Nonlin. Processes Geophys., 13, 339–352, 2006
www.nonlin-processes-geophys.net/13/339/2006/

Nonlinear Processes in Geophysics

Time series segmentation with shifting means hidden markov models

Ath. Kehagias[1] **and V. Fortin**[2]

[1]Aristotle University of Thessaloniki, School of Engineering, GR 541 24 Thessaloniki, Greece
[2]Numerical Weather Prediction Research, Meteorological Research Division, Environment Canada, Dorval, Quebec H9P 1J3, Canada

Received: 10 February 2006 – Revised: 24 May 2006 – Accepted: 31 May 2006 – Published: 1 August 2006

Abstract. We present a new family of hidden Markov models and apply these to the segmentation of hydrological and environmental time series. The proposed hidden Markov models have a discrete state space and their structure is inspired from the *shifting means* models introduced by Chernoff and Zacks and by Salas and Boes. An estimation method inspired from the EM algorithm is proposed, and we show that it can accurately identify multiple change-points in a time series. We also show that the solution obtained using this algorithm can serve as a starting point for a Monte-Carlo Markov chain Bayesian estimation method, thus reducing the computing time needed for the Markov chain to converge to a stationary distribution.

1 Introduction

The problem which motivates the current paper is the segmentation of hydrological and environmental time series. Our main goal is to develop a *fast segmentation algorithm*; an important secondary goal is to preserve a connection to the *shifiting means* model (SMM) and point of view.

The segmentation problem has been attacked by various methods. For instance, an extensive discussion of *sequential* segmentation methods appears in (Hipel and McLeod, 1994, pp. 655–733). As for *nonsequential* methods, some early papers are Lee and Heghinian (1977), Cobb (1978), and Buishand (1982, 1984); more recent work includes Hoppe and Kiely (1999); Kiely et al. (1998) and Paturel et al. (1997); Servat et al. (1997) and (from the Bayesian point of view) Perreault et al. (1999, 2000a,b,c) and Rao and Tirtotjondro (1996); there are many more examples; here we only give a small sample of the literature. All of the above references deal with a *single* change point. Segmentation algorithms for *multiple* change points have been presented by Hubert (1997, 2000), Fortin et al. (2004a,b), Kehagias (2004) and Kehagias et al. (2006).

Hidden Markov models (HMM) have been applied to various time series segmentation problems, especially in the engineering and computer science literature; two good reviews are Bengio (1998) and Rabiner (1988). An application of HMM's to hydrological segmentation appears in Kehagias (2004).

SMM's have been used to model time series with change points; we consider here two SMM variants, the first introduced by Chernoff and Zacks (1964) and the second by Salas and Boes (1980) and Yao (1988). The Salas/Yao model has been used in several papers as the basis for segmentation of hydrological time series. In particular, Fortin et al. (2004a) use a SMM to model a hydrological time series with change points and applies Markov Chain Monte Carlo (MCMC) methods to estimate the model parameters, as well as the posterior distribution of the change points. Fortin's solution is highly informative, but also computationally intensive.

As will become clear in the sequel, a SMM is a HMM of special type (see also Fortin et al., 2004a). However, there is an important difference between SMM's and more "traditional" HMM's. Namely, a "traditional" HMM is characterized by a *finite state* underlying Markovian process (Markov chain), while the underlying Markovian process of a SMM has an (uncountably) *infinite* state space. This creates considerable computational complications for the application of typical HMM algorithms (such as Baum-Welch estimation) to the time series segmentation problem. More generally, *Maximum Likelihood* (ML) estimation of the SMM is not a trivial problem (hence the use of MCMC in Fortin et al. (2004a) for estimating the parameters of SMM's). However, if the SMM could be emulated by an HMM, then ML estimation of the HMM could be performed quite efficiently.

In the current paper we present a fusion of "traditional" HMM's and SMM's. Namely, we present (four variants of) HMM's which are designed to emulate SMM's and yet have a finite state underlying Markov chain. The advantage of

Correspondence to: V. Fortin
(vincent.fortin@ec.gc.ca)

Published by Copernicus GmbH on behalf of the European Geosciences Union and the American Geophysical Union.

finite state is that typical HMM algorithms (which are computationally very efficient) can be used. The SMM emulation yields an additional advantage, namely a HMM with a smaller number of states (than, for example, the left-to-right HMM's used in Kehagias (2004)). In addition, the ML estimate can be used as a starting point for the MCMC algorithm proposed by Fortin et al. (2004a).

The paper is organized as follows. In Sect. 2 we present the two SMM's which we study in this paper and four variants of "SMM-inspired" HMM's; in Sect. 3 we present a parameter estimation/segmentation algorithm for the HMM's of the previous sections; in Sect. 4 we present some segmentation experiments; in Sect. 5 we consider the application of the HMM to initialize the MCMC estimation of the SMM; finally, in Sect. 6, we summarize and discuss our results.

2 Shifting means and hidden Markov models

2.1 Shifting means models

As we already mentioned, at least two variants of the SMM appear in the literature. We proceed to briefly describe each of them.

SMM-1. This model was introduced by Chernoff and Zacks (1964). It can be described by the following equations (for t=1, 2, ...).

$$x_t = m_t + \varepsilon_t, \qquad m_t = m_{t-1} + z_{t-1} \cdot \delta_t$$

where

1. $\varepsilon_1, \varepsilon_2, ...$ are independent and identically distributed (iid) random variables with normal distribution $\mathcal{N}(0, \sigma_\epsilon)$;
2. $z_1, z_2, ...$ are iid random variables taking the values 0, 1 with probabilities $\eta = \Pr(z_t=1)$, $1-\eta = \Pr(z_t=0)$;
3. $\delta_1, \delta_2, ...$ are iid random variables with normal distribution $\mathcal{N}(0, \sigma_\mu)$.

In other words, the means process m_t is controlled by the process z_t: when z_{t-1}=0, m_t is the same as m_{t-1} (this is what happens "most of the time" if η is close to zero); when z_{t-1}=1, then we have a change: m_{t-1} is incremented by the normal random variable δ_t to obtain m_t. The process x_t is a noisy observation of m_t. The *model parameter vector*, denoted by θ, is

$$\theta = (\eta, \sigma_\varepsilon, \sigma_\mu).$$

SMM-2. This model was studied by Salas and Boes (1980), Yao (1988) and others. It can be described by the following equations

$$x_t = m_t + \varepsilon_t, \qquad m_t = (1 - z_{t-1}) \cdot m_{t-1} + z_{t-1} \cdot (\mu + \delta_t).$$

The processes $\varepsilon_1, \varepsilon_2, ..., z_1, z_2, ...$ and $\delta_1, \delta_2, ...$ have the same properties as in SMM-1. Hence SMM-2 behaves very similarly to SMM-1, but when the mean m_t changes it takes its new value according to a Gaussian law $\mathcal{N}(\mu, \sigma_\mu)$ *independently* of m_{t-1}. The *model parameter vector* is

$$\theta = (\eta, \mu, \sigma_\varepsilon, \sigma_\mu).$$

It is worth emphasizing that a SMM actually *is* a HMM. Evidently, the pair (m_t, z_t) is a Markov process, hence $((m_t, z_t), x_t)$ is a hidden Markov model. However, m_t takes values in $\mathbb{R}$ and z_t takes values in $\{0, 1\}$, hence the possible *states* (i.e. values) of (m_t, z_t) are *uncountably infinite*. This fact creates considerable difficulties for estimation and segmentation. "Standard" HMM's have a finite number of states; as a result one can perform ML parameter estimation by the Baum-Welch (EM) algorithm. It is not immediately obvious how to apply EM to the SMM. Recall that EM is based on an alternating sequence of Expectation and Maximization steps; in the HMM case, the maximization step involves the use of dynamic programming (DP) to compute an optimal state sequence; but, when the state space is infinite, DP is not feasible because one must evaluate the cost of an infinite number of possible (m_{t-1}, m_t) pairs.

2.2 Hiden Markov models which emulate SMM's

We can handle the infinite dimensionality of SMM's by a finite-dimensional approximation. The standard way to do this is by replacing $\mathbb{R}$ with a finite set $\mathbf{S}=\{\mu_1, \mu_2, ..., \mu_K\}$ (we number the states so that $\mu_1<\mu_2<...<\mu_K$); then we can estimate the parameters of the discretized HMM and perform time series segmentation by standard HMM methods. However, a "good"approximation requires that the μ_k's are packed sufficiently close and the endpoints μ_1, μ_K are such that (the original) m_t stays in $[\mu_1, \mu_K]$ with high probability (practically equal to one). Hence K, the number of states, must be sufficiently large.

We follow a different approach, which results in a more economical model (smaller K). We actually test two variations of this basic approach, and for each variation we need two slightly different HMM's, one to approximate SMM-1 and another to approximate SMM-2. This results in four variants of the basic model; the only difference between the models is in the state transition probabilities. Hence the first part of our description applies to all models. We introduce a Markovian stochastic process $s_1, s_2, ...$ taking values in $\mathbf{S}=\{1, 2, ..., K\}$. To each state we associate a *parameter* μ_k (k=1, 2, ..., K). We also assume that the *conditional* probability of x_t, given s_t=k, is $\mathcal{N}(\mu_k, \sigma_\varepsilon)$. In other words, the *emission* function $\mathbf{f}(x) = [f_k(x)]_{k=1}^K$ (where $f_k(x)$ is the density of x_t conditional on $s_t = k$) is defined as follows (for k=1, 2, ..., K):

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_\varepsilon} \exp\left\{-\frac{1}{2} \cdot \left(\frac{x - \mu_k}{\sigma_\varepsilon}\right)^2\right\} \tag{1}$$

Hence (s_t, x_t) is an HMM. To obtain a full description of the model it remains to specify the *state transition matrix* $\mathbf{P}$. We will present four different choices, each of them being (to some degree) similar to the original SMM models. All models use a $\mathbf{P}=\left[P_{jk}\right]_{j,k=1}^{K}$ (for j, k=1, 2, ..., K):

$$P_{jk}=\begin{cases}(1-\eta)+\eta\cdot g_{jk} & \text{if } j=k\\ \eta\cdot g_{jk} & \text{if } j\neq k\end{cases},\ (j,k=1,2,...,K) \tag{2}$$

the only difference being in the quantities g_{jk}. We now present our four choices for g_{jk}.

HMM-1. In the first model, henceforth referred to as HMM-1, we define the quantity g_{jk} by:

$$\begin{aligned}&g_{jk}=c_j\cdot e^{-\left(\mu_k-\mu_j\right)^2/2\sigma_\mu^2},\\ &c_j=\left(\sum_{k=1}^{K}e^{-\left(\mu_k-\mu_j\right)^2/2\sigma_\mu^2}\right)^{-1}\ (j,k=1,2,...,K)\end{aligned} \tag{3}$$

This model is a very rough, "pointwise"approximation of SMM-1. In other words, we approximate the transition from state j to state k by the value of the normal distribution (with mean μ_j) at the point μ_k (and then we have to normalize the transition probabilities). The parameter vector of the model is $\theta=\left(\eta,\sigma_\varepsilon,\sigma_\mu,\mu_1,\mu_2,...,\mu_K\right)$

HMM-2. In the second model, henceforth referred to as HMM-2, we define the quantity g_{jk} as follows:

$$\begin{aligned}&g_{j1}=\frac{1}{\sqrt{2\pi}\sigma_\mu}\int_{-\infty}^{\frac{\mu_2+\mu_1}{2}}e^{-\left(z-\mu_j\right)^2/2\sigma_\mu^2}dz\\ &g_{jk}=\frac{1}{\sqrt{2\pi}\sigma_\mu}\int_{\frac{\mu_k+\mu_{k-1}}{2}}^{\frac{\mu_{k+1}+\mu_k}{2}}e^{-\left(z-\mu_j\right)^2/2\sigma_\mu^2}dz\ (k=2,3,...,K-1)\\ &g_{jK}=\frac{1}{\sqrt{2\pi}\sigma_\mu}\int_{\frac{\mu_K+\mu_{K-1}}{2}}^{\infty}e^{-\left(z-\mu_j\right)^2/2\sigma_\mu^2}dz\end{aligned} \tag{4}$$

(we assume that the states are numbered so as to ensure $\mu_1<\mu_2<...<\mu_K$). This model is an "interval-based" approximation of SMM-1. Namely, we correspond to the k-th state of the model the interval $\left[\frac{\mu_{k-1}+\mu_k}{2},\frac{\mu_{k+1}+\mu_k}{2}\right]$ (with the obvious modification for the 1-st and K-th states) and then compute the corresponding transition probabilities as integrals of normal distributions. The parameter vector of the model is $\theta=\left(\eta,\sigma_\varepsilon,\sigma_\mu,\mu_1,\mu_2,...,\mu_K\right)$

HMM-3. In the third model, henceforth referred to as HMM-3, we define the quantity g_{jk} by:

$$\begin{aligned}&g_{jk}=c_j\cdot e^{-(\mu_k-\mu)^2/2\sigma_\mu^2},\\ &c_j=\left(\sum_{k=1}^{K}e^{-(\mu_k-\mu)^2/2\sigma_\mu^2}\right)^{-1}\ (j,k=1,2,...,K)\end{aligned} \tag{5}$$

This model is a pointwise approximation of SMM-2. The parameter vector of the model is $\theta=\left(\eta,\sigma_\varepsilon,\sigma_\mu,\mu,\mu_1,\mu_2,...,\mu_K\right)$

HMM-4. In the fourth model, henceforth referred to as HMM-4, we define the quantity g_{jk} by:

$$\begin{aligned}&g_{j1}=\frac{1}{\sqrt{2\pi}\sigma_\mu}\int_{-\infty}^{\frac{\mu_2+\mu_1}{2}}e^{-(z-\mu)^2/2\sigma_\mu^2}dz\\ &g_{jk}=\frac{1}{\sqrt{2\pi}\sigma_\mu}\int_{\frac{\mu_k+\mu_{k-1}}{2}}^{\frac{\mu_{k+1}+\mu_k}{2}}e^{-(z-\mu)^2/2\sigma_\mu^2}dz\ (k=2,3,...,K-1)\\ &g_{jK}=\frac{1}{\sqrt{2\pi}\sigma_\mu}\int_{\frac{\mu_K+\mu_{K-1}}{2}}^{\infty}e^{-(z-\mu)^2/2\sigma_\mu^2}dz\end{aligned} \tag{6}$$

(we assume that the states are numbered so as to ensure $\mu_1<\mu_2<...<\mu_K$). This model is an interval-based approximation of SMM-2. Note that g_{jk} only depends on k. The parameter vector of the model is $\theta=\left(\eta,\sigma_\varepsilon,\sigma_\mu,\mu,\mu_1,\mu_2,...,\mu_K\right)$

3 The segmentation algorithm

The "basic" algorithm presented below applies to all the HMM's of the Sect. 2. The algorithm takes as *input* an initial estimate of the number of states $K^{(0)}$ and a scaling parameter λ; it gives as *output* estimates $\widehat{\mathbf{t}}$ (the segmentation), $\widehat{\eta}$ (the escape probability), and $\widehat{\mu}_1, \widehat{\mu}_2$, ... (the segment means). Four variants of the basic algorithm (one for each of the four HMM's) are obtained by choosing a particular formula for the reestimation of the transition matrix **P** (one of Eqs. 3–6).

The set of states (i.e. values of s_t) will be denoted by **S**, and a *state sequence* by $\mathbf{s}=(s_1,s_2,...,s_t)$. The number of states will be denoted by K and the number of segments by $\widetilde{K}$; these are **not** necessarily the same. An i superscript indicates the *estimate* of the respective quantity at the i-th step of the algorithm; for instance K is the *true* number of states, while $K^{(i)}$ is the estimate of K at the i-th step.

Before giving a listing of the algorithm, let us describe it in broad terms

1. The "total" mean μ and total standard deviation σ_x are estimated only once, during initialization; see Eq. (7).

2. The standard deviations σ_μ and σ_ε are *fixed* during initialization, as fractions of the total σ_x:

$$\sigma_\varepsilon=\lambda\cdot\sigma_x,\qquad \sigma_\mu=\sqrt{1-\lambda^2}\cdot\sigma_x.$$

   This is consistent with the constraint σ_x^2=$\sigma_\mu^2+\sigma_\varepsilon^2$. Note that λ is a *parameter* of the algorithm; because the algorithm performs robustly for a broad range of λ values, the exact values to be used are not critical. We have found by extensive experimentation (some of the relevant experiments will be presented in Sect. 4) that $\lambda\in[0.35,0.55]$ works well for a broad range of time series. All the experiments we report here use λ=0.4.[1]

3. Another parameter which must be chosen is $K^{(0)}$, the initial number of segments; again, the algorithm performs robustly for the range $K^{(0)}\in[5,15]$. All the experiments we report here use $K^{(0)}$=10.

[1] Recall that λ determines the "estimates" of σ_μ and σ_ε. Of course λ could be estimated from the data, rather than being predetermined. An alternative approach, which does not use λ at all, would be to directly reestimate (at every iteration) σ_μ from the residuals of the *segmented* x_t process and σ_ε from the residuals of the *estimated* m_t process. We have tried both these approaches and obtained results were not as good as the ones we present here, using the fixed λ parameter (these results are not presented in the current paper, because of lack of space).

4. After initialization, the main, iterative part of the algorithm is performed. The i-th iteration consists of three parts.

   (a) First estimates $\eta^{(i)}$, $\mu_1^{(i)}$, $\mu_2^{(i)}$, ... are obtained from the previous segmentation $\mathbf{t}^{(i-1)}$.

   (b) Then $\eta^{(i)}$, $\mu_1^{(i)}$, $\mu_2^{(i)}$, ... are used to recompute the $g_{jk}^{(i)}$'s and from these the transition matrix $\mathbf{P}^{(i)}$.

   (c) Finally, a new state sequence $\mathbf{s}^{(i)}$ and segmentation $\mathbf{t}^{(i)}$ are computed using the the transition matrix $\mathbf{P}^{(i)}$ and the Viterbi algorithm.

Algorithm 1 Segmentation algorithm

```
Require: Observations $\{x_t, t=1, 2, ..., T\}$,
  Initial number of states $K^{(0)}$,
  Scaling parameter $\lambda$,
  Maximum number of iterations $i_{\max}$.
Ensure: Estimated segmentation $\widehat{\mathbf{t}}=\mathbf{t}^{(i)}$,
  Estimated escape probability $\widehat{\eta}=\eta^{(i)}$,
  Estimated segment means $\widehat{\mu}_k=\mu_k^{(i)}$
  ($k=1, 2, ..., \widehat{K}$ where $\widehat{K} = K^{(i)}$).
  Initialization
  Initialize means and variances

  $$\mu = \frac{\sum_{t=1}^T x_t}{T}, \qquad \sigma_x = \sqrt{\frac{\sum_{t=1}^T (x_t-\mu)^2}{T-1}}, \tag{7}$$

  $$\sigma_\varepsilon = \lambda \cdot \sigma_x, \qquad \sigma_\mu = \sqrt{1-\lambda^2} \cdot \sigma_x \tag{8}$$

  Initialize the state set, state sequence, segmentation, number of segments

  $$\mathbf{S}^{(0)} = \left\{1, 2, ..., K^{(0)}\right\}, \tag{9}$$

  $$s_t^{(0)} = k \qquad \text{iff} \qquad \frac{k-1}{K^{(0)}} \cdot T \le t < \frac{k}{K^{(0)}} \cdot T, \tag{10}$$

  $$\begin{aligned} \mathbf{t}^{(0)} &= \left\{t_0^{(0)}, t_1^{(0)}, ..., t_{\widetilde{K}^{(0)}-1}^{(0)}, t_{\widetilde{K}^{(0)}}^{(0)}\right\} \\ &= \{0\} \cup \left\{t : s_{t-1}^{(0)} \neq s_t^{(0)}\right\} \cup \{T\}, \end{aligned} \tag{11}$$

  $$\widetilde{K}^{(0)} = \text{card}\left(\mathbf{t}^{(0)}\right) - 1. \tag{12}$$

Main
for $i=1..i_{\max}$ do
  Estimate the escape probability and the segment means

  $$\eta^{(i)} = \frac{\widetilde{K}^{(i-1)}}{T}, \tag{13}$$
  $$\mu_k^{(i)} = \frac{\sum_{t:s_t^{(i-1)}=k} x_t}{\sum_{t:s_t^{(i-1)}=k} 1} \qquad (k = 1, 2, ..., K^{(i-1)}). \tag{14}$$

  Estimate $g_{jk}^{(i)}$ using $\mu_1^{(i)}, \mu_2^{(i)}, ..., \sigma_\mu$ and the appropriate formula (HMM-1 is updated using Eq. (3); HMM-2 is updated using Eq. (4) and so on).
  Estimate the transition matrix $\mathbf{P}^{(i)}$ using $\eta^{(i)}, g_{jk}^{(i)}$:

  $$P_{jk}^{(i)} = \begin{cases} \left(1-\eta^{(i)}\right) + \eta^{(i)} \cdot g_{jk}^{(i)} & \text{if } j = k \\ \eta^{(i)} \cdot g_{jk}^{(i)} & \text{if } j \neq k \end{cases} \quad (j, k = 1, ..., K^{(i-1)}) \tag{15}$$

  Estimate the state sequence $\mathbf{s}^{(i)}$, using Viterbi algorithm (Forney (1973)) with $\mathbf{P}^{(i)}, \sigma_\varepsilon, \mu_1^{(i)}, ..., \mu_{K^{(i-1)}}^{(i)}$.
  Estimate the segmentation and number of segments:

  $$\begin{aligned} \mathbf{t}^{(i)} &= \left\{t_0^{(i)}, t_1^{(i)}, ..., t_{K^{(i)}-1}^{(i)}, t_{K^{(i)}}^{(i)}\right\} \\ &= \{0\} \cup \left\{t : s_{t-1}^{(i)} \neq s_t^{(i)}\right\} \cup \{T\} \end{aligned} \tag{16}$$

  $$\widetilde{K}^{(i)} = \text{card}\left(\mathbf{t}^{(i)}\right) - 1 \tag{17}$$

  Compute the set of states and its cardinality

  $$\mathbf{S}^{(i)} = \left\{k : \text{there is some } t \text{ such that } s_t^{(i)} = k\right\} \tag{18}$$
  $$K^{(i)} = \text{card}\left(\mathbf{S}^{(i)}\right) \tag{19}$$

  Renumber the states so that

  $$\mathbf{S}^{(i)} = \left\{1, 2, ..., K^{(i)}\right\} \tag{20}$$

  Adjust $\mathbf{s}^{(i)}$, $\mu_k^{(i)}$ and $\mathbf{P}^{(i)}$ according to the renumbering of states.
end for
```

The successive reestimations of parameters and state follow the basic EM idea but our algorithm is only an *approximate* EM algorithm (from this point on we will call it *pseudo-EM*). While the state $\mathbf{s}$ is estimated optimally (by Viterbi) in every stage of the algorithm and the same is true of η, the μ_1, ..., μ_K estimates are not optimal. Exact determination of the optimal μ values requires the solution of a difficult system of nonlinear equations. However, the formula (14) seems intuitively plausible and apparently works well in practice (see the experiments of Sect. 4).

The *order* of the model is the number of its free parameters; upon convergence it will be $\widehat{K} + 3$ (for HMM-1 and

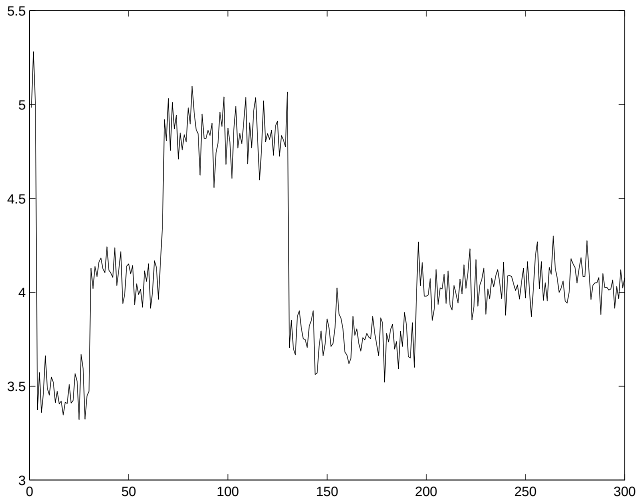

Fig. 1. A realization of the time series used in Experiment no. 1, at σ_ε=0.10 (breaks at 3, 30, 67, 130, 194).

HMM-2) or $\widehat{K}+4$ (for HMM-3 and HMM-4). Hence *our algorithm automatically determines the model order*. This is a particularly nice feature of the algorithm; compare with other algorithms (Hubert, 1997, 2000; Kehagias, 2004) which require the use of a *model order selection criterion* such as Akaike's information criterion (AIC), the Bayesian information criterion (BIC) or Scheffe's criterion.

4 Experiments

We now evaluate the performance of the segmentation algorithm by applying it to the segmentation of several time series.

4.1 Experiment no. 1: Artificial SMM time deries

In this experiment we use artificial data, which are created according to the SMM-2 model with p=0.99, σ_μ=1, μ_0=4. We generate a time series of length T=300 and use $\sigma_\varepsilon \in \{0.00, 0.05, 0.10, ..., 0.25\}$. Two typical realizations of the noisy time series appear in Fig. 1 (σ_ε=0.10) and Fig. 2 (σ_ε=0.25).

We apply the segmentation algorithm corresponding to HMM-1 with $K^{(0)}$=10 and λ=0.40. We measure segmentation accuracy by *Beeferman's segmentation metric P_k*, which can be loosely interpreted as the probability of an observation being "clearly "misclassified (for a precise definition see Appendix A). We repeat the segmentation 200 times for each value of σ_ε, compute the P_k value and average over the 200 repetitions. Hence we obtain a curve of P_k as a function of σ_ε. We repeat the procedure for HMM-2, HMM-3 and HMM-4. Hence we obtain a total of four curves, which are plotted in Fig. 3.

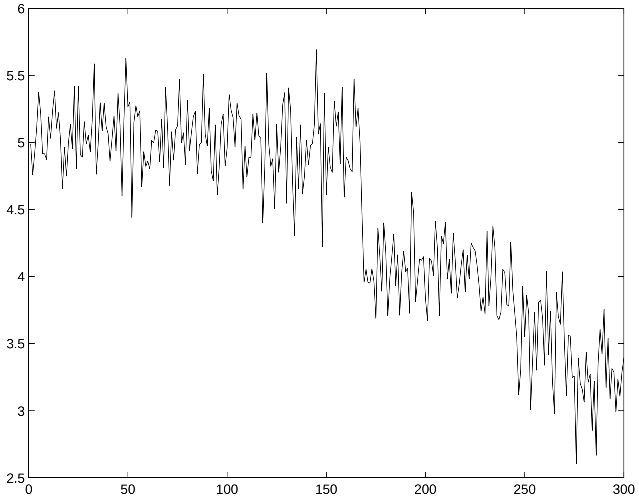

Fig. 2. A realization of the time series used in Experiment no. 1, at σ_ε=0.10 (breaks at 167, 195, 244, 273).

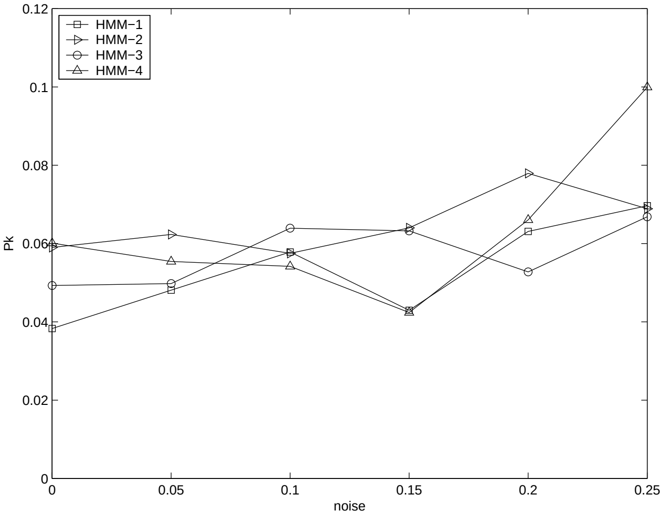

Fig. 3. Plot of P_k vs. σ_ε, as obtained in Experiment no. 1, with λ=0.40.

The low P_k values obtained indicate that the algorithm yields good segmentations; this is true even at the highest noise level used, σ_ε=0.25.

4.1.1 Regarding the choice of algorithm parameters

We have claimed in Sect. 3 that our algorithm performs well for a wide range of λ and $K^{(0)}$ values. We base our claim on extensive experimentation; in this subsection we will present some of our experiments regarding the dependence of P_k on λ and $K^{(0)}$.

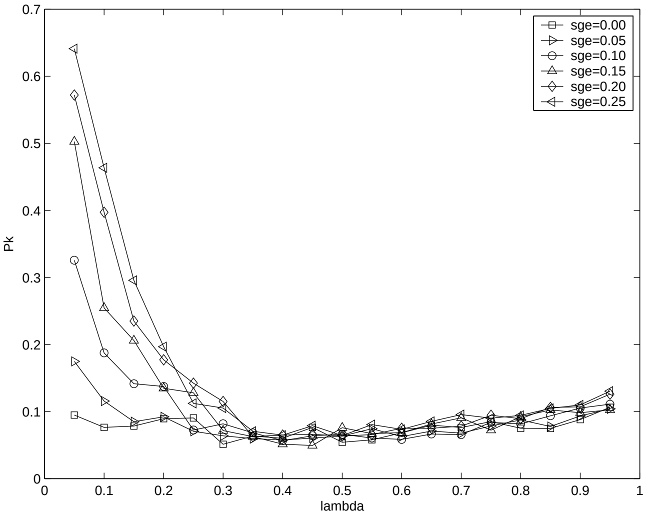

Fig. 4. Plot of P_k vs. λ for various values of σ_ε. The model HMM-1 was used with $K^{(0)}$=5.

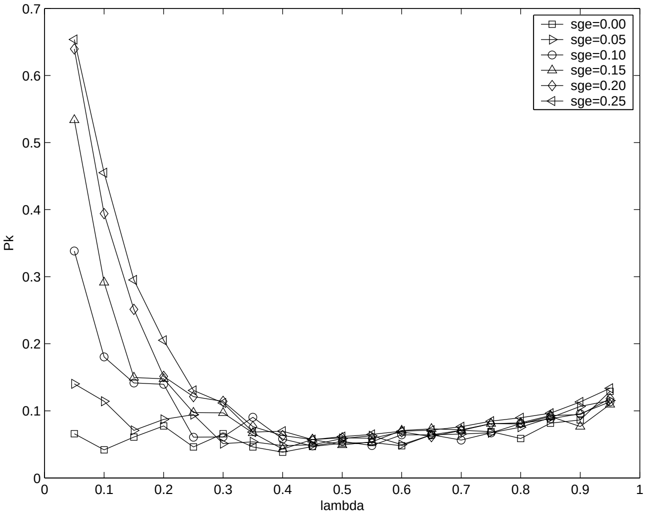

Fig. 5. Plot of P_k vs. λ for various values of σ_ε. The model HMM-1 was used with $K^{(0)}$=10.

In Fig. 4 we present a plot of P_k as a function of λ, for various values of σ_ε. Four curves are plotted, one for each type of HMM. Each point of the curves was obtained in the manner previously decribed (i.e. 200 experiments were performed at every given $(\lambda, \sigma_\varepsilon)$ pair and the resulting P_k was averaged). The value $K^{(0)}$=5 was used. It can be seen that P_k attains its best (smallest) values in the interval [0.35, 0.55].

Figures 5 and 6 present the same results for the values $K^{(0)}$=10 and $K^{(0)}$=15, respectively, and support the same conclusions as above. Note also that, for a given λ value, the P_k values are very similar for $K^{(0)}$=5, 10, 15 (compare Figs. 4–6) ; this indicates that there is no critical dependence of P_k on $K^{(0)}$.

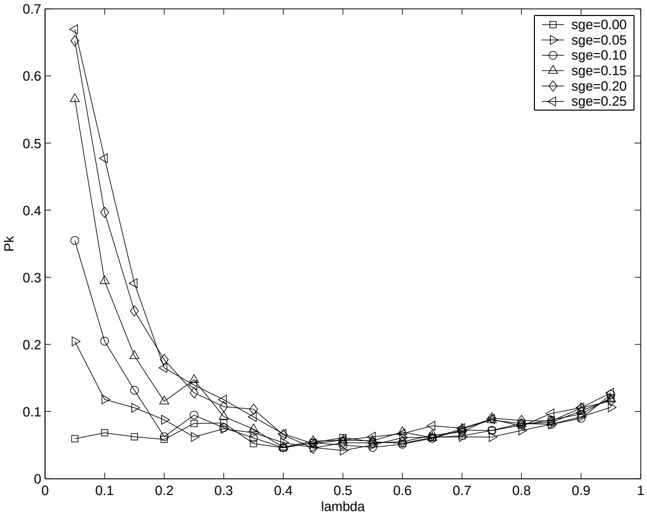

Fig. 6. Plot of P_k vs. λ for various values of σ_ε. The model HMM-1 was used with $K^{(0)}$=15.

Similar results were obtained for HMM-2, HMM-3 and HMM-4. Based on the above figures, we find λ=0.40 to be a good choice, lying in the middle of the flat part of the P_k curve.

One way to assess the impact of using λ=0.40 irrespective of the time series is to compare the performance of the algorithm using λ=0.40 to the one obtained when using $\lambda=\lambda^*$, the true value of λ. For the particular time series used in this section, we have σ_μ=1, $\sigma_\varepsilon=\lambda^*\sigma_x$. Then

$$\sigma_x^2=\sigma_\mu^2+\sigma_\varepsilon^2 \Rightarrow \left(\frac{\sigma_\varepsilon}{\lambda^*}\right)^2=1+\sigma_\varepsilon^2 \Rightarrow \lambda^*=\frac{\sigma_\varepsilon}{\sqrt{1+\sigma_\varepsilon^2}}.$$

For the values σ_ε we use we can build the following table

σ_ε	0.000	0.050	0.100	0.150	0.200	0.250
λ^*	0.000	0.050	0.995	0.148	0.196	0.243

In Fig. 7 the horizontal axis corresponds to noise level σ_ε and the vertical axis corresponds to P_k values. Two P_k curves are plotted; one is obtained using λ=0.40, the second using $\lambda^*(\sigma_\varepsilon)$ (i.e. we performing the segmentation assuming the true value of λ known); the particular curves were obtained from the HMM-1 model with $K^{(0)}=10$). It can be seen, perhaps surprisingly, that the "optimal" λ^* gives worse results than λ=0.40. Similar results were obtained for models HMM-2, HMM-3, HMM-4 and for $K^{(0)}$ equal to 5, 10 and 15. This suggests that the algorithm performs better with a biased parameter estimate for λ but still provides accurate segmentations.

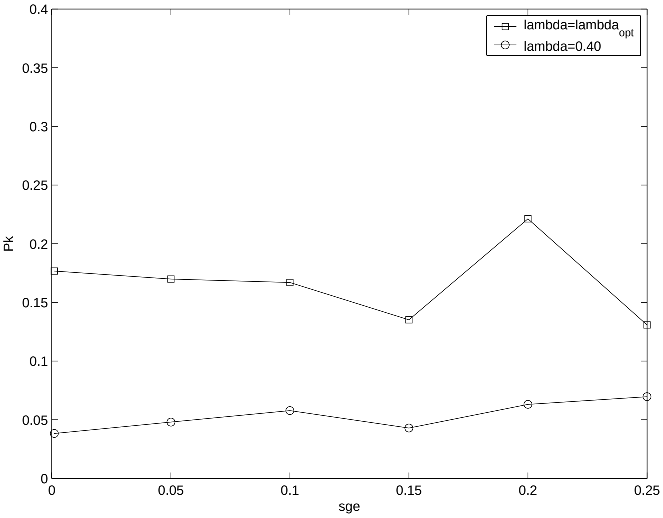

Fig. 7. Plot of P_k vs. σ_ε for λ=0.40 and λ=λ*. The model HMM-1 was used with $K^{(0)}$=10.

4.2 Experiment no. 2: Artificial "handcrafted" time series

Next we investigate the performance of the segmentation algorithm on data which are *not* generated by the SMM mechanism. We use a "handcrafted" time series where we arbitrarily selected the position and length of the segments (i.e. we did not use any particular model). The length of the time series is T=400 and it contains three breaks (four segments) at times 123, 230 and 282. In other words, $\mathbf{t}$ =(0, 123, 230, 282, 400). The mean values of the segments are 0.65, 0.90, 0.60, 0.95. We add to the time series Gaussian, zero mean white noise at various levels: $\sigma_\varepsilon \in \{0.00, 0.05, 0.10, ..., 0.25\}$. Two typical realizations of the noisy time series appear in Fig. 8 (σ_ε=0.05) and Fig. 9 (σ_ε=0.25).

We apply the segmentation algorithm (using HMM-1, ..., HMM-4) with λ=0.40 and $K^{(0)}$=10. We repeat the segmentation 200 times for each value of σ_ε, compute the P_k value and for each of HMM-1, ..., HMM-4 we average over the 200 repetitions Hence we obtain four curves of P_k as a function of σ_ε, which are plotted in Fig. 10

We can see that the segmentation is very accurate at low and medium σ_ε values and remains quite good even at σ_ε=0.25. We have obtained similar results for many other pairs $\left(\lambda, K^{(0)}\right) \in [0.35, 0.55] \times \{5, 10, 15\}$ (not reported here because of space limitations).

Numerical experiments show that the dependence of P_k on λ and $K^{(0)}$ is similar to the one presented in Figs. 4–7 in Sect. 4.1.1. Namely, good segmentations can be obtained for any pair $\left(\lambda, K^{(0)}\right) \in [0.35, 0.55] \times [5, 15]$. Hence, for example, in Fig. 11 we present the dependence of P_k on λ, for the HMM-2 model with $K^{(0)}$=10; in Fig. 12 we compare the performance of the algorithm using λ=0.40 to the one obtained when using λ*, the true value of λ. In both figures we see the good performance of λ=0.40. We omit the presentation of additional results because they are very similar to the ones presented here.

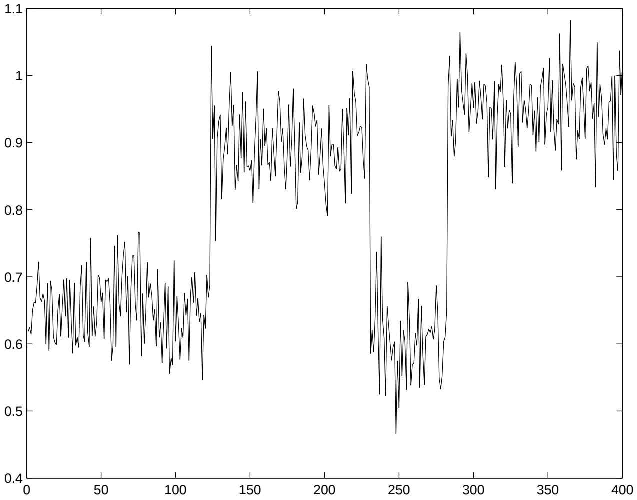

Fig. 8. A realization of the time series used in Experiment no. 2, at σ_ε=0.05 (breaks at 123, 230, 282).

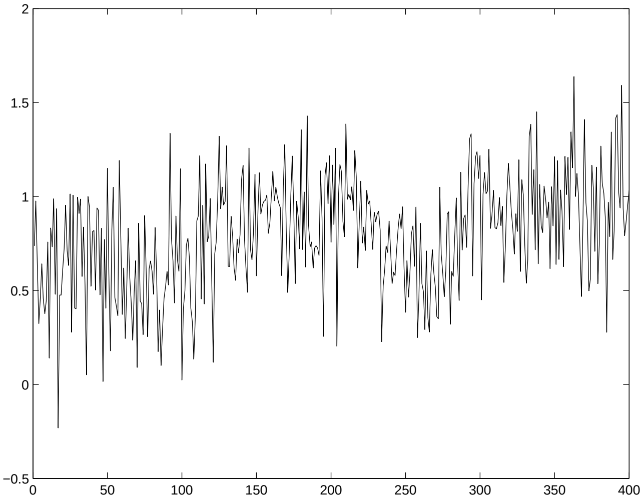

Fig. 9. A realization of the time series used in Experiment no. 2, at σ_ε=0.25 (breaks at 123, 230, 282).

4.3 Experiment no. 3: Artificial random time series

Let us now check the performance of our algorithm on a completely random time series. We take x_t=0.65+ε_t for t=1, 2, ..., 300. This time series contains no breaks and is thus simply a white noise plus a constant component. Ideally, the algorithm should identify that such a time series contains a single segment. We use $\sigma_\varepsilon \in$ {0.0, 0.05, 0.010, 0.15, 0.20, 0.25}.

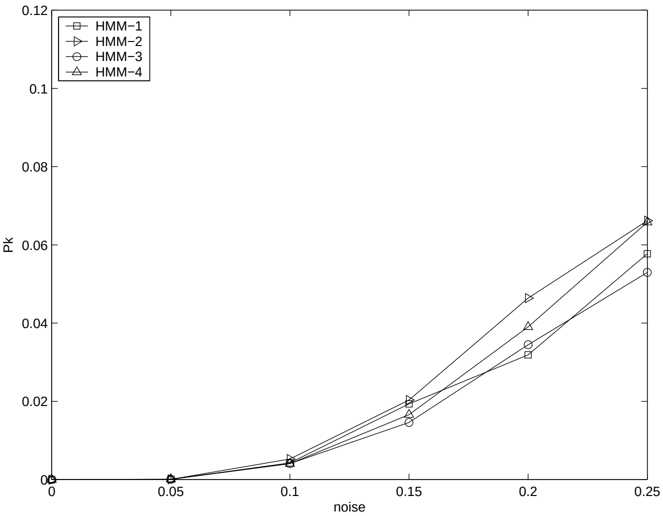

Fig. 10. Plot of P_k vs. σ_ε, as obtained in Experiment no. 2, with λ=0.40.

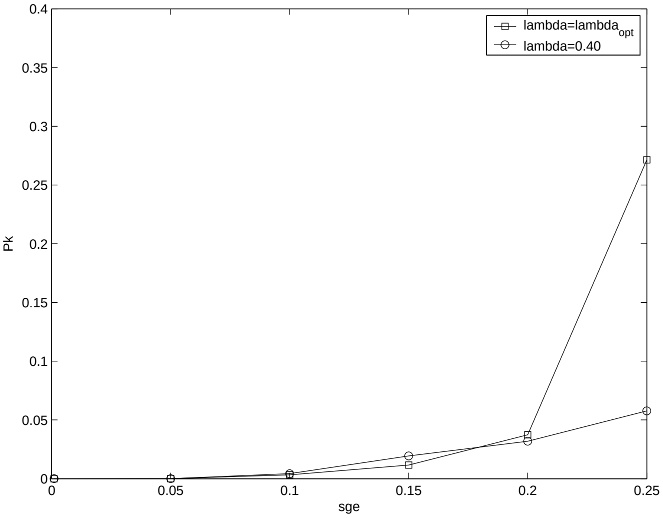

Fig. 12. Plot of P_k vs. σ_ε for λ=0.40 and $\lambda = \lambda^*$. The model HMM-1 was used with $K^{(0)}$=10.

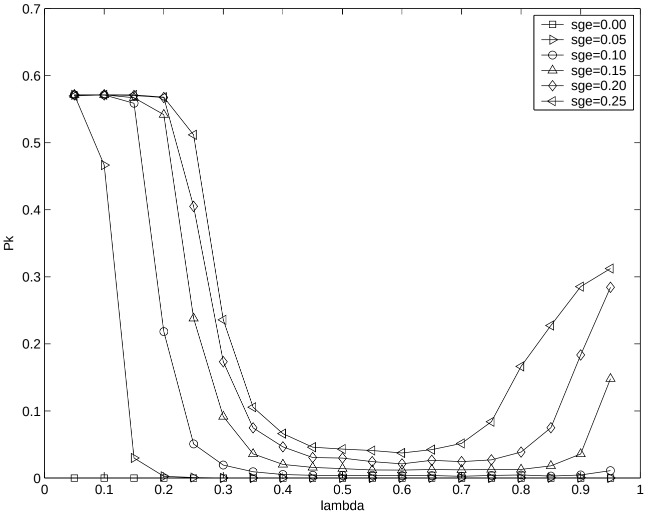

Fig. 11. Plot of P_k vs. λ for various values of σ_ε. The model HMM-2 was used with $K^{(0)}$=10.

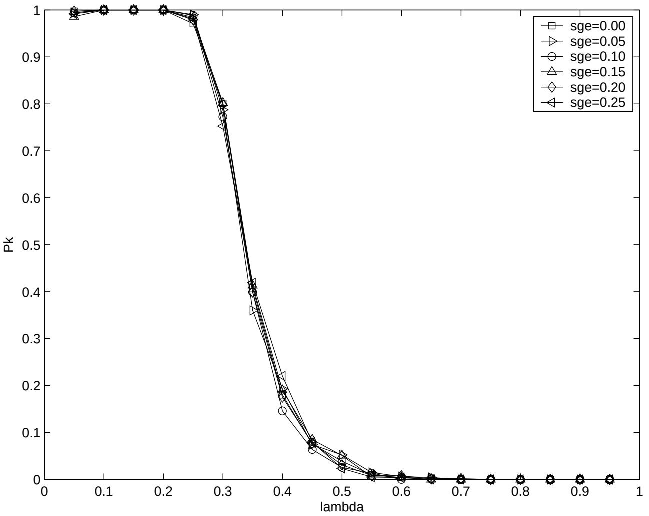

Fig. 13. Plot of P_k vs. λ for various values of σ_ε. The model HMM-1 was used with $K^{(0)}$=10.

We apply our algorithm with various values $\lambda \in \{0.05, 0.10, ..., 0.95\}$ and plot the results for model HMM-1 in Fig. 13. Similar results are obtained for HMM-2, HMM-3 and HMM-4. It can be seen here that the value of P_k varies from one to zero as we increase the value of λ, more and more of the signal variance is accounted by the random noise component of the HMM, and consequently the segment means become less variable, which in practice means that the number of segments identified goes down. This is illustrated by Fig. 14, which shows the average number of segment identified as a function of λ for different values of σ_ε. It can be seen that for λ=0.40, the average number of segment is around two, which explains why the value of P_k is relatively low, meaning that the algorithm performs reasonably well. It also means that a larger value of λ will make the algorithm more conservative, making it less likely to identify breaks in a time series when there are none, but at the cost of a higher probability P_k of misclassifying observations when there are breaks in a time series.

4.4 Experiment no. 4: Senegal river discharge

In the fourth experiment we use the time series of the Senegal river annual discharge data, measured at the Bakel station for

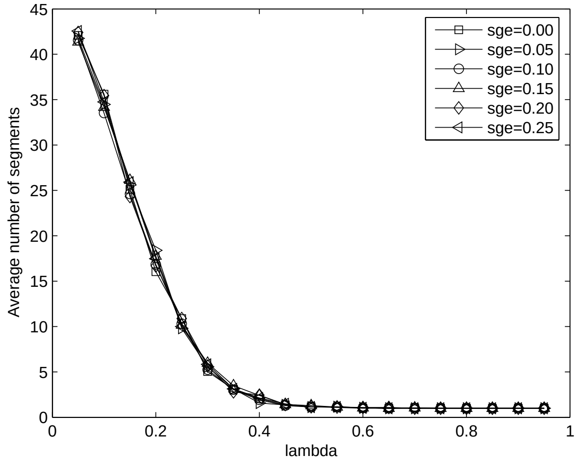

Fig. 14. Average number of segments identified vs. λ for various values of σ_ε. The model HMM-1 was used with $K^{(0)}$=10.

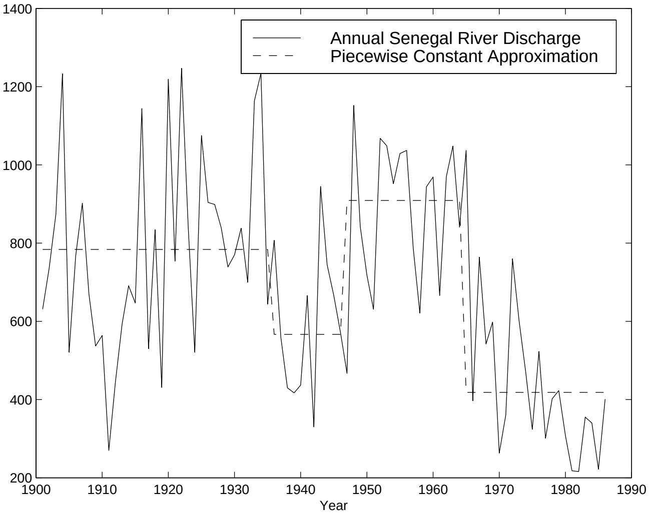

Fig. 15. Plot of the Senegal time series and a typical segmentation (obtained by HMM-1).

the years 1903–1988. This time series has been previously used in Fortin et al. (2004a), Hubert (1997, 2000), Kehagias (2004) and Kehagias et al. (2006); it consists of 86 points and its graph (together with a segmentation) appears in Fig. 15.

We applied the four variants of the segmentation algorithm to the Senegal data, using $K^{(0)}$=10 and several different values of $\lambda \in [0.35, 0.55]$. A typical segmentation (obtained by HMM-1 with λ=0.40) is plotted in Fig. 15.

Since the "true" segmentation of the Senegal river data is not known, we cannot compute P_k values. Instead, in Table 1 we list the segmentations obtained by the four variants of our algorithm (with λ=0.40). For comparison purposes we also list the segmentation obtained by both Hubert (1997) using a branch-and-bound algorithm and Kehagias (2004) using a left-to-right HMM. All algorithms give the same breaks, at years 1938 (or 1936), 1949 and 1967, but they do not give the 1921 break obtained in Hubert (1997) and Kehagias (2004).

Table 1. Segmentations of the Senegal river discharge time series.

HMM-1	1902		1938	1949	1967	1988
HMM-2	1902		1938	1949	1967	1988
HMM-3	1902		1938	1949	1967	1988
HMM-4	1902		1938	1949	1967	1988
Hubert	1902	1921	1936	1949	1967	1988

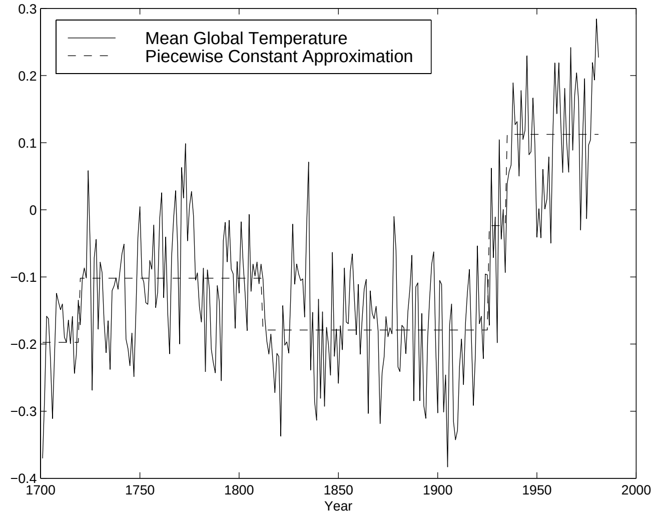

Fig. 16. Global environmental temperature: plot of the time series and a typical segmentation.

4.5 Experiment No. 5: Global environmental temperature

In this experiment we use the time series of annual mean global temperature for the period 1700–1980. The temperatures for the period 1902–1980 come from actual measurements while the remaining temperatures were *reconstructed* according to a procedure described in Mann et al. (1999) and also at the Internet address www.ngdc.noaa.gov/paleo/ei/ei_intro.html. The time series has length 281; a plot appears in Fig. 16.

The four variants of the segmentation algorithm give identical segmentations, as seen in Table 2; comparing these to the segmentation obtained by a left-to-right HMM in Kehagias (2004) we see that the only difference is in the years 1925–1934, namely the two breaks in these years are substituted by a single break in 1929 by the left-to-right HMM algorithm.

Table 2. Segmentations of the global environmental temperature time series.

HMM-1	1699	1719	1811	1925	1934	1980
HMM-2	1699	1719	1811	1925	1934	1980
HMM-3	1699	1719	1811	1925	1934	1980
HMM-4	1699	1719	1811	1925	1934	1980
Kehagias (2004)	1699	1719	1811	1929		1980

Table 3. Average execution time for one run of the segmentation algorithm.

	HMM-1	HMM-2	HMM-3	HMM-4
Experiment no. 1	0.44 s	0.95 s	0.39 s	0.94 s
Experiment no. 2	0.38 s	0.82 s	0.37 s	0.84 s
Experiment no. 3	0.18 s	0.65 s	0.17 s	0.63 s
Experiment no. 4	0.11 s	0.15 s	0.13 s	0.17 s
Experiment no. 5	0.42 s	0.55 s	0.44 s	0.53 s

4.6 About execution times

In Table 3 we give average execution time for a single segmentation using a MATLAB implementation of the algorithm (for HMM-i, i=1, 2, 3, 4) and for λ=0.40. It can be seen that the algorithm is fast, hence it is particularly easy to use as an exploration tool, running it with various parameter choices and inspecting the resulting segmentations in an interactive manner. For applications where computing time is critical, we recommend using the "pointwise"approximations of SMM's (HMM-1 and HMM-3), since they are always slightly faster and lead to segmentations that are as good as those obtained using the "interval-based"approximations of SMM's (HMM-2 and HMM-4), at least as measured by P_k.

5 Application to MCMC estimation

MCMC estimation of the parameters and change-points of SMM models can be very slow: parameter estimation for the SMM-2 model using the datasets presented in the previous sections can take hours instead of seconds using the Bayesian Gibbs sampling algorithm proposed by Fortin et al. (2004a). This computing time can potentially be reduced if reasonable values are provided to initialize the chain (both for the parameters and latent variables m_t). In this section we show that the pseudo-EM algorithm presented in this paper can help improve the performance of MCMC estimation for the SMM-2 model.

Fortin et al. (2004a) use a slightly different parameterization for the model SMM-2: they use $\sigma_x^2=\sigma_\mu^2+\sigma_\varepsilon^2$ and $\omega=\sigma_\mu^2/\sigma_x^2$ as model parameters, instead of σ_μ^2 and σ_ε^2. Denote by $\theta^{(0)}=(\eta^{(0)}, \omega^{(0)}, \mu^{(0)}, \sigma_x^{2(0)})$ the starting point of the chain and by $\tilde{\theta} = (\tilde{\eta}, \tilde{\omega}, \tilde{\mu}, \tilde{\sigma}_x^2)$ the exact (but typically unknown) values of the parameters.

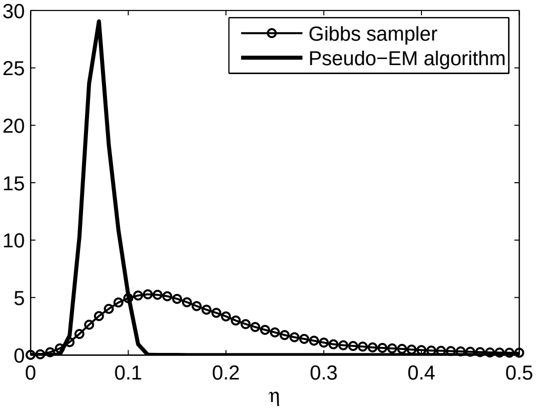

Fig. 17. Histogram of estimated values for the parameter η using the pseudo-EM algorithm and the Gibbs sampler.

In Fortin et al. (2004a), the parameters are initialized from the expectation of the prior distribution provided by the user. The prior distribution is defined by eight hyperparameters, two for each parameter of the model. If the user does not supply values for these hyperparameters, the algorithm automatically uses uniform distributions for the parameters η and ω. Flat priors are also used for the parameters μ and σ_x^2, but as the domain of definition of these parameters is not bounded, the prior expectation for μ and σ_x^2 must be reasonably close to the sample mean and variance. For this purpose, the prior for μ is centered on the mean of five observations chosen at random from the sample, and the prior for σ_x^2 is centered on the variance of these same five observations.

The latent variables m_t are initialized by sampling from the predictive distribution $p(m_t|m_{t-1}^{(0)}, y_t, \theta^{(0)})$. The performance of the Gibbs sampler is quite sensitive to the prior expectation of the transition probability η as well as to the initial values obtained for the latent variables m_t.

Given the sensitivity of the Gibbs sampler's performance to the initial value chosen for η, the first idea that comes to mind to hasten its convergence to a stationary distribution is to better choose the value of $\eta^{(0)}$. While the pseudo-EM algorithm has been shown to lead to accurate segmentations, this does not mean that the parameter estimates are reliable.

We simulated J=100 samples of size T=100 with the following population parameters: $\tilde{\eta}$=0.1, $\tilde{\omega}$=0.75, $\tilde{\mu}$=0, $\tilde{\sigma}_x^2$=1. We then estimated the parameter η by $\widehat{K}/T$ using λ set to 0.4, and $K^{(0)}$ set to 50.

We used the Gibbs sampler to obtain posterior distributions for these same simulated samples, using default values for the hyperparameters. For each sample, we performed R=2000 iterations, and kept only the last 1000. We hence obtained a sample of $J \cdot R/2$=100 000 values for each parameter.

Figure 17 compares the histograms of the parameter estimates obtained using the two estimation methods for parameter η, smoothed using the kernel density estimation function provided by the MATLAB statistics toolbox (`ksdensity`).

Recalling that the population value of η is 0.1, it is clear that the pseudo-EM algorithm underestimates η, and that the Bayesian method is much more reliable. Consequently, it will not be straightforward to initialize the η parameter using directly the output of the pseudo-EM algorithm.

At least for this case, the pseudo-EM parameter estimate for the transition probability is biased, but we know that the segmentation is accurate, at least in terms of the Beeferman's metric. This seems to be because changes of small amplitude are not picked up by the algorithm, hence do not show up as transitions, leading to an underestimation of η.

Even if the pseudo-EM algorithm leads to biased parameter estimates, it does a good job at segmenting the time series. Hence, we could use this information instead to initialize the latent variables in the algorithm. There is one caveat: the Gibbs sampler algorithm works in a similar way to the pseudo-EM algorithm for estimating η: at each iteration, it is reestimated from the ratio of the number of simulated transitions to the sample size. This means that the Gibbs sampler will start with a biased η value. For our simulation experiment, the value of η used to start the simulation will be generally too low, which can bring the chain into a trapping state from which it takes a lot of time to get out (possibly an infinite time!). To improve upon this strategy, a possibility is to add a small amount of noise at random points in the segmentation so as to increase the number of transitions to the a priori estimate of $\eta \cdot T$. In this way, the Gibbs sampler will still start both from a reasonable segmentation and an acceptable value for $\eta^{(0)}$.

Denote by $\eta^{(r,j)}$ the value of η simulated by the Gibbs sampler at the rth iteration and for the jth sample. Denote also by $m_t^{(r,j)}$ the value of m_t simulated by the Gibbs sampler at the rth iteration and for the jth sample. Denote by $\{\tilde{m_t}^{(j)}\}$ the exact values of the latent variables for the jth sample. To evaluate whether or not we are improving the performance of the Gibbs sampler by initializing it with a segmentation given by the pseudo-EM algorithm, we can look at the difference between $\eta^{(r,j)}$ and $\tilde{\eta}$, and between $m_t^{(r,j)}$ and $\tilde{m_t}^{(j)}$. We obviously need some goodness of fit measure to summarize this data.

MCMC techniques, and in particular Gibbs sampling, are used to derive posterior distributions for the parameters and latent variables of interest. One must evaluate when the Markov chain has converged to a stationary distribution, and iterate sufficiently after that to obtain a large enough sample to estimate reliably the posterior distribution. One way to compare the performance of two MCMC techniques consists of evaluating the sharpness and the accuracy of the probability distributions derived from the two chains on the second half of the iterations. After performing an even number of iterations R, we would like the cumulative posterior distribution of η approximated from the last $R/2$ iterations to be as close as possible to a Heaviside step function centered on $\tilde{\eta}$. Similarly, we would like the cumulative posterior distribution of $m_t - \tilde{m_t}^{(j)}$ to be as close as possible to a Heaviside function centered on zero. The squared distance between a cumulative distribution $F(x)$ and a Heaviside function $H(x-\tilde{x})$, centered on the target value $\tilde{x}$ and integrated over the domain of definition of the variable of interest, is a useful measure of both the sharpness and accuracy of the predictive distribution. It is known as the Continuous Ranked Probability Score, or $CRPS$ (Matheson and Winkler, 1976):

$$CRPS(F,\tilde{x}) = \int_{-\infty}^{\infty} (F(x) - H(x-\tilde{x}))^2 dx \qquad (21)$$

where $H(x-\tilde{x})$=0 if $x<\tilde{x}$ and $H(x-\tilde{x})$=1 if $x \geq \tilde{x}$. Note that the $CRPS$ is simply the absolute error between the target value and the prediction if the predictive distribution F is deterministic, i.e. if $F(x)$=$H(x-\widehat{x})$, where $\widehat{x}$ is the deterministic prediction. The $CRPS$ is a negative score, meaning that a smaller value means a better performance. Estimating the value of the $CRPS$ for an empirical cumulative distribution function is possible, but costly. We preferred to fit a parametric distribution to the samples. For the parameter η, we chose to fit a beta distribution by maximum likelihood, and then used adaptive Simpson quadrature to estimate the $CRPS$ numerically. For the latent variables, we chose to fit a normal distribution by the method of moments. When F is a normal distribution having a mean μ and a variance σ^2, the $CRPS$ is given by:

$$CRPS(F,\tilde{x}) = \sigma \cdot \left(\tilde{z}\,(2\Phi\,(\tilde{z}) - 1) + 2\varphi\,(\tilde{z}) - \pi^{-1/2}\right) \qquad (22)$$

where $\tilde{z}$= $(\tilde{x}-\mu)\,/\sigma$, Φ is the cumulative density function of the normal distribution and φ is its probability density function.

To obtain a single value of $CRPS$ for a given number of iterations R, we have pooled together all values of $\eta^{(r,j)}$ for r=$R/2+1..R$ and j=$1..J$ and all values of $m_t^{(r,j)} - \tilde{m_t}^{(j)}$ for t=$1..T$, r=$R/2+1..R$ and $j=1..J$. Figures 18 and 19 show that the $CRPS$ is lower both for η and for the latent variables when we use the pseudo-EM algorithm to initialize the Gibbs sampler.

As could be suspected, the improvement decreases as the number of iteration increases, down to about 5% for η and 8% for the latent variables when R reaches 2000. Still, the score reached after R=2000 iterations by the original Gibbs sampler is reached after only 600 iterations for η and after only 160 iterations for the latent variables when these latent variables are initialized using the pseudo-EM algorithm. While the time needed to perform the same number of iterations was only down slightly when initializing the Gibbs

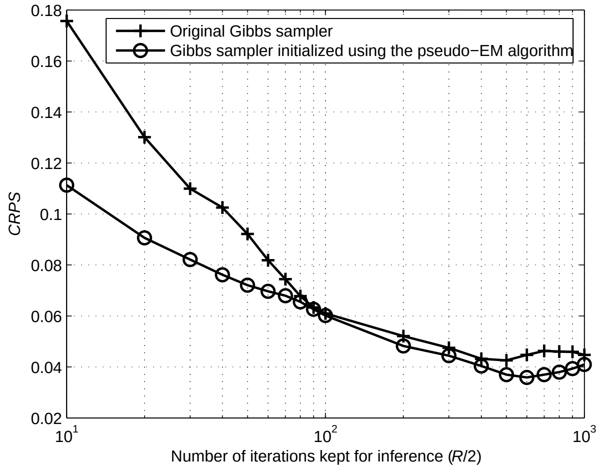

Fig. 18. $CRPS$ for the predictive distribution of η as a function of the number of iterations.

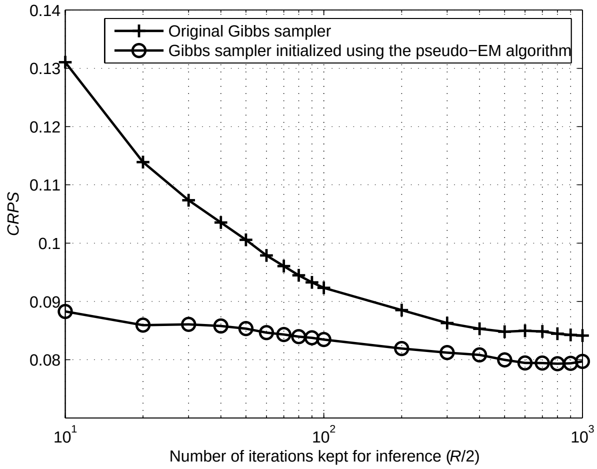

Fig. 19. $CRPS$ for the predictive distribution of $\{m_t\}$ as a function of the number of iterations.

sampler using the pseudo-EM algorithm, the $CRPS$ score obtained after the same number of iterations is improved, which means that the time needed to reach a given level of accuracy is reduced.

6 Conclusions

We have presented an HMM-based time series segmentation algorithm. We have applied (the four variants of) the algorithm to several segmentation tasks and obtained quite accurate segmentations, even in the case of noisy data; in addition the algorithm is quite fast (a couple of seconds for time series of a few hundred terms). An attractive feature of the algorithm is the automatic determination of segmentation order.

Several other computationally oriented segmentation algorithms have appeared in the literature, for example Hubert's branch-and-bound algorithm (henceforth called BB) (Hubert, 1997, 2000), Kehagias' left-to-right HMM (henceforth LR-HMM) (Kehagias, 2004) and Kehagias' DP algorithm (Kehagias et al., 2006). In addition, Fortin et al. (2004a) have proposed a more sophisticated MCMC segmentation procedure. A comparison leads to the following conclusions regarding the current algorithm.

1. It is at least as accurate and noise-robust as any of the above algorithms, somewhat faster than LR-HMM and DP and *much* faster than BB and MCMC.

2. It automatically determines the number of segments; this is also the case for MCMC, but not for BB, LR-HMM and DP.

3. It uses a smaller number of states than LR-HMM.

4. It is *not* guaranteed to find the optimal solution; this is also true of BB, LR-HMM and MCMC; the DP algorithm is guaranteed to find the globally optimal segmentation *for a given number of segments*.

The MCMC procedure proposed by Fortin et al. (2004a) differs from the other segmentation algorithms in that it attempts to find a more realistic model of the hydrological process (assuming the time series to be generated by a shifting means mechanism, it yields considerably more information regarding the segmentation and the time series parameters) and is oriented more towards model parameter estimation rather than towards segmentation. While our main interest is in the actual segmentation, an important motivation for the current algorithm was to use it as an initializer for this MCMC procedure. The experiments of Sect. 5 indicate that this is a viable approach.

Regarding future work, the rigorous study of the convergence properties of our algorithm merits further research, which will be reported in a future publication.

Appendix A The Beeferman metric P_k

Beeferman's segmentation metric P_k ($\mathbf{s}, \mathbf{t}$) measures the error of a proposed segmentation $\mathbf{s}$=$(0, s_1, ..., s_{K-1}, T)$ with respect to a "true" segmentation $\mathbf{t}$= $(0, t_1, ..., t_{L-1}, T)$. P_k first appeared in the *text* segmentation literature (Beeferman et al., 1999) but it can be applied to any segmentation problem. Here we will only give a short intuitive discussion of P_k. The interested reader can find more details in Beeferman et al. (1999).

When is $\mathbf{s}$ identical to $\mathbf{t}$? The following two conditions are necessary and sufficient.

1. Each pair of integers (i, j) which is in the same segment under $\mathbf{t}$ is also in the same segment under $\mathbf{s}$.

2. Each pair (i, j) in a different segment under $\mathbf{t}$ is also in a different segment under $\mathbf{s}$.

It follows that the difference between $\mathbf{s}$ and $\mathbf{t}$ (i.e. the *error* of $\mathbf{s}$ with respect to $\mathbf{t}$) is the amount by which the above criteria are violated. This can happen in two ways:

1. some pair (i, j) which is in the same segment under $\mathbf{t}$ is in a different segment under $\mathbf{s}$;

2. or some pair (i, j) which is in the same segment under $\mathbf{t}$ is in a different segment under $\mathbf{s}$.

We can formalize the above description as follows. Define a function $\delta_{\mathbf{s}}(i, j)$ to be 1 when i and j are in the same segment under $\mathbf{s}$ and 0 otherwise; define $\delta_{\mathbf{t}}(i, j)$ similarly. Then we are interested in the quantity

$$\sum_{i=1}^{T} \sum_{j=1}^{T} \left| \delta_{\mathbf{s}}(i, j) - \delta_{\mathbf{t}}(i, j) \right| . \tag{A1}$$

There are two problems with (A1) and they both have to do with the range of the summations. First, considering *all* possible pairs (i, j) is too time consuming. Second, it yields an *undiscriminating* criterion, because even a grossly wrong $\mathbf{s}$ will agree with $\mathbf{t}$ on many pairs (for example most pairs $(i, i+1)$ will be placed in the same segment and most (i, j) pairs which are very far apart will be placed in different segments). Beeferman et al. (1999) propose to consider only pairs which are k steps apart $(i, i+k+1)$, where k is *half the average segment length.* It has been empirically validated that this choice of k works well; Beeferman et al. (1999) also discuss some theoretical justification for this choice. Hence P_k is defined as follows

$$P_k(\mathbf{s}, \mathbf{t}) = \sum_{i=1}^{T-k-1} \left| \delta_{\mathbf{s}}(i, i+k+1) - \delta_{\mathbf{t}}(i, i+k+1) \right| \tag{A2}$$

and this is what we have used in Sect. 4.

Edited by: A. Provenzale
Reviewed by: one referee

References

Beeferman, D., Berger, A., and Lafferty, J.: Statistical models for text segmentation, Machine Learning, 34, 177–210, 1999.

Bengio, Y.: Markovian models for sequential data, Neural Comp. Surveys, 2, 129–162, 1998.

Buishand, T. A.: Some methods for testing the homogeneity of rainfall records, J. Hydrol., 58, 11–27, 1982.

Buishand, T. A.: Tests for detecting a shift in the mean of hydrological time series, J. Hydrol., 75, 51–69, 1984.

Chernoff, H. and Zacks, S.: Estimating the current mean of normal distribution which is subject to changes in time, Ann. Math. Stat., 35, 999–1018, 1964.

Cobb, G. W.: The problem of the Nile: Conditional solution to a changepoint problem, Biometrika, 65, 243–252, 1978.

Forney, G.: The Viterbi algorithm, Proceedings of the IEEE, 61, 268–278, 1973.

Fortin, V., Perreault, L., and Salas, J. D.: Retrospective analysis and forecasting of streamflows using a shifting level model, J. Hydrol., 296, 135–163, 2004.

Fortin, V., Perreault, L., and Salas, J. D.: Analyse rétrospective et prévision des débits en présence de changements de régime, 57th Annual meeting of the Canadian Water Resources Association, Montréal, June 16–18, 2004.

Hipel, A. I. and McLeod, K. W.: Time Series Modelling of Water Resources and Environmental Systems, Elsevier, 1994.

Hoppe, H. and Kiely, G.: Precipitation over Ireland – Observed changes since 1940, Phys. Chem. Earth (B), 24, 91–96, 1999.

Hubert, P.: Change points in meteorological analysis, in: Applications of Time Series Analysis in Astronomy and Meteorology, edited by: Rao, T. S., Priestley, M. B., and Lessi, O., Chapman and Hall, 1997.

Hubert, P.: The segmentation procedure as a tool for discrete modeling of hydrometeorological regimes, Stoch. Env. Res. and Risk Ass., 14, 297–304, 2000.

Kehagias, Ath.: A Hidden Markov Model Segmentation Procedure for Hydrological and Enviromental Time Series, Stoch. Env. Res. and Risk Ass., 14, 117–130, 2004.

Kehagias, Ath., Nidelkou, Ev., and Petridis, V.: A Dynamic Programming Segmentation Procedure for Hydrological and Enviromental Time Series, Stoch. Env. Res. and Risk Ass., 20, 77–94, 2006.

Kiely, G., Albertson, J. D., and Parlange, M. B.: Recent trends in diurnal variation of precipitation at Valentia on the west coast of Ireland, J. Hydrol., 207, 270–279, 1998.

Lee, A. S. F. and Heghinian, S. M.: A shift of the mean level in a sequence of independent normal random variables – a Bayesian approach, Technometrics, 19, 503–506, 1977.

Mann, M. E., Bradley, R. S., and Hughes, M. K.: Northern hemisphere temperatures during the past millennium: inferences, uncertainties, and limitations, Geophys. Res. Lett., 26, 759–762, 1999.

Matheson, J. E. and Winkler, R. L.: Scoring rules for continuous probability distributions, Manage. Sci., 22, 1087–1095, 1976.

Paturel, J. E., Servat, E., Kouame, B., Lubes, H., Ouedraogo, M., and Masson, J. M.: Climatic variability in humid Africa along the Gulf of Guinea Part II: An integrated regional approach, J. Hydrol., 191, 16–36, 1997.

Perreault, L., Hache, M., Slivitzky, M., and Bobee, B.: Detection of changes in precipitation and runoff over eastern Canada and U.S. using a Bayesian approach, Stoch. Env. Res. and Risk Ass., 13, 201–216, 1999.

Perreault, L., Parent, E., Bernier, J., Bobee, B., and Slivitzky, M.: Retrospective multivariate Bayesian change-point analysis: a simultaneous single change in the mean of several hydrological sequences, Stoch. Env. Res. and Risk Ass., 14, 243–261, 2000.

Perreault, L., Bernier, J., Bobee, B., and Parent, E.: Bayesian change-point analysis in hydrometeorological time series. Part 1. The normal model revisited, J. Hydrol., 235, 221–241, 2000.

Perreault, L., Bernier, J., Bobee, B., and Parent, E.: Bayesian change-point analysis in hydrometeorological time series. Part 2. Comparison of change-point models and forecasting, J. Hydrol., 235, 242–263, 2000.

Rabiner, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE, 77, 257–286, 1988.

Rao, A. R. and Tirtotjondro, W.: Investigation of changes in characteristics of hydrological time series by Bayesian methods, Stoch. Hydrol. and Hydraulics, 10, 295–317, 1996.

Salas, J. D. and Boes, D. C.: Shifting level modelling of hydrologic time series, Adv. Water Res., 3, 59–63, 1980.

Servat, E., Paturel, J. E., Lubes, H., Kouame, B., Ouedraogo, M., and Masson, J. M.: Climatic variability in humid Africa along the Gulf of Guinea. Part I: detailed analysis of the phenomenon in Cote d'Ivoire, J. Hydrol., 191, 1–15, 1997.

Yao, Y.-C.: Estimating the Number of Change-Points via Schwarz' Criterion, Stat. Prob. Let., 6, 181–189, 1988.