Sequence Classification Using Third-Order Moments

Troelsgaard, Rasmus; Hansen, Lars Kai

Published in:
Neural Computation

Link to article, DOI:
10.1162/neco_a_01033

Publication date:
2017

Document Version
Peer reviewed version

LETTER Communicated by Rong Ge

Sequence Classification Using Third-Order Moments

Rasmus Troelsgaard
rast@dtu.dk
Lars Kai Hansen
lkai@dtu.dk
Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby 2860, Denmark

Model-based classification of sequence data using a set of hidden Markov models is a well-known technique. The involved score function, which is often based on the class-conditional likelihood, can, however, be computationally demanding, especially for long data sequences. Inspired by recent theoretical advances in spectral learning of hidden Markov models, we propose a score function based on third-order moments. In particular, we propose to use the Kullback-Leibler divergence between theoretical and empirical third-order moments for classification of sequence data with discrete observations. The proposed method provides lower computational complexity at classification time than the usual likelihood-based methods. In order to demonstrate the properties of the proposed method, we perform classification of both simulated data and empirical data from a human activity recognition study.

1 Introduction

Classification and clustering of sequences into categories is essential to human interpretation of the data. Different methodologies have been proposed to deal with this problem, and Xing, Pei, and Keogh (2010) give a brief and general overview of the field, including model-based classification. The general approach in model-based classification is to represent each class by a generative model; hence, there are two main components in this classification system. The first is the formulation of the statistical model representing each of a given set of classes, and the second is a measure of distance between observed data and models. For probabilistic models, the obvious and common choice is to use a distance measure derived from the class-conditional likelihoods.

Each model is estimated using a set of exemplar data sequences (training set) representing a specific class. Hence, the problem can be stated as follows: given L trained models and a held-out, observed sequence of length N, find the model that best fits the observation. One classical approach to this problem is to use the (log-)likelihood of each class-conditional model

doi:10.1162/NECO_a_01033

given the test sequence as a score for the model-sequence pair. Usually the test sequence is assigned to the class model for which the (log-)likelihood is the highest.

In this letter, we consider class-conditional model-based classification for sequential data using hidden Markov models. Classification using hidden Markov models in particular has been applied in a variety of contexts. Oates, Firoiu, and Cohen (1999) take the classical model-based approach to the clustering of sequence data using one HMM per cluster. An original HMM-based representation of images is explored in Mouret, Solnon, and Wolf (2009). Wong & Stamp (2006) used HMMs to represent software virus families and a log-likelihood threshold for binary classification of benign software versus malware. Another practical example is found in Wang, Mehrabi, and Kannatey-Asibu (2002), where HMM-based classification is applied for monitoring the wear on tools in industrial machinery. Bicego, Murino, and Figueiredo (2004) used the similarities between sequences and models as features in a discriminatively trained classifier. One HMM is estimated for each training example, all sequences are then embedded in the space of estimated HMMs using log likelihood. This line of thought is also explored in García-García, Emilio, and Díaz-de-Mará (2009), who proposed a KL-divergence-based similarity measure.

Recently, methods based on spectral decomposition of observed data moments have been developed for parameter estimation in models for sequential data (Hsu, Kakade, & Zhang, 2012; Anandkumar, Hsu, & Kakade, 2012). While these methods provide exciting results regarding both global convergence and the computational complexity of the parameter estimation problem, the complexity of likelihood calculations, which is of particular interest when performing model-based sequence classification, is unchanged. In settings where the amount of data to be classified is vast and time spent on model estimation is of minor importance, we find ourselves in need of a fast approximation to the likelihood that does not require calculating matrix products for every observation in a given sequence. The advances in spectral learning using moments enable us to view the third-order moments as sufficient statistics under the model assumptions of Hsu et al. (2012) and Anandkumar et al. (2012). Based on this interpretation, we propose a simple framework for classification of sequences of discrete observations, using only observed third-order moments. The distance measure we propose to substitute for likelihood calculations is based on Kullback-Leibler divergence between empirical and theoretical third-order moments, and we show that it has lower computational complexity at classification time, while achieving indistinguishable performance. An implementation of the proposed method is available at https://github.com/tro4els/HMM-moment-classification.

This rest of this letter is organized as follows. Section 3 introduces the proposed score function in the context of both stationary and nonstationary HMMs and relates it to a particular composite likelihood. Next, we compare

the computational complexity of the proposed method to the likelihood-based approach. Finally, an upper bound on the convergence time of a Markov chain is exploited to reduce memory requirements for the proposed method. Section 4 sketches an approach to sequence embedding wherein the distance score for sequence-model pairs plays a central role. In sections 5 and 6, we present classification results of both simulated and real-world data sets respectively.

2 Definitions

In this letter, we use the standard parameterization of the discrete hidden Markov model with S hidden states and K observation symbols:

$$\begin{aligned}
\boldsymbol{\pi}^{(1)} \in \mathbb{R}^S&: \text{ Initial state probability vector}\\
&\quad \boldsymbol{\pi}^{(1)}_h = P(z_1 = h)\\
\boldsymbol{T} \in \mathbb{R}^{S\times S}&: \text{ Transition probability matrix}\\
&\quad \boldsymbol{T}_{g,h} = P(z_n = g|z_{n-1} = h) \qquad n \geq 2\\
\boldsymbol{O} \in \mathbb{R}^{K\times S}&: \text{ Observation probability matrix}\\
&\quad \boldsymbol{O}_{i,h} = P(x_n = i|z_n = h) \qquad n \geq 1
\end{aligned}$$

for $g, h \in \{1, 2, \ldots, S\}$, $i \in \{1, 2, \ldots, K\}$.

3 KL Divergence of Third-Order Moments

In this section, we develop the basic ideas of using third-order moments for classification of sequence data. Recently, the work of Hsu et al. (2012) and Anandkumar et al. (2012) proved that parameter estimation in the hidden Markov model is possible with observed moments of orders as low as 3 under certain rank conditions of the parameter matrices. This means that third-order moments act as sufficient statistics for the HMM under the mild conditions rank($\boldsymbol{T}$)= rank($\boldsymbol{O}$) $= S$. We now use this interpretation of the third-order moments as sufficient statistics of the HMM to formulate a score function relating an observed sequence to an estimated HMM.

The main idea is to use the third-order moments of observed discrete sequences as multinomial probability distributions. These distributions can then be related to the theoretical third-order moments due to a set of model parameters, via a suitable probabilistic measure such as the KL divergence. Because the third-order moments in the general case are dependent on the initial state distribution $\boldsymbol{\pi}^{(1)}$, we start by describing the simplified case of assumed stationarity of the HMM ($\boldsymbol{\pi}^{(1)} = \hat{\boldsymbol{\pi}}$).

3.1 Stationary Markov Processes. Let $\bar{P}_{1,2,3}$ be the empirical third-order moment of the observed sequence, and let $P_{1,2,3}$ be the corresponding theoretical third-order moment due to model parameters:

$$P_{1,2,3}(\cdot, k, \cdot) = \boldsymbol{O}\,\mathrm{diag}(\hat{\boldsymbol{\pi}})\boldsymbol{T}^\top \mathrm{diag}(\boldsymbol{O}(k, \cdot))\boldsymbol{T}^\top \boldsymbol{O}^\top \qquad k \in [1, K].$$

We can then use the KL divergence of $P_{1,2,3}$ from $\bar{P}_{1,2,3}$ as a measure of difference between a model and an observed sequence:

$$\mathrm{KL}(\bar{P}_{1,2,3} \| P_{1,2,3}) = \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K} \bar{P}_{1,2,3}(i,\, j, k) \log \frac{\bar{P}_{1,2,3}(i,\, j, k)}{P_{1,2,3}(i,\, j, k)}.$$

Note that this is also valid in the nonstationary case if observations from a suitable burn-in period are discarded (see section 3.5). This, however, requires that the length of the test sequence is at least as long as the maximum convergence time of all the class models, which might limit the practical usefulness of the method.

With the goal of avoiding discarding burn-in data for classification in the nonstationary case, we now present the main contribution of this letter.

3.2 Nonstationary Markov Processes. If stationarity cannot be assumed, the expectation of the state distribution changes along the underlying Markov chain. Hence, we have to consider the third-order moments for each triplet in the observed sequence separately. Let $\bar{P}_{n,n+1,n+2}$ be the empirical third-order moment of the triplet starting at position n in the sequence, and let $P_{n,n+1,n+2}$ be the corresponding theoretical third-order moment due to model parameters:

$$P_{n,n+1,n+2}(\cdot, k, \cdot) = \boldsymbol{O}\,\mathrm{diag}(\boldsymbol{T}^{n-1}\boldsymbol{\pi}^{(1)})\boldsymbol{T}^\top \mathrm{diag}(\boldsymbol{O}(k, \cdot))\boldsymbol{T}^\top \boldsymbol{O}^\top \quad k \in [1, K].$$

We can then, for an arbitrary position n, calculate the KL divergence of $P_{n,n+1,n+2}$ from $\bar{P}_{n,n+1,n+2}$:

$$\begin{aligned}\mathrm{KL}^{(n)} &= \mathrm{KL}(\bar{P}_{n,n+1,n+2} \| P_{n,n+1,n+2}) \\ &= \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K} \bar{P}_{n,n+1,n+2}(i,\, j, k) \log \frac{\bar{P}_{n,n+1,n+2}(i,\, j, k)}{P_{n,n+1,n+2}(i,\, j, k)}. \end{aligned} \tag{3.1}$$

Each $\mathrm{KL}^{(n)}$ can then interpreted as a cost describing how well $\bar{P}_{n,n+1,n+2}$ approximates the theoretical third-order moment of that particular triplet $P_{n,n+1,n+2}$.

Note that in the typical classification scenario, the cost is calculated for a single sequence $\boldsymbol{x} = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$. Thus, for any given $n \in \{1, 2, \ldots, N-2\}$, equation 3.1 reduces to $-\log(P_{n,n+1,n+2}(x^{(n)}, x^{(n+1)}, x^{(n+2)}))$. To obtain a cost using the full sequence, we calculate the arithmetic mean across all triplets, which is exactly equivalent to considering the joint

discrete probability distribution of all triplets in the sequence:

$$\frac{1}{N-2}\sum_{n=1}^{N-2}\mathrm{KL}^{(n)} = \frac{1}{N-2}\sum_{n=1}^{N-2} -\log\left(P_{n,n+1,n+2}(x^{(n)}, x^{(n+1)}, x^{(n+2)})\right). \tag{3.2}$$

With N being the length of the observed candidate sequence, the described procedure requires the calculation of up to the Nth power $\boldsymbol{T}$, which can be demanding in terms of memory. However, the Markov chain converges to its stationary distribution, and for a given allowed distance ε from this stationary distribution, it is possible to derive a bound on the convergence time for the chain. This can be exploited to limit the maximum power of $\boldsymbol{T}$ to calculate. In section 3.5, we derive such a convergence time bound. Let $c_{i,j,k} \geq 0$ be the number of occurrences of the triplet $(i,\ j,\ k)$ in the stationary part of the sequence $\boldsymbol{x}$, and let $c_s = \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K} c_{i,j,k}$ be the number of triplets beyond the convergence time. We can then simply calculate the KL divergence from the stationary distribution and use the weighted arithmetic mean:

$$\begin{aligned}
&\frac{1}{N-2}\sum_{n=1}^{N-2-c_s}\mathrm{KL}^{(n)} + \frac{c_s}{N-2}\mathrm{KL}^{\text{stationary}} \\
&= \frac{1}{N-2}\sum_{n=1}^{N-2-c_s} -\log\left(P_{n,n+1,n+2}(x^{(n)}, x^{(n+1)}, x^{(n+2)})\right) \\
&\quad + \frac{c_s}{N-2}\sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K}\frac{c_{i,j,k}}{c_s}\log\frac{\frac{c_{i,j,k}}{c_s}}{\hat{P}_{1,2,3}(i,\ j,\ k)} \\
&= \frac{1}{N-2}\sum_{n=1}^{N-2} -\log\left(P_{n,n+1,n+2}(x^{(n)}, x^{(n+1)}, x^{(n+2)})\right) \\
&\quad + \frac{1}{N-2}\sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K} c_{i,j,k}\log\frac{c_{i,j,k}}{c_s},
\end{aligned} \tag{3.3}$$

where $\hat{P}_{1,2,3}$ denotes the stationary third-order moment. We observe that equation 3.3 is just equation 3.2 plus the additional term on the last line of equation 3.3, which is due to the Shannon entropy of the empirical stationary third-order moment.

3.3 Interpretation as Composite Likelihood. An empirical moment estimated from a single triplet is clearly a very crude approximation. Contrast the usual practice of method of moments, where averaging over

a huge number of samples is exploited. The intuition behind using the one-sample approximations along the chain is that each of the terms $\mathrm{KL}^{(n)} = -\log\left(P_{n,n+1,n+2}(i, j, k)\right)$ on average is lower for a matching pair of sequence and model than for nonmatching pairs. Furthermore, by viewing the model-based third-order moments along a Markov chain as a reparameterization of the HMM, when disregarding the entropy term, equation 2.3 corresponds to a negative per sample composite log likelihood of this model given the observations (triplets). The pseudo-likelihood was introduced in Besag (1975) as a product of possibly correlated local conditional likelihood terms. Later, under the term *composite likelihood* Lindsay (1988) generalized the concept to also include marginal likelihood terms of subcomponents. This interpretation of the KL-divergence-based distance score further justifies the proposed approach. Based on the above analysis, we propose the following composite log-likelihood score function for model-based classification using HMMs:

$$\mathcal{D}(\boldsymbol{x}, \boldsymbol{P}) = \frac{1}{N-2} \sum_{n=1}^{N-2} -\log\left(P_{n,n+1,n+2}(x^{(n)}, x^{(n+1)}, x^{(n+2)})\right). \tag{3.4}$$

3.4 Computational Complexity. We now compare the computational complexity of the proposed method and the classical likelihood-based approach. The cost of estimating the L class HMMs is disregarded as we focus solely on the classification step.

We start by examining the total complexity of scoring a single observed sequence by L estimated models. The likelihood calculations scale with $\mathcal{O}(LNS^2)$. Thus, we obtain a mean per class complexity of $\mathcal{O}(NS^2)$.

In the stationary case, the third-order moment of the test sequence can be calculated in $\mathcal{O}(N)$, and because it is independent of the number of classes, it has to be calculated only once. Comparison of the third-order moment of a test sequence to moments of all the trained class models takes $\mathcal{O}(\min(N, K^3)L)$. Here, it is exploited that the cost function depends on only the N triplets that are actually observed. This means that in the stationary case, the total computational complexity of the moment comparison becomes $\mathcal{O}(\min(N, K^3)L + N)$ and $\mathcal{O}(\min(N, K^3) + \frac{N}{L})$ for the per class complexity.

In the nonstationary case we have to consider all triplets in the test sequence separately, resulting in a total complexity of $\mathcal{O}(NL)$ and per class complexity $\mathcal{O}(N)$. This analysis shows that the classification task in theory can be performed faster when using third-order moments compared to the classical likelihood approach.

Although the computational complexity remains unchanged, the memory requirements will of course increase compared to the stationary situation as we have to store third-order moments for all possible positions in a chain (in theory, infinitely many). Section 3.5 outlines a method to limit the

amount of required memory based on an upper bound on the convergence time of a Markov chain (to the stationary distribution).

3.5 Estimating Convergence Time for a Markov Chain. This section describes how to calculate an upper bound on the convergence time of an ergodic Markov chain given an upper bound on the total variation distance at any given time instance t. We begin by stating a bound for the slightly simpler case of a reversible Markov chain and then proceed to the more general case of a nonreversible chain. The convergence time of a reversible irreducible Markov chain with transition probability matrix $\boldsymbol{T}$ and stationary distribution $\hat{\boldsymbol{\pi}}$ can be bounded using an upper bound on the relative pointwise distance $\Delta(t)$. This quantity is larger than the total variation distance $\Delta(t) = \max_{i,j} |\frac{\boldsymbol{T}^t_{i,j}}{\hat{\boldsymbol{\pi}}_i} - 1|$ for which the following bound exists: $\Delta(t) \leq \frac{\beta_1(\boldsymbol{T})^t}{\hat{\boldsymbol{\pi}}_{\min}}$ where $\beta_1(\cdot)$ denotes the second largest eigenvalue (Durrett, 2007).

For a general nonreversible Markov chain, a similar result exists for the multiplicative reversibilization of $\boldsymbol{T}$, $M(\boldsymbol{T}) = \boldsymbol{T}\tilde{\boldsymbol{T}}$, where $\tilde{\boldsymbol{T}}_{j,i} = \frac{\hat{\boldsymbol{\pi}}_j \boldsymbol{T}_{i,j}}{\hat{\boldsymbol{\pi}}_i}$ (Fill, 1991). Let $\mathcal{X}_0^2 = \sum_{x=1}^{S} \frac{(\boldsymbol{\pi}^{(1)}_x - \hat{\boldsymbol{\pi}}_x)^2}{\hat{\boldsymbol{\pi}}_x}$. Then, according to Fill (1991), the upper bound on the total variation distance at time step t is

$$\left\| \boldsymbol{T}^t \boldsymbol{\pi}^{(1)} - \hat{\boldsymbol{\pi}} \right\|_{\text{TV}} = \frac{1}{2} \sum_{x=1}^{S} \left| (\boldsymbol{T}^t \boldsymbol{\pi}^{(1)})_x - \hat{\boldsymbol{\pi}}_x \right| \leq \frac{(\beta_1(M(\boldsymbol{T})))^{\frac{t}{2}}}{2} \mathcal{X}_0,$$

from which we can construct an upper bound on t given an acceptable total variation distance $\varepsilon \in [0, \min\left(1, \frac{\mathcal{X}_0}{2}\right)$:]

$$\begin{aligned} & \varepsilon \geq \frac{(\beta_1(M(\boldsymbol{T})))^{\frac{t}{2}}}{2} \mathcal{X}_0 \\ \iff & t \geq 2 \frac{\log\left(\frac{2\varepsilon}{\mathcal{X}_0}\right)}{\log \beta_1 \left(M(\boldsymbol{T})\right)}. \end{aligned} \tag{3.5}$$

This bound can be used to limit the number of third-order moments to store in memory and thereby make classification more feasible.

3.6 Classification Procedure. We have now introduced all the necessary tools for a procedure to classify sequences using third-order moment representation of class-conditional HMMs. The procedure is described in algorithm 1. For simplicity, the algorithm assumes equal prior class probabilities, but an extension using a nonuniform prior is straightforwardly obtained by subtracting the logarithm of the prior class probabilities from the corresponding distance scores.

Algorithm 1: Classification Procedure Based on Third-Order Moments.

Input: Test sequences: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_{\text{test}}}\}$. Class-conditional hidden Markov models: $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_L\}$. Maximum total variation distance from stationarity: ε.

Output: Test data class labels: $(y_1, y_2, \cdots, y_{N_{\text{test}}})$.

for $c = 1$ **to** L **do**

Calculate number of third order moments, t_c, to make and store using the ε and equation 3.5.

Use model parameters $\mathcal{M}_c$ to calculate theoretical third-order moments at all time steps up to t_c and the stationary third-order moment. Store as $\boldsymbol{P}^{(c)}$.

end for

for $i = 1$ **to** N_{test} **do**

for $c = 1$ **to** L **do**

$d_c = D(\boldsymbol{x}_i, \boldsymbol{P}^{(c)})$

end for

$y_i = \underset{c \in [1,L]}{\arg\min}\ d_c$

end for

3.7 Exploiting Approximate Convergence. We now show an example of how classification performance can be affected by the size of ε. We illustrate the effect by analyzing a simulated five-class problem using the proposed composite likelihood as a distance score in the classifier (as described in section 3.2). For the purpose of illustration, all class models share parameters $\boldsymbol{T}$ and $\boldsymbol{S}$ but differ by their initial distributions $\boldsymbol{\pi}^{(1)}$. Thus, all class-conditional models have identical stationary distributions and identical stationary third-order moments. We assess the classification performance using the well-known F_1-measure. Figures 1 and 2 show how the classification performance decreases when the accepted distance to the stationary distribution is increased.

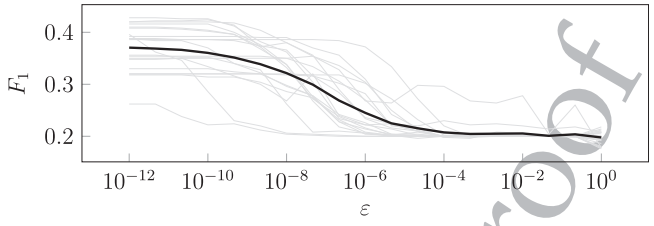


Figure 1: Classification performance using KL divergence as a function of ε. Repetitions of the experiment are in gray, and the mean classification score of the repetitions is in black.

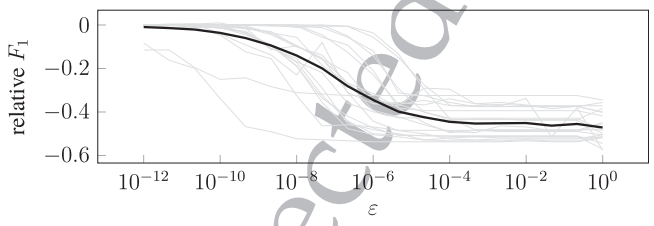


Figure 2: Classification performance using KL divergence as a function of ε. This plot shows the performance relative to using $\varepsilon = 10^{-20}$ (not assuming convergence). Repetitions of the experiment are in gray, and the mean classification score of the repetitions is in black.

4 "Embedding" Sequences for Classification

To improve on the classical model-based classification approach, several authors have suggested "embedding" the observed test sequences in a space spanned by the training sequences (García-García et al., 2009; Bicego, Murino, & Figueiredo, 2004). An arbitrary discriminatively trained classifier can then be applied to leverage this new representation of sequences.

The main idea is to estimate a single HMM for each training example and let the embedding of a sequence be defined by the distance scores relating it to all the training models.

Similar to the work in García-García et al. (2009), for a single sequence, we normalize its scores relating it to the training sequences, such that it sums to 1. This allows us to use the Jensen-Shannon divergence as the similarity score in the embedding space. Given a test sequence to be classified, one has to evaluate the distance score for all trained models. Hence, the distance score remains a central component of the classification procedure. The procedure is described in algorithm 2 in appendix B.

We include this classification strategy to provide an alternative evaluation of the proposed composite likelihood distance score. For the results

presented in sections 5 and 6, we used a K-nearest-neighbor classifier where K was chosen via five-fold cross-validation on the training sequences.

5 Classification of Simulated Time Series

This section illustrates how the proposed composite likelihood score, $\mathcal{D}$, compares to the negative log likelihood, ℓ, under different simulated conditions such as lengths of the observed sequences, diagonality of the transition matrices, and how interrelated the class-conditional models are.

For estimation of the class-conditional models, we rely on the classical Baum-Welch/EM algorithm (Baum, Petrie, Soules, & Weiss, 1970; Dempster, Laird, & Rubin, 1977). Although alternatives such as spectral estimation techniques presented by Anandkumar et al. (2012), Anandkumar, Ge, and Hsu (2014), and Troelsgaard and Hansen (2016) in principle could be used as well, in order not to unintentionally favor the moment-based classification scheme, the likelihood-based estimation is preferred.

The numbers of symbols in the training and test sequences are $\sim$ Poisson($\bar{N}$), $\bar{N} \in \{10, 50, 200, 1000\}$. The numbers of training and test sequences per class are fixed at 30 and 50, respectively.

The diagonality is controlled by the parameter $T_{\text{diag}} \in]0, 1[$. The parameter $\rho \in]0; 1]$ controls the variance of the elements of $\boldsymbol{T}$ and is used as a means to generate sets of more or less interrelated HMMs. For a detailed description of the construction of the HMMs used in these classification experiments, we refer readers to appendix A.

We consider a simulated classification problem with $L = 5$ classes, where each class-conditional model (unless explicitly stated otherwise) is an $S = 4$ state HMM with $K = 15$ discrete observation symbols.

5.1 Results. The performance is reported in terms of the F_1-measure. The reported evaluation quantities are mean values over all classes. Each experiment was repeated 20 times to quantify variation in performance. The error bars denote the standard deviations of the estimated mean values.

Figure 3 shows how classification performance is improved by longer observed sequences. Furthermore, class-conditional models closer to each other are harder to distinguish between. These observations hold for both ℓ and $\mathcal{D}$. The performances of the two methods are virtually indistinguishable, with the exception that for long sequences ($\bar{N} \gtrsim 1000$) and class-conditional models quite close to each other ($\rho \lesssim 0.05$), $\mathcal{D}$ seems to be superior. To better illustrate the minor differences, Figure 4 shows the mean of the pairwise relative performances relative to ℓ. Hence the results for ℓ are constant at 1.

In total, we performed 428 experiments with different combinations of parameters. With the null hypothesis that $F_1(\ell) \geq F_1(\mathcal{D})$, we can calculate p-values for the experiment by applying Bonferroni correction to paired-samples binomial sign tests. Hence, we calculate the probability of observing the experimental results or more extreme results under the null

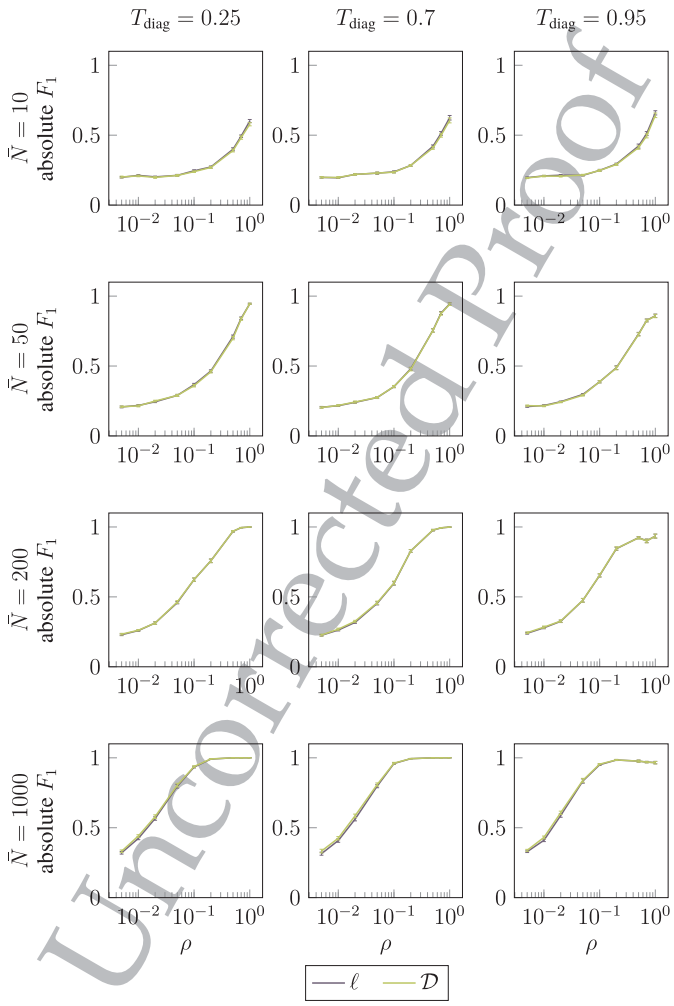


Figure 3: This figure shows how the performances of ℓ and $\mathcal{D}$ vary for different amounts of diagonality of $\boldsymbol{T}$ and the parameter ρ. The results are reported in terms of F_1 using $\varepsilon = 0.001$ in the calculation of the bound on the convergence time.

hypothesis. For a couple of the classification problems with low values of ρ shown in the lower plots of Figure 4 ($\bar{N} = 1000$, $T_{\text{diag}} \in \{0.25, 0.7, 0.95\}$), we find (corrected) p-values in the range $[0.0008, 0.0327]$ indicating that the null hypothesis is very unlikely for these particular classification problems.

For the null hypothesis $F_1(\ell) \leq F_1(\mathcal{D})$, the three lowest obtained p-values were 0.0620, 0.1722, and 0.3118, indicating no general tendency to reject the null hypothesis. Because the true likelihood is the best possible score

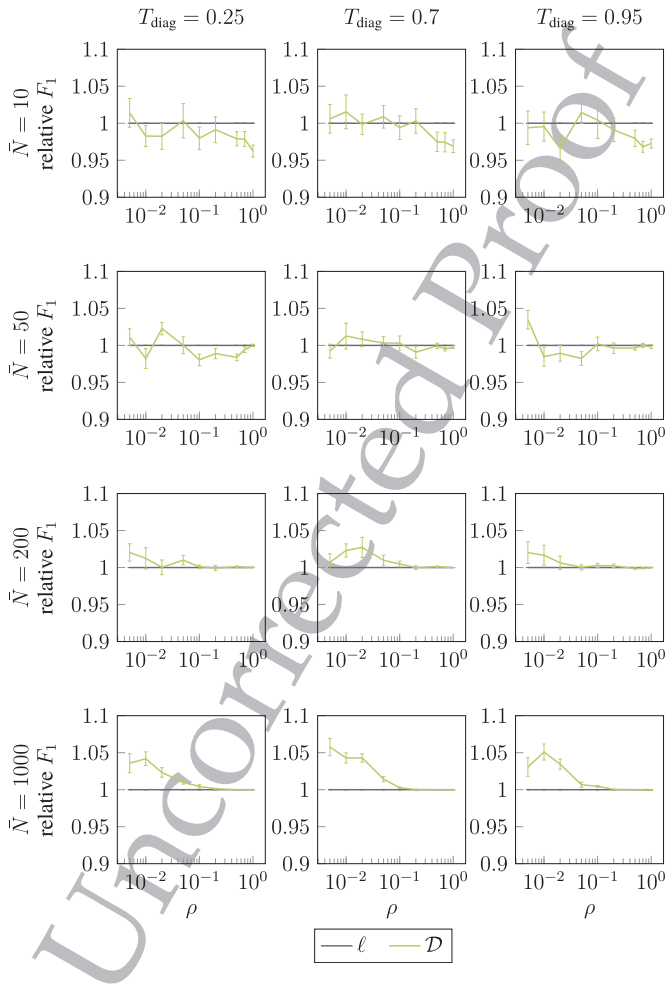


Figure 4: This figure shows how the relative performances of ℓ and $\mathcal{D}$ vary for different amounts of diagonality and the parameter ρ. See Figure 3 for absolute performance. The results are reported in terms of F_1 relative to the score of ℓ. In the calculation of the bound on the convergence time, we set $\varepsilon = 0.001$.

function if the assumed model is correct, the obtained results should raise suspicion if the class-conditional models were exact. This is, however, not the case in these experiments, where both training and test data are simulated from a set of HMMs. We ascribe the obtained results to the fact that the class-conditional models are estimated from a finite set of example sequences, but detailed analyses of this phenomenon are beyond the scope of this letter.

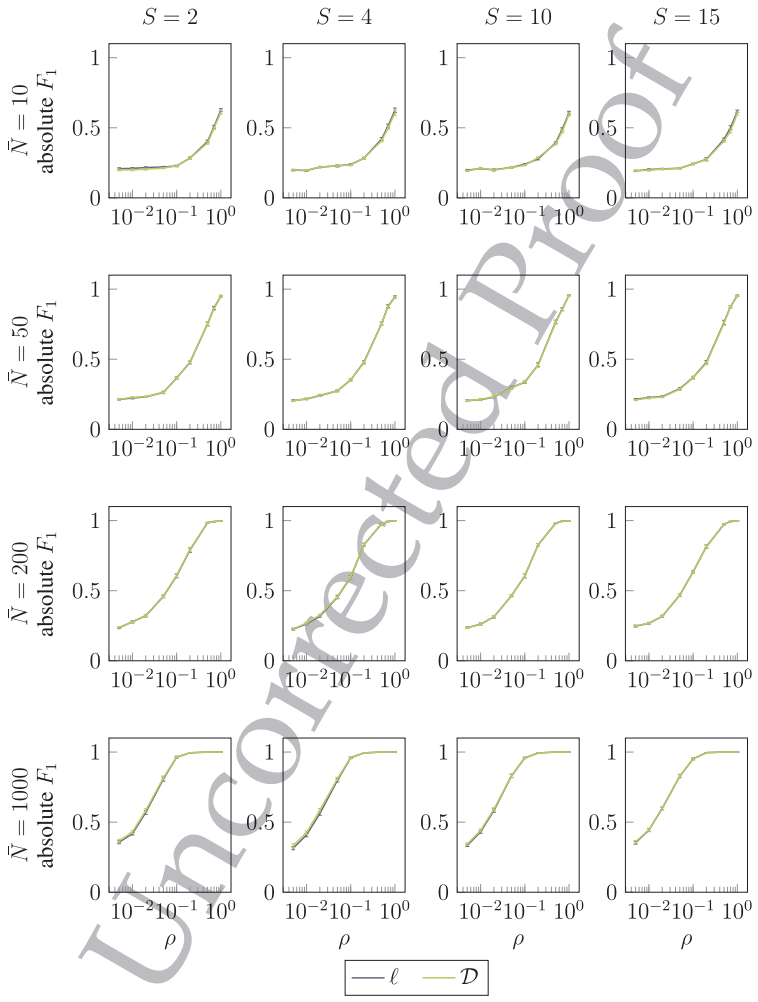


Figure 5: This figure shows that the performances of ℓ and $\mathcal{D}$ remain comparable for different assumed sizes, S, of the state space. This property is important because the relative advantage of the composite likelihood with regard to computational complexity increases with S. In the calculation of the bound on the convergence time, we set $\varepsilon = 0.001$. The results are conditioned on $T_{\text{diag}} = 0.7$.

To illustrate how the distance score $\mathcal{D}$ performs in regimes where it becomes increasingly cost effective (i.e., larger values of S; see section 3.4), Figure 5 shows the absolute performance of ℓ and $\mathcal{D}$ for $S \in \{2, 4, 10, 15\}$. We observe no clear performance difference, which further strengthens the eligibility of using the proposed composite likelihood as a score function

in this particular HMM classification setting. Figure 7 shows time spent on calculating score function $\mathcal{D}$ relative to ℓ for different values of S and $\bar{N}$. As expected, the performance advantage of using the composite likelihood increases with the assumed state-space size S. The computational advantage, however, does not exactly match the theoretical improvement, which we ascribe to implementation and internal optimization of Matlab. In summary, the statistical tests indicate that using $\mathcal{D}$ as the distance score in HMM-based classification of discrete sequence data provides equally good results compared to the classical likelihood score ℓ at a reduced computational cost.

5.2 Classification Results for Sequence Embedding. Using the sequence embedding procedure described in section 4, we now compare performance to the classical model-based approach. Figure 6 shows that ℓ and $\mathcal{D}$ perform equally well in all the simulated classification problems. Furthermore, we observe that the embedding improves performance slightly for moderate to long sequences ($\bar{N} \in 50, 200, 1000$) when class-conditional models are quite different and have a dominating diagonal structure ($T_{\text{diag}} = 0.95$). On the contrary, the embedding seems to have a negative impact on performance under the conditions of more interrelated class-conditional models and fewer diagonal transition matrices. These performances of ℓ and $\mathcal{D}$ for high values of ρ are significantly better than without the embedding (cf. the significance test in the previous section).

Figure 7 shows the time spent on classification relative to the time of ℓ. The figure clearly illustrates the gains of the reduced computational complexity of using $\mathcal{D}$ over ℓ for everything but very short sequences.

5.3 Conclusion of Experiment with Simulated Data. For short sequences, using the classical log-likelihood approach is both faster and more accurate in terms of F_1 score. For increased sequence lengths, in addition to being faster, the performance of the composite likelihood score catches up and produces results indistinguishable from the log likelihood. Embedding test sequences in the space of training sequences seems to be most beneficial for long sequences ($\gtrsim 50$) as long as class-conditional models are quite dissimilar.

6 Classification of Human Activities

We now turn to application of the proposed method on nonsimulated sequence data. We use the UCI HAR benchmark data set (Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2013), a human activity recognition data set consisting of inertial measurements from a waist-mounted mobile device during six different activities. We perform five-fold cross-validation on the training set (21 persons) for finding the optimal number of states S, for each class. Table 1 shows the class labels and the optimal number of hidden states for each of the six classes. As input, we used body acceleration and

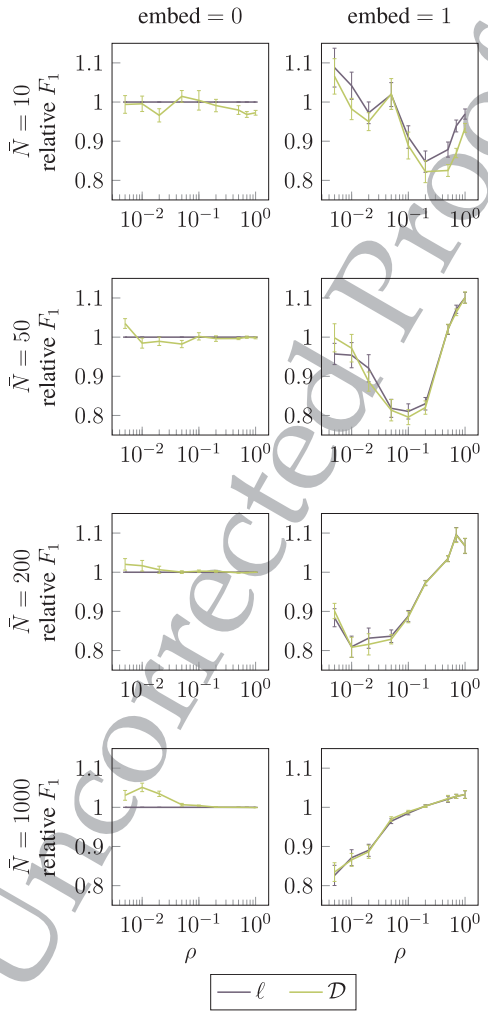


Figure 6: Relative performance when using sequence embedding for classification. The presented results are conditioned on $\varepsilon = 0.001$, $S = 4$, and $T_{\text{diag}} = 0.95$.

angular velocity, and all variables were scaled to unit variance, whitened, and quantized into 50 "symbols" using K-means clustering (Elkan, 2003).

In this experiment, we assume that the boundaries of activities are known in advance such that every training and test sequence contains data from only a single activity. Thus, the task is to provide a label for each test segment.

Using the values in Table 1, we estimated one HMM per class using the full training set and then classified the sequences corresponding to the nine left-out persons. The confusion matrix shown in Figure 8 is obtained from

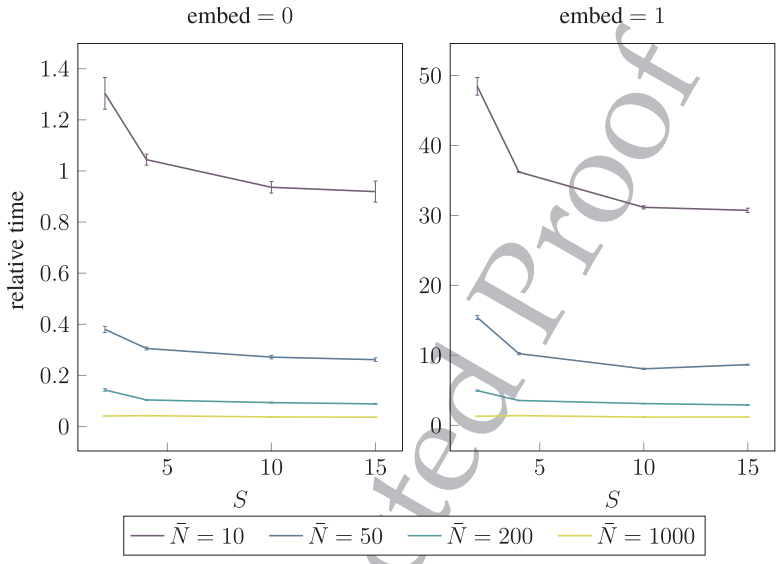


Figure 7: Mean timing factor (relative to ℓ). For long sequences, the proposed composite likelihood-based method is superior to the log-likelihood calculations. It is also evident that the embedding procedure is quite costly because of the higher number of model estimations and score function evaluations.

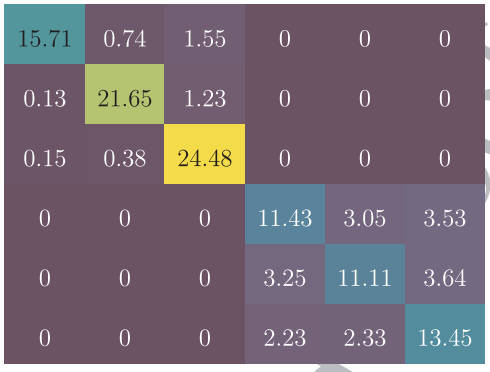
Table 1: Optimal Number of Hidden States Obtained via Five-Fold Cross-Validation on the Training Data Set.

Walking	13
Walking-upstairs	6
Walking-downstairs	4
Sitting	4
Standing	3
Lying	3

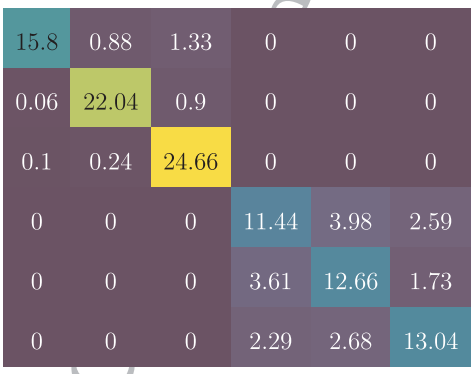
80 random repetitions of the experiment. Thus, variation in results are due to random initializations of HMM parameters and of cluster centers in the quantization process.

The mean (microaveraged)F_1 scores for ℓ and $\mathcal{D}$, respectively, are $\frac{1}{80}\sum_{i=1}^{80} F_1(\ell^{(i)}) = 0.8152$ and $\frac{1}{80}\sum_{i=1}^{80} F_1(\mathcal{D}^{(i)}) = 0.8303$. Although these two numbers seem very close, both score functions are applied to the same test data and class-conditional models at each random repetition. Hence we are dealing with paired samples of F_1, which enables us to evaluate the pairwise differences instead of two separate measures.

6.1 Paired-Samples Binomial Sign Test. To assess whether the difference is significant, we perform a one-sided paired samples sign test with



(a) Score function: ℓ



(b) Score function: $\mathcal{D}$, $\epsilon = 0.0001$

Figure 8: Mean confusion tables from 80 repetitions of the UCI HAR classification task. The colors are based on the values in the figure (yellow: larger values; blue: smaller values) and are meant to visually clarify the structure in the matrices.

the null hypothesis $F_1(\ell) \geq F_1(\mathcal{D})$ and the alternative hypothesis that $F_1(\ell) \leq F_1(\mathcal{D})$. The number of pairs where $F_1(\ell^{(i)}) < F_1(\mathcal{D}^{(i)})$ is 49, and the opposite is 12. This results in a p-value of $9.85 \cdot 10^{-7}$, that is, the probability of observing 12 or fewer negative differences if the null hypothesis is true. The result of this test suggests that $\mathcal{D}$ performs slightly better than ℓ for this particular classification problem. As discussed in section 5, the true likelihood is the optimal score function if the class-conditional models are correct. The performance of the composite likelihood for this particular problem suggests that representing the HMM by its time-dependent third-order moments can lead to better classification performance in cases exhibiting certain

imbalances between training data and test data regarding the number of examples and their distributional properties.

7 Conclusion

We have proposed a new score function for use in hidden Markov model-base classification problems, dominated by long sequences of discrete observations. The score is based on expectations of triplets along a Markov chain, and can be interpreted as a composite likelihood for a moment-based hidden Markov model representation. We show how the memory requirements of the proposed method can be controlled by considering the convergence time of Markov chains. Finally, we show that the proposed score performs at least on par with the commonly used likelihood-based score, but at a substantially reduced computation time in classification of long data sequences.

Appendix A: Construction of Simulated HMM Classification Problems

Each column in the transition matrices is constructed by a single draw from a Dirichlet distribution with base measure $\boldsymbol{\alpha} = \sum_{j=1}^{S} \alpha_j$ and concentration parameter σ. To be able to control the diagonal structure of the transition matrices, the distribution of the ith column, $\boldsymbol{T}_i$, is sampled from Dirichlet distribution with the base measure given by

$$\alpha_j = \begin{cases} T_{\text{diag}} & j = i \\ \frac{1-T_{\text{diag}}}{S-1} & j \neq i \end{cases},$$

where $T_{\text{diag}} \in]0, 1[$

The interpolation parameter $\rho \in]0, 1]$ controls the variance of the simulated multinomial elements. We let ρ determine the relative size of the variance to a maximum variance $v_{\max}$, which is determined by a given minimal concentration parameter $\sigma_{\min}$.

Although the variances of diagonal and off-diagonal elements in general are different, the relation between σ and ρ is independent of the base measure and is given by

$$\sigma = \frac{\sigma_{\min} + 1}{\rho} - 1.$$

In our experiment, the columns of $\boldsymbol{T}$ are generated using $\sigma_{\min} = S$. Hence, the set of most unrelated models is drawn using $\sigma = S$, which is obtained by setting $\rho = 1$, and for $\rho \to 0$, the Dirichlet distribution becomes the Dirac delta function: $\delta(\boldsymbol{\alpha})$.

Appendix B: Embedding Algorihm

Algorithm 2: Classification via Sequence Embedding.

Input: Training and test sequences: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_{\text{train}}}\}$ and $\{\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \ldots, \bar{\boldsymbol{x}}_{N_{\text{test}}}\}$. Training labels: $(y_1, y_2, \cdots, y_{N_{\text{train}}})$. Distance score function $D(\boldsymbol{x}, \mathcal{M})$ relating a sequence $\boldsymbol{x}$ to a model $\mathcal{M}$. Classification algorithm C.

Output: Test labels: $(\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_{N_{\text{test}}})$

---

for $i = 1$ **to** N_{train} **do**

Estimate an HMM $\mathcal{M}_i$ from $\boldsymbol{x}_i$

end for

for $i = 1$ **to** N_{train} **do**

for $j = 1$ **to** N_{train} **do**

Calculate $D(\boldsymbol{x}_i, \mathcal{M}_j)$

end for

end for

Train C using the (normalized) distance scores of the training sequences for all estimated models as features and $(y_1, y_2, \cdots, y_{N_{\text{train}}})$ as labels

for $i = 1$ **to** N_{test} **do**

for $j = 1$ **to** N_{train} **do**

Calculate $D(\bar{\boldsymbol{x}}_i, \mathcal{M}_j)$

end for

$\bar{y}_i = C(\bar{\boldsymbol{x}}_i)$

end for

Acknowledgments

This work was supported in part by Innovation Fund Denmark under the CoSound project, case number 0603-00475B. This publication reflects only our own views.

References

Anandkumar, A., Ge, R., & Hsu, D. (2014). Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research, 15*, 2773–2832. http://dl.acm.org/citation.cfm?id=2697055

Anandkumar, A., Hsu, D., & Kakade, S. (2012). A method of moments for mixture models and hidden Markov models. In *JMLR: Workshop and Conference Proceedings, 23*, 1–31. arXiv:1203.0683v3.

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 24–26). http://www.i6doc.com/en/livre/?GCOI=28001100131010

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, *41*(1), 164–174. doi: 10.1214/09-STS284

Besag, J. (1975). Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D*, *24*(3), 179–195.

Bicego, M., Murino, V., & Figueiredo, M. a. T. (2004). Similarity-based classification of sequences using hidden Markov models. *Pattern Recognition*, *37*(12), 2281–2291. doi: 10.1016/j.patcog.2004.04.005

Dempster, a. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, *39*(1), 1–38. doi: 10.1.1.133.4884

Durrett, R. (2007). *Random graph dynamics.* Cambridge: Cambridge University Press.

Elkan, C. (2013). Using the triangle inequality to accelerate K-means. In *Proceedings of the International Conference on Machine Learning* (pp. 147–153). Cambridge, MA: AAAI Press.

Fill, J. A. (1991). Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *Annals of Applied Probability, 1*, 62–87. http://www.jstor.org/stable/2959625

García-García, D., Emilio, H. P., & Díaz-de-María, F. (2009). A new distance measure for model-based sequence clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(7), 1325–1331. http://e-archivo.uc3m.es/handle/10016/8978

Hsu, D., Kakade, S. M., & Zhang, T. (2012). A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, *78*(5), 1460–1480. arXiv:0811.4413v6

Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics*, *80*(1), 221–239.

Mouret, M., Solnon, C., & Wolf, C. (2009). Classification of images based on hidden Markov models. In *Proceedings of the 2009 Seventh International Workshop on Content-Based Multimedia Indexing*. doi: 10.1109/CBMI.2009.22

Oates, T., Firoiu, L., & Cohen, P. R. (1999). Clustering time series with hidden Markov models and dynamic time warping. In *Proceedings of the IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning.* Bethesda, MD: Institute of Mathematical Statistics.

Troelsgaard, R., & Hansen, L. K. (2016). *Spectral learning of hidden Markov models in non-stationary data*. Manuscript submitted for publication.

Wang, L., Mehrabi, M. G., & Kannaatey-Asibu, E. (2002). Hidden Markov model-based tool wear monitoring in turning. *Journal of Manufacturing Science and Engineering*, *124*(3), 651. 10.1115/1.1475320

Wong, W., & Stamp, M. (2006). Hunting for metamorphic engines. *Journal in Computer Virology*, *2*(3), 211–229. doi: 10.1007/s11416-006-0028-7

Xing, Z., Pei, J., & Keogh, E. (2010). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*. Retrieved from http://dl.acm.org/citation.cfm?id=1882478

Received October 24, 2016; accepted July 31, 2017.