REVERSIBLE JUMP MARKOV CHAIN MONTE CARLO METHODS AND SEGMENTATION ALGORITHMS IN HIDDEN MARKOV MODELS

R. PAROLI and L. SPEZIA

Università Cattolica SC, Milano and Biomathematics & Statistics Scotland

Summary

We consider hidden Markov models with an unknown number of regimes for the segmentation of the pixel intensities of digital images that consist of a small set of colours. New reversible jump Markov chain Monte Carlo algorithms to estimate both the dimension and the unknown parameters of the model are introduced. Parameters are updated by random walk Metropolis–Hastings moves, without updating the sequence of the hidden Markov chain. The segmentation (i.e., the estimation of the hidden regimes) is a further aim and is performed by means of a number of competing algorithms. We apply our Bayesian inference and segmentation tools to digital images, which are linearized through the Peano–Hilbert scan, and perform experiments and comparisons on both synthetic images and a real brain magnetic resonance image.

Key words: clustering; digital images; label-switching; Peano–Hilbert scan; post-processing.

1. Introduction

Hidden Markov models (HMMs) are widely used tools for dealing with time series with incomplete data. Consider a sequence of unobserved variables $X_N = (X_1, \ldots, X_N)$ that can be analysed only through the realizations of an auxiliary observed process $Y_N = (Y_1, \ldots, Y_N)$, by assuming that the observed variables are conditionally independent given the unobserved ones and that their distributions depend only on the contemporary realizations of the latent, or hidden, process. HMMs arise when the hidden process is assumed apriori to be a finite-state Markov chain. Exhaustive studies on HMMs are available in the monographs by MacDonald & Zucchini (1997), Cappé, Moulines & Rydén (2005) and Frühwirth-Schnatter (2006).

Bayesian inference in HMMs with a fixed number of regimes was conducted by Robert, Celeux & Diebolt (1993) through Gibbs sampling, with single-site updating of any hidden state. Chib (1996) proposed the simultaneous generation of the sequence of the hidden states. In the case of zero-mean Gaussian distributions, Robert, Rydén & Titterington (2000) provided a reversible jump Markov chain Monte Carlo (RJMCMC) method to estimate both the unknown parameters and the number of hidden regimes. Green & Richardson (2002) formalized spatial HMMs for disease mapping and developed a suitable RJMCMC algorithm.

*Author to whom correspondence should be addressed.

1 Dipartimento di Scienze Statistiche, Università Cattolica SC, Via Necchi 9, 20123 Milano, Italy.
2 Biomathematics & Statistics Scotland, Macaulay Institute, Craigiebuckler, Aberdeen AB15 8QH, UK.

Acknowledgments. The authors are grateful to Laura Origgi for the images reproduced in Figure 1 and to Mark Brewer for useful comments on the first draft of the paper. The authors also want to thank the Associate Editor, an anonymous referee, and the Technical Editor for their helpful comments. The MRI scan was provided by Maria Grazia Bruzzone, Istituto Neurologico Carlo Besta, Milano. The latter stages of LS’s contribution were funded by the Scottish Government’s Rural and Environment Research and Analysis Directorate.
whereas Cappé, Robert & Rydén (2003) introduced continuous-time MCMC algorithms for HMMs.

HMMs with an unknown number of regimes are considered here. One of the aims of the paper is the implementation of efficient algorithms to estimate both the dimension and the unknown parameters of the model. The first algorithm we consider performs parameter estimation when the number of regimes is fixed. The main problem to tackle is label switching, resulting from the multimodality of the posterior density: when we have \( m \) regimes, we have \( m! \) ways to label them. So, if the prior distributions are invariant to the relabelling of the regimes and they do not set artificial identifiability constraints, the posterior density is defined on \( m! \) subspaces. Hence, when we sample from the unconstrained posterior density, \( m! \) labellings can occur during the MCMC iterations. Note that if label switching occurs, inference cannot be performed by taking the ergodic averages of the simulated values. This is the so-called label-switching problem. We tackle it using the post-processing algorithm of Marin, Mengersen & Robert (2005), by which one of the \( m! \) modal regions of the posterior density is selected \textit{ex post}, and then the proximity of the entries of the MCMC sample to this region manages the relabelling. Label switching also causes problems of convergence, because the MCMC algorithm can be trapped in one subspace and fail to explore the whole support of the posterior density. Our algorithm is encouraged to visit all \( m! \) subspaces by random permutation sampling (Frühwirth-Schnatter 2001): each draw is concluded by generating one of the \( m! \) ways of labelling the regimes, and then all parameters are permuted according to the random ordering.

Instead of exploiting the conjugacy of the prior distributions to implement Gibbs sampling, we prefer to update the parameters by random walk Metropolis–Hastings moves, because the Gibbs sampler is less able to traverse the posterior surface and to escape local modes.

All sweeps of our MCMC algorithm are characterized by the updating of the parameters without the updating of the sequence of the hidden Markov chain, which is never simulated, in order to eliminate unnecessary randomness from the procedure and to reduce the dimension of the parameter space. This device, called ‘no completion’ (Cappé et al. 2003), improves the precision of the simulation and accelerates the convergence of the algorithm.

The second MCMC algorithm introduced in this paper allows estimation of the number of hidden states, when it is a random variable. We consider a RJMCMC algorithm in which the posterior density is unconstrained and the parameters are updated by random walk Metropolis–Hastings moves with the absence of completion; the dimension of the model changes in split-and-merge and birth-and-death moves, and all subspaces are visited by means of random permutation sampling.

The segmentation of the hidden regimes is a further aim. The segmentation problem consists of estimating the unobserved realization \( x^N \) from the observed realization \( y^N \), given the transition matrix and the parameters of the conditional distributions. We perform segmentation by means of various competing algorithms, whose performances we compare through simulation experiments. The four segmenters we consider are: the Viterbi algorithm (VA, MacDonald & Zucchini 1997, pp. 64, 65), the iterated conditional mode (ICM, Besag 1986), forward filtering–backward sampling (FF-BSa, Carter & Kohn 1994; Frühwirth-Schnatter 1994), and forward filtering–backward smoothing (FF-BSm, Kim 1993). From our results, it emerges that the two algorithms used by the signal processing community, namely VA and
ICM, have the worst performance, and that FF-BSa and, above all, FF-BSm, which is popular only among econometricians, work much better.

We apply our Bayesian inference and segmentation tools to digital images. Complex stochastic image segmenters are based on hidden Markov random fields, which have the drawback of requiring the cumbersome computation of the normalizing constant of the Gibbs density. Computational advantages can be obtained by replacing the hidden Markov random field with a hidden Markov chain (e.g. Giordana & Pieczynski 1997) by transforming the two-dimensional (2D) Markov random field into a one-dimensional (1D) Markov chain through a Peano–Hilbert scan of the image. The segmented 2D image is then reconstructed from \( x^N \) using an inverse Peano–Hilbert scan. By segmenting a sequence of pixel intensities, we show that HMMs can cluster the observations efficiently, even when the number of groups is unknown and no classification threshold is fixed.

The plan of the paper is as follows. In Section 2, the set-up of Gaussian HMMs is described. In Section 3, we introduce two MCMC algorithms for parameter estimation, when the number of hidden states is fixed; and for model dimension estimation, when the number of regimes is variable. Section 4 reports some rival segmentation algorithms. The Peano–Hilbert scan for the linearization of an image is presented at Section 5. The MCMC algorithms and the segmentation tools are illustrated and compared through some simulation experiments and a real digital image in Sections 6 and 7, respectively.

2. Gaussian hidden Markov models

Let us consider a pair of stochastic processes \( \{X_t\} \) and \( \{Y_t\} \), taking values in \( S_X = \{1, \ldots, m\} \) and in \( \mathbb{R} \), respectively. Process \( \{X_t\} \) is a priori a discrete-time, first-order, homogeneous Markov chain on a finite space \( S_X \). The transition matrix is \( \gamma = [\gamma_{ij}] \), where \( \gamma_{ij} = \Pr(X_t = j \mid X_{t-1} = i) \), for any \( i, j \in S_X \) and for any \( t = 2, \ldots, N \), with \( 0 < \gamma_{ij} < 1 \). Process \( \{Y_t\} \), given \( \{X_t\} \), is an observed sequence of conditionally independent random variables, whose conditional distributions depend on \( \{X_t\} \) only through the contemporary \( X_t \) and are assumed to be Gaussian.

As a result of these assumptions, the stochastic process \( (\{X_t\}, \{Y_t\}) \) is a Gaussian hidden Markov model (GHMM) that can be represented as the ‘signal plus noise’ model \( Y_t = \mu_i + \sigma_i E_t \), where \( \{E_t\} \) is a standardized Gaussian white noise process with \( E_t \sim N(0, 1) \), so that \( (Y_t \mid X_t = i) \sim N(\mu_i, \sigma_i^2) \), for any \( i \in S_X \) and for any \( t = 1, \ldots, N \), where \( N(\cdot, \cdot) \) denotes the Gaussian, or Normal, distribution. The inferential procedures and the computational tools we introduce can also be applied, however, when a different conditional distribution is assumed.

Following Cappé et al. (2003), we can reparameterize \( \gamma = [\gamma_{ij}] \) by \( \omega = [\omega_{ij}] \), according to the equality \( \gamma_{ij} = \omega_{ij} / \sum_j \omega_{ij} \), with \( \omega_{ij} > 0 \), for any \( i, j \in S_X \), to facilitate the random-walk Metropolis–Hastings moves of the MCMC algorithm.

The vector \( (\mu, \sigma^2, \Omega, m) \) contains the unknown parameters of the GHMM to be estimated, where \( \mu \) is the vector of the \( m \) signals \( \mu_i \) and \( \sigma^2 \) is the vector of the \( m \) variances \( \sigma_i^2 \). We can elicit the following independent and relabelling-invariant prior distributions, where \( G(\cdot) \) denotes the gamma distribution and \( U_D(a, b) \) denotes the discrete uniform distribution over the integers \( \{a, a + 1, \ldots, b\} \).
(i) \( \mu_i \sim N(\theta, \tau^2) \), for any \( i = 1, \ldots, m \); that is, a non-informative prior density is defined by placing \( \theta \) equal to the mid-point of the range of the observations and \( \tau \) equal to the range.

(ii) \( \sigma_i^{-2} \sim G(0.5, 0.5) \), for any \( i = 1, \ldots, m \); that is, we have a priori a gamma variable with mean 1 and variance 2, leading to low variability within each state.

(iii) \( \omega_{ij} \sim G(m \times I(i = j) + 0.6 \times I(i \neq j), 1) \), for any \( i, j = 1, \ldots, m \), where \( I(A) \) is the indicator function, which takes the value 1 if \( A \) is true and the value 0 otherwise. Cappé et al. (2005, p. 487) chose the gamma \( G(\cdot, 1) \), for any \( \omega_{ij} \), because it gives a priori a Dirichlet distribution on the corresponding row of the transition matrix \( (\gamma_{11}, \ldots, \gamma_{mm}) \). A priori the probability of persistence is greater than the probability of transition. The probability of persistence is about 0.7 and it slowly decreases as the number of states increases (for \( m = 2, 3, 4, 5 \), \( E(\gamma_{ij}) = 0.77, 0.72, 0.70, 0.68 \), respectively, whereas when \( i \neq j \), \( E(\gamma_{ij}) = 0.23, 0.14, 0.10, 0.08 \), respectively).

(iv) \( m \sim U_D(1, m_{\text{max}}) \), where \( m_{\text{max}} \) is the maximum number of hidden regimes we admit a priori.

The sequence of the observations is denoted by \( y^N = (y_1, \ldots, y_N) \). So, the joint density of all variables included in the model is

\[
p(m, \mu, \sigma^{-2}, \Omega, y^N) = p(y^N | \mu, \sigma^{-2}, \Omega, m) p(\mu | m) p(\sigma^{-2} | m) p(\Omega | m) p(m),
\]

where

\[
p(y^N | \mu, \sigma^{-2}, \Omega, m) = \prod_{t=1}^{N} \sum_{i=1}^{m} p(y_t | \mu_i, \sigma_t^2) \xi_{t|t-1}(i),
\]

with, for any \( t = 1, \ldots, N \),

\[
p(y_t | \mu_i, \sigma_t^2) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp \left\{ -\frac{1}{2} \left( \frac{y_t - \mu_i}{\sigma_t} \right)^2 \right\},
\]

and \( \xi_{t|t-1}(i) = \Pr(X_t = i | y^{t-1}, \mu, \sigma^2, \Omega, m) \), with \( y^{t-1} = (y_1, \ldots, y_{t-1}) \). Filtered probabilities \( \xi_{t|t-1}(i) \) can be computed recursively by iterating the following equations, for any \( t = 1, \ldots, N \) and for any \( i = 1, \ldots, m \):

\[
\xi_{t-1|t-1}(i) \propto \xi_{t-1|t-2}(i) p(y_{t-1} | \mu_i, \sigma_t^2), \quad \xi_{t|t-1}(i) = \sum_{j=1}^{m} \xi_{t-1|t-1}(j) \gamma_{ji},
\]

with \( \xi_{t-1|t-1}(i) = \Pr(X_{t-1} = i | y^{t-1}, \mu, \sigma^2, \Omega, m) \) and \( \xi_{1|0}(i) = 1/m \) (Hamilton 1994, ch. 22).

3. Markov chain Monte Carlo algorithms

We introduce two MCMC algorithms to simulate from the posterior distribution. The first allows us to obtain the posterior estimates of the parameters when the number of regimes is fixed. The second allows computation of the number of regimes when this number is a random variable.

© 2010 Australian Statistical Publishing Association Inc.
3.1. Fixed number of regimes

First, we present an MCMC algorithm in which the parameters are updated by random walk Metropolis–Hastings moves with no completion when the number of hidden regimes is fixed. The posterior density is unconstrained and all \( m! \) subspaces are visited by the algorithm by means of random permutation sampling (Frühwirth-Schnatter 2001). The ergodic means of the parameters are computed after the MCMC sample has been post-processed according to Marin et al. (2005).

Our MCMC algorithm consists of many sweeps, each one consisting of two main steps, namely (a) parameter simulation and (b) parameter permutation.

(a) Generate \( \mu_i, \ln \sigma_i^{-2} \) and \( \ln \omega_{ij} \), for any \( i, j \in S_X \), by random walk proposals; that is, \( \mu_i = \mu_i^{(\text{old})} + U_M; \ln \sigma_i^{-2} = \ln(\sigma_i^{-2})^{(\text{old})} + U_{\Sigma}; \ln \omega_{ij} = \ln \omega_{ij}^{(\text{old})} + U_{\Omega} \), where \( U_B \sim N(0, \sigma_B^2) \), with \( B \in \{ M, \Sigma, \Omega \} \).

Accept the proposals \( \mu, \sigma^2 \) and \( \Omega \), respectively, if \( u_B \leq \min\{1, A_B\} \), where \( u_B \) is a random number generated from the uniform distribution \( U(0, 1) \), and the acceptance ratios \( A_B(B \in \{ M, \Sigma, \Omega \}) \) are:

\[
A_M = \frac{p(y^N | \mu, \sigma^2, \Omega) p(\mu)}{p(y^N | \mu^{(\text{old}), [\sigma^2]^{(\text{old})}, \Omega^{(\text{old})}) p(\mu^{(\text{old})})},
\]

\[
A_{\Sigma} = \frac{p(y^N | \mu, \sigma^2, \Omega) p(\sigma^{-2}) \prod_{i=1}^{m} \sigma_i^{-2}}{p(y^N | \mu^{(\text{old}), [\sigma^2]^{(\text{old})}, \Omega^{(\text{old})}) p(\sigma^{-2})^{(\text{old})} \prod_{i=1}^{m} \sigma_i^{-2}^{(\text{old})}},
\]

\[
A_{\Omega} = \frac{p(y^N | \mu, \sigma^2, \Omega) \prod_{i=1}^{m} \prod_{j=1}^{m} \omega_{ij}}{p(y^N | \mu^{(\text{old}), [\sigma^2]^{(\text{old})}, \Omega^{(\text{old})}) \prod_{i=1}^{m} \prod_{j=1}^{m} \omega_{ij}^{(\text{old})}},
\]

which are the ratios of the products of likelihoods, prior densities and Jacobians for the logarithmic transformations of the \( \sigma_i^{-2} \)'s and the \( \omega_{ij} \)'s (Green & Richardson 2002).

(b) Let \( H \) be the class of the \( m! \) permutations \( \eta_j \) of the labels (\( \eta_j \in H \), for any \( j = 1, \ldots, m! \)). Randomly select one of the equiprobable permutations \( \eta_j \) and permute the current values of the parameters to obtain \( \eta_j(\mu, \sigma^{-2}, \Omega) \). For any iteration \( k \) of the MCMC algorithm, set \( (\mu^{(k)}, [\sigma^{-2}]^{(k)}, \Omega^{(k)}) = \eta_j(\mu^{(k)}, [\sigma^{-2}]^{(k)}, \Omega^{(k)}). \)

At the end of the iterations, we can tackle label switching by post-processing the MCMC sample. Such an algorithm has been proposed by Marin et al. (2005) for finite mixture models, and provides the Bayesian estimates of the parameters by minimizing the posterior expectation of a suitable loss function. So we can compute the ergodic averages of the parameters unaffected by the label switching, given that one of the \( m! \) modal regions is selected, and then the proximity of the entries of the MCMC sample to this region manages the relabelling.

After the burn-in of \( M \) iterations, if a sample of size \( Q \) is simulated, the post-processing algorithm works as follows:

(i) compute the posterior mode \( (\mu^*, [\sigma^{-2}]^*, \Omega^*) \), such that

\[
(\mu^*, [\sigma^{-2}]^*, \Omega^*) = \arg \max_{k=M+1, \ldots, M+Q} p(\mu^{(k)}, [\sigma^{-2}]^{(k)}, \Omega^{(k)} | y^N);
\]
(ii) for any $k = M + 1, \ldots, M + Q$, compute $\eta^*$ such that

$$
\eta^* = \arg \min_{\eta_j \in H} \left\| \eta_j(\mu^{(k)}, [\sigma^{-2}]^{(k)}, \Omega^{(k)}) - (\mu^*, [\sigma^{-2}]^*, \Omega^*) \right\|
$$

and set $(\mu^{(k)}, [\sigma^{-2}]^{(k)}, \Omega^{(k)}) = \eta^*(\mu^{(k)}, [\sigma^{-2}]^{(k)}, \Omega^{(k)})$.

In step (ii), for any entry of the MCMC sample, we first compute the Euclidean norm between any permuted vector of parameters and the posterior mode. Then we select that particular reordered vector that is the nearest to the posterior mode. In so doing, the label-switching problem is circumvented without the need to select any artificial identifiability constraint. Finally, ergodic averages of the processed values are taken to estimate the unknown parameters.

### 3.2. Variable number of regimes

Now we can consider the RJMCMC algorithm to estimate the number of hidden regimes, which this number is a random variable, by changing the dimension of the model in split-and-merge and birth-and-death moves. Our RJMCMC algorithm is based on four main steps: (A) update the parameters as in (a); (B) split one regime of the HMM into two or merge two regimes into one; (C) give birth or death to a regime; (D) permute the parameters as in (b).

In step (B), the split is randomly chosen with probability $b_m = \frac{m(m - 1) + 0.5 \times I(2 \leq m < m_{\text{max}})}{m_{\text{max}}}$, whereas the merge move is randomly chosen with probability $d_m = 1 - b_m$. In the merge move, two adjacent regimes, for example $i_1$ and $i_2 = i_1 + 1$, are randomly selected and combined in regime $i^*$, reducing by 1 the number of hidden regimes. The corresponding parameters are combined as follows:

$$
\mu_{i^*} = (\mu_{i_1} + \mu_{i_2})/2, \quad \sigma_{i^*}^2 = \left(\sigma_{i_1}^2 \sigma_{i_2}^2\right)^{1/2},
$$
$$
\omega_{i^*, i} = \omega_{i_1, i_1} + \omega_{i_2, i_2}, \forall i \neq i^*, \quad \omega_{i^*, j} = \omega_{i_1, j} \omega_{i_2, j}^{1/2}, \forall j \neq i^*,
$$
$$
\omega_{j, i^*} = (\omega_{i_1, i_1} \omega_{i_2, i_2})^{1/2} + (\omega_{i_1, i_2} \omega_{i_2, i_1})^{1/2}.
$$

In the split, a regime $i^*$ is picked at random and split into the two adjacent regimes $i_1$ and $i_2$. The corresponding parameters are split as follows, respecting the five equalities in (1). First, we generate the following $2m + 3$ random values: $u_1$ from $N(0, 0.5); u_2$ from $G(1, 5); v_i$ from $U(0, 1)$, for any $i \neq i^*$; $w_j$ from $G(1, 5)$, for any $j \neq i^*$; $\rho$ from $U(0, 1); \tau_1$ and $\tau_2$ from $G(1, 5)$. Then we set:

$$
\mu_{i_1} = \mu_{i^*} - \sigma_{i^*} u_1 \quad \mu_{i_2} = \mu_{i^*} + \sigma_{i^*} u_1
$$
$$
\sigma_{i_1}^2 = \sigma_{i^*}^2 u_2 \quad \sigma_{i_2}^2 = \sigma_{i^*}^2 / u_2
$$
$$
\omega_{i_1, i_1} = \omega_{i^*, i} v_i \quad \omega_{i_2, i_2} = \omega_{i^*, i} (1 - v_i)
$$
$$
\omega_{i_1, i_2} = \omega_{i^*, j} w_j \quad \omega_{i_2, i_1} = \omega_{i^*, j} / \tau_1
$$
$$
\omega_{i_1, i_2} = \omega_{i^*, i} (1 - \rho) \tau_1 \quad \omega_{i_2, i_2} = \omega_{i^*, i} (1 - \rho) / \tau_2.
$$

The split is accepted with probability $\min\{1, A\}$, whereas the merge is accepted with probability $\min\{1, A^{-1}\}$. Let tilde mark the parameters in the model with $m + 1$ regimes, with
respect to those in the model with \( m \) regimes. The analytic expression for \( A \) is

\[
\frac{p(y^N | \bar{\mu}, \bar{\sigma}^2, \bar{\Omega})}{p(y^N | \mu, \sigma^2, \Omega)} \times \frac{p(m + 1)}{p(m)} \times \frac{p(\bar{\mu}) p(\bar{\sigma}^{-2}) p(\bar{\Omega})}{p(\mu) p(\sigma^{-2}) p(\Omega)} \times \frac{d_{m+1}/m}{b_m/m} \times \frac{m + 1}{p(\mu_1) p(\sigma_1^2) \prod_{i \neq i^*} p(\omega_{i^*, i}) \prod_{j \neq i^*} p(\omega_{i, j^*})} \times |J|,
\]

where \( p(m + 1)/p(m) \) cancels, \( b_m/m \) is the probability of splitting the particular regime \( i^* \), whereas \( d_{m+1}/m \) is the probability of merging one of the \( m \) pairs \( (i_1, i_2) \) of adjacent regimes, the term \((m + 1)!/m!\), in which the factorials arise from the exchangeability assumption on the regimes, and \(|J|\) is the absolute value of the Jacobian of the transformation from \((\omega_{i^*, i}, \omega_{i^*, j}, w_j, \omega_{i, j^*}, \rho, \tau_1, \tau_2, \mu_{i^*}, u_1, \sigma_{i^*}^2, u_2)\) to \((\tilde{\omega}_{i_1, i}, \tilde{\omega}_{i_2, i}, \tilde{\omega}_{i_1, j}, \tilde{\omega}_{i_2, j}, \tilde{\omega}_{i_1, i^*}, \tilde{\omega}_{i_2, i^*}, \tilde{\omega}_{i_1, i^*}, \tilde{\omega}_{i_2, i^*}, \tilde{\mu}_{i_1}, \tilde{\mu}_{i_2}, \tilde{\sigma}_{i^*}^2, \tilde{\sigma}_{i^*}^2)\).

Note that the Jacobian can be decomposed into the product of four subdeterminants, namely \( J_1 \) for the transformation from \((\omega_{i^*, i}, v_i)\) to \((\tilde{\omega}_{i_1, i}, \tilde{\omega}_{i_2, i})\), \( J_2 \) for the transformation from \((\omega_{i^*, j}, w_j)\) to \((\tilde{\omega}_{i_1, j}, \tilde{\omega}_{i_2, j})\), \( J_3 \) for the transformation from \((\omega_{i^*, i}, \rho, \tau_1, \tau_2)\) to \((\tilde{\omega}_{i_1, i}, \tilde{\omega}_{i_1, i^*}, \tilde{\omega}_{i_2, i}, \tilde{\omega}_{i_2, i^*})\), and \( J_4 \) for the transformation from \((\mu_{i^*}, u_1, \sigma_{i^*}^2, u_2)\) to \((\tilde{\mu}_{i_1}, \tilde{\mu}_{i_2}, \tilde{\sigma}_{i^*}^2, \tilde{\sigma}_{i^*}^2)\). Hence we have

\[
|J| = |J_1 \cdot J_2 \cdot J_3 \cdot J_4| = 2^{m+2} \times \frac{\sigma_{i^*}^3 \omega_{i^*, i}^3 \rho (1 - \rho)}{u_2 \tau_1 \tau_2} \times \prod_{i \neq i^*} \omega_{i, i^*} \times \prod_{j \neq i^*} \omega_{i^*, j}.
\]

In step (C), birth and death are chosen with probability \( b_m \) and \( d_m \), respectively. In a death move, a regime is selected at random and then suppressed along with the corresponding parameters. In a birth move, a new regime \( i^* \) is added to the previous \( m \) regimes, the new parameters are drawn from their respective prior distributions, and the position of the new state is generated at random. The birth move is accepted with probability \( \min\{1, A\} \), whereas the death move is accepted with probability \( \min\{1, A^{-1}\} \). The analytic expression for \( A \) is

\[
\frac{p(y^N | \bar{\mu}, \bar{\sigma}^2, \bar{\Omega})}{p(y^N | \mu, \sigma^2, \Omega)} \times \frac{p(m + 1)}{p(m)} \times \frac{p(\bar{\mu}) p(\bar{\sigma}^{-2}) p(\bar{\Omega})}{p(\mu) p(\sigma^{-2}) p(\Omega)} \times \frac{d_{m+1}/(m + 1)}{b_m/(m + 1)} \times \frac{m + 1}{p(\mu_{i^*}) p(\sigma_{i^*}^2) \prod_{i \neq i^*} p(\omega_{i^*, i}) \prod_{j \neq i^*} p(\omega_{i, j^*})} \times |J|,
\]

where \( p(m + 1)/p(m) \) cancels, the ratio of the products of the prior densities multiplied by the reciprocal of the product of the densities of the new-born parameters is equal to 1, \( b_m/(m + 1) \) is the probability of giving birth to a new regime in the particular position \( i^* \), and \( d_{m+1}/(m + 1) \) is the probability of killing a particular regime, the term \((m + 1)\) has the same meaning as in (2), and the Jacobian \( J \) is 1.

Even if we introduce two different algorithms, in practice we need to make only one run to estimate both the dimension and the parameters of the model. We compute the posterior mode by RJMCMC. Then we select the entries of the MCMC sample corresponding to the true dimension of the model, and finally we post-process the sample and compute the ergodic averages of the parameters. However, when the analysis is highly complex, it could be safer to perform two runs, one for the selection of the dimension of the model and the second
for computation of the estimates, as we do in Section 6, where we consider both synthetic and medical images.

4. Segmentation algorithms

By means of the post-processed MCMC sample \( \eta^*(\mu^{(k)}, [\sigma^{-2}]^{(k)}, \Omega^{(k)}) \), for any \( k = M + 1, \ldots, M + Q \), we can compute the ergodic averages

\[
\hat{\mu}_i = Q^{-1} \sum_{k=M+1}^{M+Q} \eta^*(\mu_i^{(k)}), \quad \hat{\sigma}_i^{-2} = Q^{-1} \sum_{k=M+1}^{M+Q} \left[ \eta^*([\sigma_i^{-2}]^{(k)}) \right]^{-1},
\]

\[
\hat{\gamma}_{ij} = Q^{-1} \sum_{k=M+1}^{M+Q} \left[ \eta^*(\omega_{ij}^{(k)}) \right] / \left( \sum_{j=1}^m \eta^*(\omega_{ij}^{(k)}) \right),
\]

for any \( i, j = 1, \ldots, m \). Once \( \hat{\mu}, \hat{\sigma}^2 \) and \( \hat{\gamma} \) are available, we can obtain the segmentation of the sequence of the hidden states by computing the estimate \( \hat{x}^N \) of the hidden Markov chain \( x^N \) by applying four rival algorithms: VA, ICM, FF-BSa and FF-BSm. Henceforth, the conditioning on \( \hat{\mu}, \hat{\sigma}^2, \hat{\gamma} \) is dropped from the notation.

4.1. Viterbi algorithm

The Viterbi algorithm (VA) is a recursive procedure, developed in the engineering literature on speech recognition, that allows computation of the maximum a posteriori \( x^N \) of \( x^N \); that is, \( \hat{x}^N = \arg \max_{x_1, \ldots, x_N} \Pr(X_1 = x_1, \ldots, X_N = x_N | y^N) \).

Starting from \( \lambda_{1i} = p(y_1, x_1 = i) \), for any \( i = 1, \ldots, m \), the joint densities

\( \lambda_{it} = \arg \max_{x_t} p(y_1, \ldots, y_t, x_1, \ldots, x_{t-1}, x_t = i) \)

are recursively computed forwards, for any \( t = 2, \ldots, N \) and for any \( i = 1, \ldots, m \), and then the hidden states \( \hat{x}_t \) are computed backwards, as follows: (i) compute \( \lambda_{1i} = p(y_1, x_1 = i) \), for any \( i = 1, \ldots, m \); (ii) for any \( t = 2, \ldots, N \), compute \( \psi_t = \max_{(\lambda_{t-1,i}, \hat{y}_{t-1})} \lambda_{t-1} = \psi_t p(y_t, x_t = i) \), for any \( i = 1, \ldots, m \); (iii) compute \( \hat{x}_N = \arg \max_i \lambda_{NI} \); (iv) for any \( t = N - 1, \ldots, 1 \), compute \( \hat{x}_t = \arg \max_i (\lambda_{ti}, \hat{y}_{t+1}) \).

4.2. Iterated conditional mode algorithm

The iterated conditional mode (ICM) algorithm was proposed by Besag (1986), in a spatial process context, to reconstruct a true but unknown image. When the ICM algorithm is applied to HMMs, it is an iterative procedure for computing any marginal posterior mode (MPM) \( \hat{x}_t \) of \( x_t \); that is, \( \hat{x}_t = \arg \max_{x_t} \Pr(X_t = i | y_t, x_{t-1}, x_{t+1}) \).

To make this computation feasible, we assume the circularity condition on the Markov chain; that is, a transition from \( x_N \) to \( x_1 \) is always possible. The ICM algorithm works as follows: (i) start the algorithm by computing \( x_t^{(0)} = \arg \max_i p(y_t | x_t = i) \) and set the iteration counter \( h \) at 1; (ii) compute

\[
x_t^{(h)} = \arg \max_i \left[ \sum_j \gamma_{x_t^{(h-1)}, y_t | x_t = i} p(y_t | x_t = i) \right]^{(h-1)} \left[ \sum_j \gamma_{x_t^{(h-1)}, y_t | x_t = j} p(y_t | x_t = j) \right]^{(h-1)}
\]
for any $t = 1, \ldots, N$; (iii) compute $\pi^{(h)} = \Pr(X_1 = x_1^{(h)}, \ldots, X_N = x_N^{(h)} | y^N)$; (iv) if $\pi^{(h)} - \pi^{(h-1)} \approx 0$, the procedure can be stopped, by setting $\hat{x}_N = [x_N^{(h)}]$; otherwise, set the iteration counter $h$ to $h + 1$ and go to (ii). Note that $\pi^{(h)}$ never decreases at any stage $h = 1, 2, \cdots$, and eventual convergence is assured.

4.3. Forward filtering–backward sampling algorithm

The forward filtering–backward sampling (FF-BSa) algorithm was independently suggested by Carter & Kohn (1994) and Frühwirth-Schnatter (1994) to generate by Gibbs sampling the state variables in state–space models. The basis of the FF-BSa algorithm is the following equality:

$$p(x_N | y^N) = p(x_N | y^N) \prod_{t=1}^{N-1} p(x_t | x_{t+1}, y^t).$$

We can obtain the MPM $\hat{x}_N$ of $x_N$ by maximizing backwards any probability $\Pr(X_t | x_{t+1}, y^N)$; that is, $\hat{x}_t = \arg \max_i \Pr(X_t = i | y^N, \hat{x}_{t+1})$. The algorithm is iterated backwards from the starting point $\{\Pr(X_N = 1 | y^N), \ldots, \Pr(X_N = m | y^N)\}$, which is obtained by iterating forwards the filtered probabilities $\xi_{t|t}(i)$ and $\xi_{t+1|t}(i)$, for any $t = 1, \ldots, N$ and any $i = 1, \ldots, m$, as described in Section 2.

4.4. Forward filtering–backward smoothing algorithm

Finally, the MPM $\hat{x}_N$ of $x_N$ can be composed from any maximizer $\hat{x}_t$ of the smoothed probabilities $\Pr(X_t | y^N)$; that is, $\hat{x}_t = \arg \max_i \Pr(X_t = i | y^N)$. The smoothed probabilities of any state, that is, the probabilities of any state, at any time, given all observations, are computed backwards using the forward filtering–backward smoothing (FF-BSm) algorithm, developed to estimate the smoothed probabilities of state variables in state–space models with regime switching:

$$\Pr(X_t = i | y^N) = \xi_{t|t}(i) \sum_{j=1}^{m} y_{ij} \frac{\Pr(X_{t+1} = j | y^N)}{\xi_{t+1|t}(j)}.$$ (3)

The forward-filtering part works as in FF-BSa, then it is iterated backwards, according to (3), from the starting point $\{\Pr(X_N = 1 | y^N), \ldots, \Pr(X_N = m | y^N)\}$, by always considering the whole sequence of observations $y^N$.

5. The Peano–Hilbert space-filling curve

In order to use HMMs as segmenters of digital images, we have to transform the 2D set of pixels into a 1D set through the Peano–Hilbert space-filling curve on the image. This scan recursively traverses each quadrant of the image entirely before moving to the next quadrant and, thus, increases the similarity of neighbouring pixels in the scan. The linearization of the image, even if it reduces the complexity of models and algorithms with respect to those that maintain the spatial structure, produces satisfying results, competitive with those obtained through hidden Markov random fields (Giordana & Pieczynski 1997).
Figure 1. Construction of the Peano–Hilbert scan to transform the two-dimensional set of pixels into a one-dimensional set.

The Peano–Hilbert curve is obtained by a recursive procedure, and the first four stages of its construction are presented in Figure 1, starting with a four-pixel image and, at each step, multiplying the number of pixels of the image by four. In any figure, the curve starts at the centre of the north-west square and arrives at the centre of the north-east square. The curve always joins the centres of two contiguous squares. Continuation of this sequence creates the Peano–Hilbert scan on a \( W \times W \) image, where \( W = 2^k \), \( k \in \mathbb{N}_+ \) and \( N = W^2 \). The recursive procedure is defined by starting from a base element, such as that reproduced in Figure 1(a), for \( k = 1 \). At any iteration \( (k = 2, 3, 4; \text{Fig. 1b–d}) \), the curve is constructed by repeating the previous element four times, reducing its dimensions by a fourth. The orientation of the squares in the south-west and the south-east quadrant does not change, whereas north-west and north-east squares are rotated \( 90^\circ \) to the left and to the right, respectively. Finally, the four new squares are joined by three new segments. By iterating this procedure infinitely, we can obtain a limit curve that fills the whole area of the unit square, without producing any intersection in the curve.

6. Simulation experiments

In this section, the analysis of three simulated examples will be presented, by considering synthetic images with two, three and four colours, where colour in this context is defined as different levels of grey. We consider square synthetic images of size \( 128 \times 128 \) pixels. Each pixel \((i, j)\), for any \( i, j = 1, \ldots, 128 \), has intensity \( I_{ij} \), which is an integer belonging to the interval \([0, 255]\), where 0 is black and 255 is white. By the Peano–Hilbert scan, the 2D image is transformed into a 1D image, mapping \( I_{ij} \) to \( I_t \), and then pixel intensities are standardized:

\[
y_t = \frac{[I_t - E(I_t)]}{\sqrt{V(I_t)}}, \text{ for any } t = 1, \ldots, N, \text{ where } N = 128^2.
\]

The number of hidden states was estimated by running RJMCMC for 100,000 iterations. Our inference on \( m \) was based on the visualization of both the traces and the posterior running means of the number of hidden states, computed on a large number of iterations without a burn-in phase, in order to improve the chain mixing through the space of model components. Then we iterated the fixed-dimension MCMC algorithm 10,000 times, collecting all draws, after a burn-in period of 10,000 sweeps, and post-processed the MCMC sample. After that, the estimates \( \hat{\mu} \), \( \hat{\sigma}^2 \) and \( \hat{\Phi} \) were computed and the hidden image \( \hat{x}^N \) was segmented \textit{ex post} by means of the four algorithms described in Section 4. Finally, a reverse Peano–Hilbert scan was performed to obtain the 2D segmented image.

To validate the performance of the HMM segmenter, we compared segmented images with the corresponding synthetic hidden images created for the various experiments.
Comparisons were made by computing the misclassification ratio (MR), which is the ratio of the number of misclassified pixels to the total number of pixels. The starting values of all algorithms were generated randomly from their respective prior distributions; RJMCMC algorithms started from $m = 1$. In all examples, visual inspection of monitoring statistics was carried out.

The results of our simulations comprise 48 images. They can be seen in a technical report that is available online (see Supporting Information on the journal website, Data S1). Here, we display only some of the images, obtained in the example with three colours.

### 6.1. Two-colour synthetic image

Consider the following two-colour true image: a black background on which is painted a large white circle with a smaller black circle inside. The synthetic images were produced as follows: intensities were created that equalled 60 for the black pixels and 150 for the white pixels. To these we added a Gaussian noise whose standard deviation can take any of the values 25, 50 or 80 for either state. The RJMCMC algorithm visits different subspaces: from the posterior probabilities of the visited models, a strong support for the correct model emerged. In particular, for $m = 2$ the posterior probabilities (see Table 1) were between 0.85 and 0.95 for all three images. After having obtained the most probable HMM, we ran the MCMC algorithm for $m = 2$ to estimate its parameters and reconstruct the hidden image, given the values of the signals, the variances and the transition matrix. The application of the four competing segmentation algorithms produced the MRs shown in Table 2. From investigations of both the quality of the segmented images (Data S1) and the MR values, it can be seen that the FF-BSm algorithm worked much better than its rivals and its performance was satisfactory even when the values of the standard deviations were high. The FF-BSa algorithm gave good results as well, whereas the VA gave a poor performance, with MRs of up to 27%.

### Table 1

**Posterior probabilities of the number of hidden states $m$ for the two-colour synthetic images, obtained by blurring the original images with a Gaussian noise of standard deviation $\sigma = 25, 50$ or 80**

<table>
<thead>
<tr>
<th>$m$</th>
<th>$\sigma = 25$</th>
<th>$\sigma = 50$</th>
<th>$\sigma = 80$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>2</td>
<td>0.95</td>
<td>0.91</td>
<td>0.85</td>
</tr>
<tr>
<td>3</td>
<td>0.03</td>
<td>0.04</td>
<td>0.07</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>$\geq 5$</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
</tbody>
</table>

### Table 2

**Misclassification ratios for the two-colour synthetic images, obtained by blurring the original images with a Gaussian noise of standard deviation $\sigma = 25, 50$ or 80**

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>FF-BSm</th>
<th>VA</th>
<th>ICM</th>
<th>FF-BSa</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>0.47%</td>
<td>0.78%</td>
<td>0.56%</td>
<td>0.53%</td>
</tr>
<tr>
<td>50</td>
<td>1.67%</td>
<td>7.45%</td>
<td>3.62%</td>
<td>2.63%</td>
</tr>
<tr>
<td>80</td>
<td>3.15%</td>
<td>27.20%</td>
<td>11.65%</td>
<td>7.12%</td>
</tr>
</tbody>
</table>
6.2. Three-colour synthetic image

The three-colour true image is presented in Figure 2(a): we painted white and black triangles as a background on which there are two circles – a grey circle with a black circle inside. The synthetic images were produced as follows: intensities were created that equalled 20 for the black pixels, 120 for the grey pixels, and 150 for the white pixels. To these we added a Gaussian noise whose standard deviation can take any of the values 25, 50 or 80 for all states. The RJMCMC algorithm visits different subspaces; again the values of the posterior probabilities of the visited models strongly supported the correct one. In particular, for \( m = 3 \), the posterior probabilities (see Table 3) were near 0.80, for all values of the standard deviations \( \sigma_j \). After having obtained the most probable HMM, we ran the MCMC

\[ \text{Table 3} \]

<table>
<thead>
<tr>
<th>( m )</th>
<th>( \sigma = 25 )</th>
<th>( \sigma = 50 )</th>
<th>( \sigma = 80 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>2</td>
<td>0.05</td>
<td>0.06</td>
<td>0.07</td>
</tr>
<tr>
<td>3</td>
<td>0.84</td>
<td>0.81</td>
<td>0.79</td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>( \geq 5 )</td>
<td>0.02</td>
<td>0.02</td>
<td>0.03</td>
</tr>
</tbody>
</table>
algorithm for \( m = 3 \). The segmented images \( \hat{x}_N \), when \( \sigma_1 = \sigma_2 = \sigma_3 = 80 \), are reproduced in Figure 2, in which the four competing segmentation algorithms are applied, and all MRs are given in Table 4. Also in this case, by investigating both the quality of the segmented images (not all reproduced here) and the MR values, we can see that the FF-BSm algorithm always worked much better than the others, giving a good performance even when the values of the standard deviations are high (its MRs are not greater than 5.2%). The FF-BSa algorithm gave good results, too, whereas the VA gave a poor performance, with MRs of up to 42%.

### 6.3. Four-colour synthetic image

To obtain the four-colour true image we took a black and white checked background on which we painted first a deep grey triangle and then a light grey circle. The synthetic images were produced as follows: intensities were created that equalled 20 and 220 for the background squares, and 120 and 80 for the triangle and the circle, respectively. To these we added a Gaussian noise whose standard deviation can take any of the values 25, 50 or 80 for all states. The RJMCMC algorithm visits different subspaces and, in this trial, the value of the posterior probability of the true model was lower than those of the two previous experiments. This was possible because sometimes the two levels of grey, of the circle and of the triangle, were identified as one colour only; in fact, the posterior probabilities of the models with \( m = 3 \) and \( m = 4 \) are very close (see Table 5). The MCMC algorithm for the estimates of the parameters and the reconstruction of the hidden image was run both for \( m = 3 \) and \( m = 4 \) but it gave approximately the same results. Application of the four competing segmentation algorithms produced the MRs shown in Table 6. The analysis of the segmented images (Data S1) revealed that the segmentation algorithms, even if they classified more pixels incorrectly than in the two previous examples, where the basic figures have simpler shapes, could reconstruct the geometric figures in the images correctly and segmented the background well. By contrast, the triangle and the circle tended to be segmented by the same colour, especially

### Table 4

<table>
<thead>
<tr>
<th></th>
<th>FF-BSm</th>
<th>VA</th>
<th>ICM</th>
<th>FF-BSa</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma = 25 )</td>
<td>1.56%</td>
<td>5.29%</td>
<td>4.91%</td>
<td>2.20%</td>
</tr>
<tr>
<td>( \sigma = 50 )</td>
<td>2.01%</td>
<td>38.35%</td>
<td>4.08%</td>
<td>2.40%</td>
</tr>
<tr>
<td>( \sigma = 80 )</td>
<td>5.16%</td>
<td>42.00%</td>
<td>23.68%</td>
<td>7.43%</td>
</tr>
</tbody>
</table>

### Table 5

<table>
<thead>
<tr>
<th></th>
<th>( \sigma = 25 )</th>
<th>( \sigma = 50 )</th>
<th>( \sigma = 80 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m = 1 )</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>( m = 2 )</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>( m = 3 )</td>
<td>0.37</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>( m = 4 )</td>
<td>0.51</td>
<td>0.46</td>
<td>0.43</td>
</tr>
<tr>
<td>( m \geq 5 )</td>
<td>0.11</td>
<td>0.12</td>
<td>0.14</td>
</tr>
</tbody>
</table>

© 2010 Australian Statistical Publishing Association Inc.
TABLE 6
Misclassification ratios for the four-colour synthetic images, obtained by blurring the original images with a Gaussian noise of standard deviation $\sigma = 25, 50$ or $80$

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>FF-BSm</th>
<th>VA</th>
<th>ICM</th>
<th>FF-BSa</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>12.06%</td>
<td>12.08%</td>
<td>12.72%</td>
<td>12.74%</td>
</tr>
<tr>
<td>50</td>
<td>11.92%</td>
<td>58.53%</td>
<td>13.94%</td>
<td>12.16%</td>
</tr>
<tr>
<td>80</td>
<td>15.36%</td>
<td>34.29%</td>
<td>24.05%</td>
<td>17.86%</td>
</tr>
</tbody>
</table>

when the standard deviations increased, and so their intensities were not always classified correctly. The comparison of the various MRs confirms that the FF-BSm algorithm produced the lowest values (not greater than 15.4%), whereas the VA produced the highest ones (up to 58.5%). The analysis of the results we have from both these synthetic images and those of the previous experiments makes it clear that the shapes of the figures are crucial in assessing the results of the segmentations.

7. Analysis of a real digital image

Image segmentation is an important task in computational neuroimaging and is concerned with the development of efficient and automated techniques for the interpretation of brain images. In particular, in recent years the problem of automating the segmentation of brain images using magnetic resonance imaging (MRI) has received much attention. MRI scans provide information on human soft-tissue anatomy, and analysis of them plays an important role in the diagnosis of various neural diseases. In this section we describe the analysis of a real 2D MRI scan (Fig. 3a) of $256 \times 160$ pixels. To be able to apply the Peano–Hilbert scan for the transformation of the 2D image into a 1D image, we must have an image of size $2^k \times 2^k$, so we analysed two sub-images, of size $64 \times 64$ and $128 \times 128$ (Fig. 3b–c, respectively), and a $256 \times 256$ image, obtained by adding background pixels to the original MRI scan (Fig. 3d). On these images we first ran RJMCMC for 100 000 iterations and then iterated the fixed-dimension MCMC algorithm 10 000 times, collecting all draws, after a burn-in period of 10 000 sweeps, and post-processing the MCMC sample to obtain the parameter estimates. After that, the segmentation was performed using the FF-BSm algorithm, as this is the algorithm that performed best, as shown in the results of the simulations in Section 6. The RJMCMC algorithms indicated that the number of hidden states for the three images is $m = 6$. The segmented images are represented in Figure 4. The comparisons of our MRI scan, for all three sizes (Fig. 3), with the corresponding segmented images we obtained (Fig. 4) were
satisfactory, because the classifications of the pixel intensities in the new images were very powerful. All parts of the brain were reproduced consistently and their borders can be seen easily, because of the chromatic contrasts of the different tissues. For example, in the centre of Figures 3(b)–(d) we can see some tissues looking like a butterfly with a light grey body and deep grey wings shading into lighter tones. All these features are reproduced in Figure 4, where the darkest tissue (i.e. the lateral ventricles) is always well defined. Furthermore, if we examine the insular cortex, that is, the external part of the brain near the skull, which looks like an indented coastline with gulf and peninsulas, we note that these features are mapped well onto the segmented images and that their volumes are clearly identifiable.

8. Conclusions

We applied hidden Markov models to the segmentation of the pixel intensities of a brain magnetic resonance image in a small set of colours, whose cardinality was unknown, and which has been estimated by new reversible jump Markov chain Monte Carlo algorithms.

Complex stochastic image segmenters are based on hidden Markov random fields, which have the drawback of the cumbersome computation of the normalizing constant of the Gibbs density. By contrast, our image segmenters have computational advantages because we replaced the hidden Markov random field with a hidden Markov chain, by transforming the 2D Markov random field into a 1D Markov chain through a Peano–Hilbert scan of the image.

We also carried out experiments on synthetic images and performed segmentation by means of four competing algorithms, whose performance we compared through simulation experiments. We stretched the limits of our experiments by blurring the images through noises with standard deviations of 25, 50 and 80, which are much greater than those we encountered in our real images, in which the standard deviations of the pixel intensities are between 25 and 37.

References


© 2010 Australian Statistical Publishing Association Inc.

Supporting information

Additional Supporting Information may be found in the online version of this article at http://www.interscience.wiley.com/jpages/1369-1473:

Data S1. Supplementary material to Section 6.

Please note: Wiley-Blackwell are not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.

© 2010 Australian Statistical Publishing Association Inc.