

B.2. Different sample sizes

Here we study how the bagging clustering methods perform for different lengths T (of size 250, 500, 1000, 1500, 2000 and 3000) of the generated sequence, i.e., sample sizes. We consider the following three Markov chains defined by the transition matrices:

Table B.6
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Semi-Markov processes), is presented for the four composite methods, the worst and the best uncombined method from Table C.10 in Appendix C as well as the Viterbi algorithm applied on the resulting HMM models trained by the EM algorithm with both a random and optimum (only applied to the homogeneous cases) initialization strategy. The number in bold shows for each case the method, among the composite methods, that has the highest classification accuracy. The asterisk (*) indicates the methods where a significance difference (on 0.05 level) is detected when comparing the methods to the best uncombined method, using a two-tailed paired t-test.

Case	$C\{BV\}$	$C\{BE\}$	$C\{aBV\}$	$C\{aBE\}$	Worst uncombined method	Best uncombined method	*Viterbi* (random)	*Viterbi* (optimum)
S2.M	93.5*	**93.8**	93.5*	93.6	92.9*	93.8	94.1	94.1
S2.M.logN	92.9	**93.0**	92.9	93.0	91.0*	93.0	92.8	93.6*
S2.M.Po	**94.9**	94.9	93.9*	93.2*	92.6*	94.9	95.1	95.1
A1.M	86.1*	91.2*	89.3*	**92.5**	85.0*	93.4	92.5	94.7*
A1.M.logN	83.3*	91.6*	87.1*	**93.0**	82.0*	93.5	90.7*	94.1*
A1.M.Po	85.4*	91.8*	88.9*	**92.9***	84.6*	93.2	94.1	95.1*
N2.M	86.3*	86.2*	87.8*	**88.0**	84.3*	88.5	87.9*	
N2.M.logN	86.3*	86.1*	**87.4***	87.4	83.7*	87.9	86.8*	
N2.M.Po	78.4*	78.9*	**82.5***	80.1*	78.4*	85.4	83.8*	

Table B.7
The average classification accuracy in percentage, based on 100 realisations (of different lengths T) generated from different cases (Markov processes), is presented for the four composite methods, the worst and the best uncombined method from Table C.11 in Appendix C as well as the Viterbi algorithm applied on the resulting HMM models trained by the EM algorithm with both a random and optimum (only applied to the homogeneous cases) initialization strategy. The number in bold shows for each case the method, among the composite methods, that has the highest classification accuracy. The asterisk (*) indicates the methods where a significance difference (on 0.05 level) is detected when comparing the methods to the best uncombined method, using a two-tailed paired t-test.

Case	$C\{BV\}$	$C\{BE\}$	$C\{aBV\}$	$C\{aBE\}$	Worst uncombined method	Best uncombined method	*Viterbi* (random)	*Viterbi* (optimum)
S4.E5.T250	80.5*	81.2	80.9	**81.3**	77.1*	81.6	74.3*	78.5*
S4.E5.T500	82.2*	81.7*	**82.3***	81.7*	79.6*	82.8	78.0*	79.3*
S4.E5.T1000	**83.8**	83.5	83.4*	83.3*	82.5*	84.1	81.6*	81.6*
S4.E5.T1500	**83.6***	83.4	83.1*	83.3*	81.9*	83.8	81.5*	81.4*
S4.E5.T2000	**84.2**	83.0*	83.7*	82.7*	82.6*	84.2	82.4*	82.4*
S4.E5.T3000	**84.0***	82.4*	83.2*	83.1*	80.8*	84.1	82.4*	82.4*
A4.E5.T250	75.9*	81.1	77.9*	**81.5**	74.5*	81.2	71.1*	80.7
A4.E5.T500	78.0*	81.6	80.7*	**82.6**	77.9*	83.0	82.2	84.8*
A4.E5.T1000	81.2*	82.0*	82.7*	**82.9**	81.1*	83.5	85.4*	85.6*
A4.E5.T1500	82.2*	82.9*	**83.4**	83.4	82.1*	83.7	85.8*	86.1*
A4.E5.T2000	82.8*	83.5*	**84.0**	83.8	82.8*	84.1	86.7*	86.6*
A4.E5.T3000	83.2*	83.9	84.1	**84.5**	83.2*	84.3	87.1*	87.1*
N4.E5.T250	76.7*	76.0*	**78.3**	77.4*	73.9*	78.6	70.4*	
N4.E5.T500	79.9*	79.7*	**80.4***	80.2*	76.7*	81.3	77.1*	
N4.E5.T1000	80.9*	80.1*	**81.4***	81.2*	79.7*	82.4	80.2*	
N4.E5.T1500	80.6*	80.7*	**81.1***	81.0*	80.6*	82.2	80.7*	
N4.E5.T2000	82.2*	81.7	**82.4***	81.8	81.2*	82.5	80.6*	
N4.E5.T3000	81.9	81.9	**82.1**	81.9	81.3	82.5	80.9*	

$$\mathbb{P}_{10} = \frac{1}{40}\begin{bmatrix} 38 & 1 & 1 \\ 1 & 38 & 1 \\ 1 & 1 & 38 \end{bmatrix}, \quad \mathbb{P}_{11} = \frac{1}{200}\begin{bmatrix} 180 & 10 & 10 \\ 5 & 190 & 5 \\ 2 & 2 & 196 \end{bmatrix},$$

$$\mathbb{P}_{12}(t) = \mathbf{1}_{\{t \le T/2\}}\mathbb{P}_1 + \mathbf{1}_{\{t > T/2\}}\mathbb{P}_{13},$$

where

$$\mathbb{P}_{13} = \frac{1}{20}\begin{bmatrix} 18 & 1 & 1 \\ 1 & 18 & 1 \\ 1 & 1 & 18 \end{bmatrix},$$

and the emission matrix:

$$\mathbb{E}_5 = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}.$$

The Markov chain defined by $\mathbb{P}_{10}$ represents a symmetric homogeneous case with mean sojourn times 20 (named *S4.E5*), $\mathbb{P}_{11}$ corresponds to an asymmetric homogeneous case with mean sojourn times 10, 20, and 50 (named *A4.E5*), and finally $\mathbb{P}_{12}$ is a time non-homogeneous case with mean sojourn times 20 for the first half of the time and then 10 for the second half (named *N4.E5*). Note that the mean sojourn times between the states are much shorter here than those in Section 3, as a result to ensure that even small sample size data sets include all latent states for a non-negligible amount of time. In order to obtain reasonable estimates of the underlying structure when we have short sojourn times we need to consider an emission matrix with relatively high degree of separability. Note that $\mathbb{E}_5$ has a higher degree of separability than those studied in Section 3. Moreover, a natural effect of entropy is that it approaches 0 as n (the number of cells in the tessellation) approaches the sample size (as the uncertainty in the final labelling decreases). Consequently, the uncombined methods have been applied for different values of n depending on the sample size. For the sample sizes: 250, 500, 1000, 1500, 2000 and 3000 the following maximum values of n have been used: 33, 60, 120, 200, 300 and 400, respectively (see Section 3.2 for a complete list of the values of n used).

The performance of the composite methods applied to the three cases for the different considered sample sizes are presented in Table B.7 (see Table C.11 in Appendix C for performance of uncombined methods). The results of the composite methods show us that the accuracy (percentage of correctly classifying the observations) increases with the sample size

Table B.8
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Markov processes), is presented for the four composite methods, the worst and the best uncombined method from Table C.12 in Appendix C as well as the Viterbi algorithm applied on the resulting HMM models trained by the EM algorithm with both a random and optimum (only applied to the homogeneous cases) initialization strategy. The number in bold shows for each case the method, among the composite methods, that has the highest classification accuracy. The asterisk (*) indicates the methods where a significance difference (on 0.05 level) is detected when comparing the methods to the best uncombined method, using a two-tailed paired t-test.

Case	$C\{BV\}$	$C\{BE\}$	$C\{aBV\}$	$C\{aBE\}$	Worst uncombined method	Best uncombined method	*Viterbi* (random)	*Viterbi* (optimum)
S2.E6.d2	79.4*	71.3*	**81.0**	73.1*	60.3*	81.5	63.2*	79.4
S2.E2.d3	93.5*	**93.8**	93.5*	93.6	92.9*	93.8	94.1	94.1
S2.E7.d4	90.4	91.0	**91.1**	90.9	78.6*	91.1	90.2	91.2
A1.E6.d2	74.9*	84.7	77.7*	**86.2**	74.6*	85.0	77.4*	88.2*
A1.E2.d3	86.1*	91.2*	89.3*	**92.5**	85.0*	93.4	92.5	94.7*
A1.E7.d4	82.2*	**89.7**	85.9*	89.5	79.0*	90.1	84.5*	91.6*
N2.E6.d2	71.3*	67.3*	**73.2**	69.5*	62.5*	73.9	63.1*	
N2.E2.d3	86.3*	86.2*	87.8*	**88.0**	84.3*	88.5	87.9*	
N2.E7.d4	82.4*	82.1*	83.4	**83.5**	71.0*	83.4	81.0*	

and that it levels off when the sample size reaches approximately 1000 at $\sim 81\%$, 82% and 83% accuracy in the non-homogeneous, asymmetric and symmetric cases, respectively. Thus, increasing the sample size beyond 1000 in these cases does not improve the accuracy notably. Also, the general conclusions remain the same for small sample sizes, that is, that the (adaptive) edge tessellation strategy is preferable in the asymmetric cases, that the adaptive methods perform better in the non-homogeneous cases, while there is little difference between the methods in the symmetric cases. However, for larger sample sizes we see that performance of the methods are about the same regardless of the case. The result of the Viterbi algorithm (both random and optimum) shows that it, in comparison with the best uncombined methods, performs worse in the symmetric and non-homogeneous cases regardless of the sample size. In the asymmetric cases it however performs significantly better for larger sample sizes.

B.3. Different number of emission labels

Here we study the performances of the composite methods for cases with different number of observations labels (d) for each hidden state. In particular, besides the emission matrix $\mathbb{E}_2$ defined in Section 3 which corresponds to the case when $d = 3$ we also consider the emission matrices:

$$\mathbb{E}_6 = \begin{bmatrix} 0.35 & 0.65 \\ 0.65 & 0.35 \\ 0.50 & 0.50 \end{bmatrix}, \quad \mathbb{E}_7 = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.3 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.4 & 0.2 \end{bmatrix},$$

which corresponds to the cases when $d = 2$ and $d = 4$, respectively. Combining these emission matrices with the transition matrices $\mathbb{P}_2$ ($S2$), $\mathbb{P}_4$ ($A1$) and $\mathbb{P}_8(t)$ ($N2$) defined in Section 3 we obtain nine cases (three symmetric, three asymmetric and three non-homogeneous cases).

The performance of the composite methods applied to these cases are presented in Table B.8 (see Table C.12 in Appendix C for performance of uncombined methods). Note that, the cases *S2.E2.d3*, *A1.E2.d3* and *N2.E2.d3* are the same as the cases *S2.M*, *A1.M* and *N2.M* defined in Section 3. The results show that the introduced composite methods also perform well for other choices than $d = 3$. Similar to previous conclusions for when $d = 3$, we see that same conclusions can be drawn for when $d = 2$ and 4 except for the symmetric case when $d = 2$ ($S2.E6.d2$) where it can be noted that the Voronoi tessellation strategy is clearly more efficient than the edge tessellation strategy (instead of similar results as when $d = 2$ or 4). Moreover, we can also note that Viterbi (optimum) performs in par with the best uncombined method in the symmetric cases and significantly better in the asymmetric cases.

B.4. Different type of separability between states

Here we investigate how the composite methods perform for emission matrices with even more extreme separability scenarios than those investigated in Section 3. In addition to $\mathbb{E}_1$ (L), $\mathbb{E}_2$ (M) and $\mathbb{E}_3$ (H) we also consider the emission matrices:

Table B.9
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Markov processes), is presented for the four composite methods, the worst and the best uncombined method from Table C.13 in Appendix C as well as the Viterbi algorithm applied on the resulting HMM models trained by the EM algorithm with both a random and optimum (only applied to the homogeneous cases) initialization strategy. The number in bold shows for each case the method, among the composite methods, that has the highest classification accuracy. The asterisk (*) indicates the methods where a significance difference (on 0.05 level) is detected when comparing the methods to the best uncombined method, using a two-tailed paired t-test.

Case	$C\{BV\}$	$C\{BE\}$	$C\{aBV\}$	$C\{aBE\}$	Worst uncombined method	Best uncombined method	*Viterbi* (random)	*Viterbi* (optimum)
S2.$\underline{\text{L}}$	**69.0**	58.7*	68.9	61.1*	52.4*	69.3	49.9*	64.2*
S2.L	87.9	**88.7***	88.0	88.5	76.4*	88.0	84.2*	87.1*
S2.M	93.5*	**93.8**	93.5*	93.6	92.9*	93.8	94.1	94.1
S2.H	96.2*	**96.3***	96.2*	96.2*	96.1*	96.5	96.8*	96.8*
S2.$\overline{\text{H}}$	99.3	**99.3**	98.8*	98.8*	98.7*	99.3	99.5*	99.5*
A1.$\underline{\text{L}}$	62.3*	72.8	63.1*	**73.0**	59.7*	73.1	61.7*	74.9*
A1.L	77.7*	88.1	80.9*	**88.3**	75.4*	88.7	80.2*	87.3
A1.M	86.1*	91.2*	89.3*	**92.5**	85.0*	93.4	92.5	94.7*
A1.H	91.3*	94.3*	93.4*	**96.0**	91.3*	96.4	96.8	97.3*
A1.$\overline{\text{H}}$	**99.3**	99.2*	99.1*	99.1*	99.0*	99.3	99.1	99.6*
N2.$\underline{\text{L}}$	**63.8**	56.3*	64.9*	58.4*	53.6*	63.8	51.5*	
N2.L	78.4*	79.2	80.1	**80.2**	69.6*	79.9	75.2*	
N2.M	86.3*	86.2*	87.8*	**88.0**	84.3*	88.5	87.9*	
N2.H	90.9*	91.0*	**92.5**	92.3	90.9*	92.8	92.7	
N2.$\overline{\text{H}}$	**98.5**	98.4*	98.2*	98.2*	98.1*	98.5	98.9*	

$$\mathbb{E}_0 = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0.4 \end{bmatrix}, \quad \mathbb{E}_4 = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix},$$

which represents even lower ($\underline{L}$) and higher ($\overline{H}$) degree of separability than the emission matrices $\mathbb{E}_1$ and $\mathbb{E}_3$, respectively. Combining the five emission matrices with the transition matrices $\mathbb{P}_2$ ($S2$), $\mathbb{P}_4$ ($A1$) and $\mathbb{P}_8(t)$ ($N2$) defined in Section 3 we obtain 15 cases.

The performance of the composite methods applied to these cases are presented in Table B.9 (see Table C.13 in Appendix C for performance of the uncombined methods). First note the natural results that accuracy increases with the degree of separability in the emission matrices. From Sections 3 and 4 we have previously seen, in the more modest cases of separability (L, M, H), that the (adaptive) edge tessellation strategy is preferable in the asymmetric cases, that the adaptive methods perform better in the non-homogeneous cases and that the differences between the methods are small for almost all the symmetric cases. The exception is the symmetric case $S1.L$ with short mean sojourn times and low separability of the state-specific emission probability distributions (L), where the Voroni tessellations methods work much better. When the mean sojourn times increase given the same emission matrices, i.e. $S2.L$ and $S3.L$, the edge strategy works in par with the Voronoi methods (cf. Table 5). For the extreme cases with low separability ($\underline{L}$), we here note that the Voronoi tessellation strategy is preferable in the symmetric and non-homogeneous cases ($S2.\underline{L}$ and ($N2.\underline{L}$) while the edge tessellation strategy is still preferable in the asymmetric cases (Table B.9). Note that $S2.\underline{L}$ has the same mean sojourn times as $S2.L$ but less distinct emission probability distributions. This illustrates that "separability" of states of the latent signal is not only defined by how distinctly different the state-specific emission probability distributions are. It also relates to the mean sojourn times of the states, as briefly touched upon in Appendix B.2. Furthermore, for all extreme cases with high separability ($\overline{H}$), we note that the non-adaptive methods have a slight tendency to perform better than the adaptive methods although we obtain a high accuracy with all methods (Table B.9). It indicates that when the separability of the states is high, they are easy to detect for all methods, and therefore no gain in using more complex methods such as the adaptive tessellations schemes. Comparisons of the results from the Viterbi (both random and optimum) reveal that it performs well for modest to high separability cases while it does not work as good for low separability cases.

Appendix C. Additional tables for the extended simulation study

Here additional tables of performance of the uncombined bagging clustering methods are given for the cases of the extended simulation study presented in Appendix B.

Table C.10
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Semi-Markov processes), is presented for different methods. The number in bold shows for each case the method that has the highest classification accuracy. The numbers in parentheses present the standard errors (of the classification accuracies) of the methods.

Case	*BVKM*	*BEKM*	*BVwKM*	*BEwKM*	*aBVKM*	*aBEKM*	*aBVwKM*	*aBEwKM*
S2.M	**93.8**(0.29)	93.3(0.29)	93.5(0.31)	93.7(0.29)	93.5(0.26)	92.9(0.27)	93.4(0.27)	93.4(0.26)
S2.M.logN	93.0(0.30)	91.0(0.64)	92.7(0.33)	92.9(0.33)	**93.0**(0.29)	91.4(0.50)	92.7(0.31)	92.8(0.30)
S2.M.Po	94.8(0.16)	94.4(0.20)	**94.9**(0.15)	94.9(0.17)	93.9(0.17)	92.6(0.22)	93.9(0.16)	93.2(0.18)
A1.M	89.7(0.74)	91.8(0.35)	85.0(0.99)	86.1(0.94)	91.7(0.59)	**93.4**(0.31)	87.4(0.89)	87.6(0.84)
A1.M.logN	86.6(1.22)	91.5(0.38)	82.0(1.31)	83.3(1.24)	88.8(1.09)	**93.5**(0.30)	84.4(1.19)	85.5(1.11)
A1.M.Po	90.1(0.70)	92.1(0.20)	84.6(0.86)	85.4(0.81)	91.6(0.58)	**93.2**(0.19)	86.6(0.77)	86.5(0.70)
N2.M	87.5(0.45)	84.3(0.86)	86.3(0.62)	86.1(0.63)	**88.5**(0.36)	85.6(0.73)	87.5(0.51)	87.4(0.55)
N2.M.logN	86.8(0.47)	83.7(0.89)	86.2(0.44)	86.1(0.50)	**87.9**(0.40)	84.5(0.83)	87.4(0.39)	87.2(0.44)
N2.M.Po	**85.4**(0.61)	79.9(1.19)	78.4(1.04)	78.9(0.97)	85.2(0.55)	79.8(1.05)	79.4(0.93)	80.1(0.87)

Table C.11
The average classification accuracy in percentage, based on 100 realisations (of different lengths T) generated from different cases (Markov processes), is presented for different methods. The number in bold shows for each case the method that has the highest classification accuracy. The numbers in parentheses presents the standard errors (of the classification accuracies) of the methods.

Case	*BVKM*	*BEKM*	*BVwKM*	*BEwKM*	*aBVKM*	*aBEKM*	*aBVwKM*	*aBEwKM*
S4.E5.T250	81.4(0.71)	77.1(1.27)	80.4(0.81)	81.3(0.78)	**81.6**(0.71)	77.6(1.19)	80.7(0.80)	81.5(0.75)
S4.E5.T500	**82.8**(0.39)	79.6(1.10)	82.2(0.50)	82.2(0.52)	82.5(0.38)	79.6(1.06)	82.1(0.45)	82.3(0.45)
S4.E5.T1000	**84.1**(0.26)	82.7(0.80)	83.8(0.36)	83.8(0.36)	83.6(0.27)	82.5(0.77)	83.4(0.35)	83.6(0.33)
S4.E5.T1500	83.6(0.24)	82.5(0.85)	83.6(0.25)	**83.8**(0.25)	83.0(0.24)	81.9(0.82)	83.1(0.24)	83.3(0.24)
S4.E5.T2000	84.1(0.22)	83.5(0.65)	**84.2**(0.21)	83.9(0.41)	83.7(0.22)	82.6(0.64)	83.7(0.22)	83.1(0.38)
S4.E5.T3000	84.0(0.18)	81.5(1.13)	84.0(0.18)	**84.1**(0.17)	83.3(0.17)	80.8(1.11)	83.2(0.18)	83.5(0.16)
A4.E5.T250	79.4(0.89)	80.3(1.11)	74.5(1.06)	76.6(1.00)	80.1(0.86)	**81.2**(1.01)	75.7(1.02)	77.8(0.97)
A4.E5.T500	80.9(0.70)	82.1(0.90)	77.9(0.87)	79.2(0.88)	82.6(0.56)	**83.0**(0.91)	79.6(0.78)	80.4(0.77)
A4.E5.T1000	82.0(0.54)	82.1(0.76)	81.1(0.71)	82.1(0.69)	**83.5**(0.45)	83.4(0.72)	82.3(0.64)	82.8(0.63)
A4.E5.T1500	82.4(0.44)	83.1(0.57)	82.1(0.56)	82.6(0.62)	83.5(0.39)	**83.7**(0.55)	83.0(0.48)	82.7(0.55)
A4.E5.T2000	83.0(0.40)	83.6(0.43)	82.8(0.53)	83.4(0.44)	**84.1**(0.36)	84.1(0.39)	83.6(0.48)	83.2(0.42)
A4.E5.T3000	83.2(0.28)	83.8(0.53)	83.2(0.29)	83.8(0.29)	84.3(0.25)	**84.3**(0.54)	84.0(0.26)	83.7(0.26)
N4.E5.T250	77.3(0.78)	73.9(1.13)	76.8(0.76)	77.1(0.81)	**78.6**(0.72)	75.1(1.12)	78.3(0.71)	78.1(0.77)
N4.E5.T500	80.9(0.48)	76.7(1.11)	79.8(0.61)	80.2(0.65)	**81.3**(0.43)	77.3(1.06)	80.4(0.56)	80.8(0.59)
N4.E5.T1000	82.1(0.29)	79.7(0.95)	80.9(0.49)	80.9(0.56)	**82.4**(0.26)	80.0(0.91)	81.4(0.41)	81.4(0.48)
N4.E5.T1500	82.0(0.27)	81.5(0.70)	80.6(0.53)	80.7(0.56)	**82.2**(0.24)	81.2(0.73)	80.9(0.49)	81.0(0.50)
N4.E5.T2000	82.3(0.20)	81.5(0.73)	82.2(0.21)	82.3(0.27)	**82.5**(0.20)	81.2(0.70)	82.4(0.19)	82.3(0.24)
N4.E5.T3000	82.3(0.23)	81.5(0.84)	81.9(0.43)	82.2(0.43)	**82.5**(0.21)	81.3(0.85)	82.1(0.40)	82.3(0.41)

Table C.12
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Markov processes), is presented for different methods. The number in bold shows for each case the method that has the highest classification accuracy. The numbers in parentheses present the standard errors (of the classification accuracies) of the methods.

Case	*BVKM*	*BEKM*	*BVwKM*	*BEwKM*	*aBVKM*	*aBEKM*	*aBVwKM*	*aBEwKM*
S2.E6.d2	74.7(1.09)	60.3(1.14)	79.8(0.74)	79.6(0.70)	76.2(1.14)	60.8(1.16)	**81.5**(0.72)	81.3(0.73)
S2.E2.d3	**93.8**(0.29)	93.3(0.29)	93.5(0.31)	93.7(0.29)	93.5(0.26)	92.9(0.27)	93.4(0.27)	93.4(0.26)
S2.E7.d4	**91.1**(0.36)	79.4(0.90)	90.2(0.60)	90.5(0.54)	90.7(0.32)	78.6(0.95)	90.4(0.47)	90.5(0.44)
A1.E6.d2	74.9(1.20)	84.6(1.20)	74.7(1.04)	74.6(1.07)	77.9(1.23)	**85.0**(1.21)	77.5(1.11)	77.3(1.11)
A1.E2.d3	89.7(0.74)	91.8(0.35)	85.0(0.99)	86.1(0.94)	91.7(0.59)	**93.4**(0.31)	87.4(0.89)	87.6(0.84)
A1.E7.d4	89.0(0.73)	88.4(0.56)	79.0(1.19)	81.0(1.09)	**90.1**(0.64)	88.2(0.64)	81.5(1.09)	82.6(1.02)
N2.E6.d2	69.4(0.82)	62.5(1.00)	71.8(0.69)	72.0(0.71)	71.2(0.81)	63.1(1.05)	73.5(0.70)	**73.9**(0.75)
N2.E2.d3	87.5(0.45)	84.3(0.86)	86.3(0.62)	86.1(0.63)	**88.5**(0.36)	85.6(0.73)	87.5(0.51)	87.4(0.55)
N2.E7.d4	82.5(0.64)	71.2(0.90)	82.3(0.58)	82.2(0.67)	83.4(0.66)	71.0(0.91)	**83.4**(0.53)	83.3(0.56)

Table C.13
The average classification accuracy in percentage, based on 100 realisations (of length $T = 3000$) generated from different cases (Markov processes), is presented for different methods. The number in bold shows for each case the method that has the highest classification accuracy. The numbers in parentheses present the standard errors (of the classification accuracies) of the methods.

Case	*BVKM*	*BEKM*	*BVwKM*	*BEwKM*	*aBVKM*	*aBEKM*	*aBVwKM*	*aBEwKM*
S2.$\underline{\text{L}}$	61.2(1.28)	52.4(1.14)	68.3(1.16)	69.2(1.13)	61.3(1.21)	52.5(1.12)	68.6(1.12)	**69.3**(1.11)
S2.L	87.3(0.67)	76.4(1.08)	87.5(0.63)	87.8(0.65)	87.2(0.58)	77.5(1.04)	87.8(0.52)	**88.0**(0.53)
S2.M	**93.8**(0.29)	93.3(0.29)	93.5(0.31)	93.7(0.29)	93.5(0.26)	92.9(0.27)	93.4(0.27)	93.4(0.26)
S2.H	96.2(0.16)	**96.5**(0.13)	96.2(0.19)	96.2(0.19)	96.3(0.14)	96.2(0.12)	96.1(0.15)	96.1(0.15)
S2.$\overline{\text{H}}$	99.3(0.03)	99.3(0.03)	99.3(0.03)	**99.3**(0.03)	98.9(0.04)	98.9(0.04)	98.8(0.04)	98.7(0.05)
A1.$\underline{\text{L}}$	64.8(1.87)	73.0(1.33)	59.7(1.15)	63.3(1.32)	65.2(1.83)	**73.1**(1.33)	60.8(1.20)	63.8(1.37)
A1.L	84.0(1.12)	88.1(0.50)	75.4(1.13)	76.6(1.07)	85.3(1.04)	**88.7**(0.47)	77.5(1.05)	78.1(1.01)
A1.M	89.7(0.74)	91.8(0.35)	85.0(0.99)	86.1(0.94)	91.7(0.59)	**93.4**(0.31)	87.4(0.89)	87.6(0.84)
A1.H	93.6(0.64)	95.1(0.22)	91.3(0.84)	91.6(0.82)	94.9(0.46)	**96.4**(0.15)	92.7(0.77)	92.9(0.73)
A1.$\overline{\text{H}}$	99.1(0.08)	**99.3**(0.03)	99.3(0.04)	99.2(0.04)	99.1(0.03)	99.0(0.06)	99.0(0.04)	99.0(0.05)
N2.$\underline{\text{L}}$	58.2(0.92)	53.6(0.96)	63.0(1.03)	63.3(0.94)	59.3(0.95)	54.2(1.00)	**63.8**(1.00)	63.8(0.95)
N2.L	78.6(0.71)	69.6(0.86)	78.5(0.76)	78.4(0.80)	79.7(0.62)	70.7(0.89)	**79.9**(0.68)	79.6(0.75)
N2.M	87.5(0.45)	84.3(0.86)	86.3(0.62)	86.1(0.63)	**88.5**(0.36)	85.6(0.73)	87.5(0.51)	87.4(0.55)
N2.H	91.8(0.24)	92.1(0.34)	90.9(0.47)	91.0(0.47)	92.6(0.19)	**92.8**(0.25)	91.9(0.36)	92.0(0.41)
N2.$\overline{\text{H}}$	98.5(0.04)	98.4(0.04)	**98.5**(0.04)	98.4(0.04)	98.3(0.04)	98.2(0.05)	98.2(0.04)	98.1(0.06)

References

Abramowicz, K., Arnqvist, P., Secchi, P., Sjöstedt de Luna, S., Vantini, S., Vitelli, V., 2017. Clustering misaligned dependent curves applied to varved lake sediment for climate reconstruction. Stoch. Environ. Res. Risk Assess. 31 (1), 71–85. https://doi.org/10.1007/s00477-016-1287-6.

Abramowicz, K., Schelin, L., Sjöstedt de Luna, S., Strandberg, J., 2019. Multiresolution clustering of dependent functional data with application to climate reconstruction. Stat 8 (1), e240. https://doi.org/10.1002/sta4.240.

Andreao, R., Dorizzi, B., Boudy, J., 2006. ECG signal analysis through hidden Markov models. IEEE Trans. Biomed. Eng. 53 (8), 1541–1549. https://doi.org/10.1109/TBME.2006.877103.

Bhati, S., Kamper, H., Murty, K.S.R., 2018. Phoneme based embedded segmental k-means for unsupervised term discovery. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 5169–5173.

Egozcue, J.J., Pawlowsky-Glahn, V., Mateu-Figueras, G., Barcelo-Vidal, C., 2003. Isometric logratio transformations for compositional data analysis. Math. Geol. 35 (3), 279–300. https://doi.org/10.1023/A:1023818214614.

Gan, G., Ma, C., Wu, J., 2007. Data Clustering: Theory, Algorithms, and Applications. SIAM.

Kamper, H., Jansen, A., Goldwater, S., 2017a. A segmental framework for fully-unsupervised large-vocabulary speech recognition. Comput. Speech Lang. 46, 154–174. https://doi.org/10.1016/j.csl.2017.04.008.

Kamper, H., Livescu, K., Goldwater, S., 2017b. An embedded segmental k-means model for unsupervised segmentation and clustering of speech. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, pp. 719–726.

Lemmens, A., Croux, C., Stremersch, S., 2012. Dynamics in the international market segmentation of new product growth. Int. J. Res. Mark. 29 (1), 81–92. Special section on Global Brand Management.

Lin, J., 1991. Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory 37 (1), 145–151. https://doi.org/10.1109/18.61115.

Menafoglio, A., Secchi, P., 2017. Statistical analysis of complex and spatially dependent data: a review of object oriented spatial statistics. Eur. J. Oper. Res. 258 (2), 401–410. https://doi.org/10.1016/j.ejor.2016.09.061.

R Core Team, 2019. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Ross, S., 2012. Simulation. Knovel Library. Elsevier Science.

Secchi, P., Vantini, S., Vitelli, V., 2013. Bagging Voronoi classifiers for clustering spatial functional data. Int. J. Appl. Earth Obs. Geoinf. 22, 53–64. https://doi.org/10.1016/j.jag.2012.03.006.

Stephens, M., 2000. Dealing with label switching in mixture models. J. R. Stat. Soc., Ser. B, Stat. Methodol. 62 (4), 795–809. https://doi.org/10.1111/1467-9868.00265.

Tseng, G.C., 2007. Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data. Bioinformatics 23 (17), 2247–2255. https://doi.org/10.1093/bioinformatics/btm320.

Turner, R., 2020. hmm.discnp: hidden Markov models with discrete non-parametric observation distributions. https://CRAN.R-project.org/package=hmm.discnp. R package version 3.0-6.

Zucchini, W., MacDonald, I., Langrock, R., 2017. Hidden Markov Models for Time Series: An Introduction Using R, second edition. Chapman & Hall/CRC Monographs on Statistics and Applied Probability. CRC Press.