On-line inference for hidden Markov models via particle filters

Well-log data

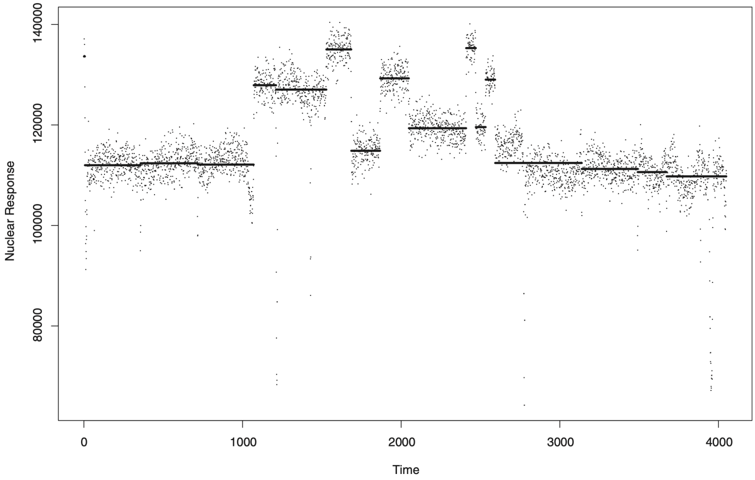

Figure 1: The problem of detecting break points in well-log data

Problem description

- Measurements of the nuclear magnetic response of underground rocks.

Problem description

- Measurements of the nuclear magnetic response of underground rocks.
- The underlying signal is piecewise constant, each constant segment relating to a stratum of a single type of rock. The jump discontinuities in the signal occur each time that a new rock stratum is met.

Problem description

- Measurements of the nuclear magnetic response of underground rocks.
- The underlying signal is piecewise constant, each constant segment relating to a stratum of a single type of rock. The jump discontinuities in the signal occur each time that a new rock stratum is met.
- In drilling for oil, it is important to predict changes in rock type to prevent blow-outs.

Problem description

- Measurements of the nuclear magnetic response of underground rocks.
- The underlying signal is piecewise constant, each constant segment relating to a stratum of a single type of rock. The jump discontinuities in the signal occur each time that a new rock stratum is met.
- In drilling for oil, it is important to predict changes in rock type to prevent blow-outs.
- They can be avoided by adjusting the pressure in the borehole each time that a new type of rock is met.

Problem description

- Measurements of the nuclear magnetic response of underground rocks.
- The underlying signal is piecewise constant, each constant segment relating to a stratum of a single type of rock. The jump discontinuities in the signal occur each time that a new rock stratum is met.
- In drilling for oil, it is important to predict changes in rock type to prevent blow-outs.
- They can be avoided by adjusting the pressure in the borehole each time that a new type of rock is met.
- To detect changes in rock strata as drilling proceeds, data need to be collected from around the drill head.

HMM

Three component HMM.

- I_t: hidden state variable.
- X_t: measurable state(?)
- Y_t: measurements.

Hidden states

$I_t \in \{1, ..., R\}$: follow a Markov chain:

$$P(I_t = l | I_{t-1} = i) = P_{il}.$$

Measurement process

$$X_t = f_i(X_{t-1}, V_t)$$

$$Y_t = g_i(X_t, W_t)$$

V_t, W_t represent random noise.

HMM for well-log data

- $I_t = (S_t, O_t)$, $S_t, O_t \in \{1, 2\}$.

4 possible states:

- $I_t = (1, 1)$
- $I_t = (1, 2)$
- $I_t = (2, 1)$
- $I_t = (2, 2)$

HMM for well-log data

- $X_t = X_{t-1}$ if $S_t = 1$. Models piece-wise const function.
- $X_t = \mu + \sigma V_t$ if $S_t = 2$.

The model for X_t allows for jumps centered around μ.

HMM for well-log data

- $Y_t = X_t + \tau_1 W_t$ if $O_t = 1$
- $Y_t = \nu + \tau_2 W_t$ if $O_t = 2$.

V_t, W_t are uncorrelated, standard Gaussian RVs.

The model for Y_t allows for clusters of outliers around ν.

Particle filters for discrete space

Since I_t takes on a fininte support ($R = 4$ possible values), for each of the N particles, we can enumerate RN new particles rather than taking a random sample.

Particle filters for discrete space

Since I_t takes on a fininte support ($R = 4$ possible values), for each of the N particles, we can enumerate RN new particles rather than taking a random sample.

The standard SMC performs multinomial resampling to prune the population down to N; but this is redundant: there is no reason to keep multiple copies of the same particle.

Particle filters for discrete space

Since I_t takes on a fininte support ($R = 4$ possible values), for each of the N particles, we can enumerate RN new particles rather than taking a random sample.

The standard SMC performs multinomial resampling to prune the population down to N; but this is redundant: there is no reason to keep multiple copies of the same particle.

If we are limited by budget of N particles, how can we prune the population from RN down to N?

Particle filters for well-log data

Particle $\beta_t = \{\alpha_t, \gamma_t^2, i_t, \tau_t\}$:

- α_t, γ_t^2: mean and variance of the posterior distribution over X_t.
- i_t: the hidden states.
- τ_t: the time of last change point in X_t.

Particle filters for well-log data

At iteration t of SMC:

- We have N particle-weight pairs $\{(\beta_{t-1,j}, q_{t-1,j})\}$.

Particle filters for well-log data

At iteration t of SMC:

- We have N particle-weight pairs $\{(\beta_{t-1,j}, q_{t-1,j})\}$.
- Enumerate over all R possible states for $j = 1, ..., N$. This yields RN particles at time t: $\{\beta_{t,j,k}\}, k = 1, ..., R$.

Particle filters for well-log data

At iteration t of SMC:

- We have N particle-weight pairs $\{(\beta_{t-1,j}, q_{t-1,j})\}$.
- Enumerate over all R possible states for $j = 1, ..., N$. This yields RN particles at time t: $\{\beta_{t,j,k}\}, k = 1, ..., R$.
- Calculate new weights $q_{t,j,k} = q_{t-1,j} P_{il} L(y_t|\beta_{t,j,k})$.

Particle filters for well-log data

At iteration t of SMC:

- We have N particle-weight pairs $\{(\beta_{t-1,j}, q_{t-1,j})\}$.
- Enumerate over all R possible states for $j = 1, ..., N$. This yields RN particles at time t: $\{\beta_{t,j,k}\}, k = 1, ..., R$.
- Calculate new weights $q_{t,j,k} = q_{t-1,j} P_{il} L(y_t | \beta_{t,j,k})$.
- Resample from a population of RN particles down to N particles to get $(\beta_{t,j}, q_{t,j})$ $j = 1, ..., N$.

Particle filters for well-log data

At iteration t of SMC:

- We have N particle-weight pairs $\{(\beta_{t-1,j}, q_{t-1,j})\}$.
- Enumerate over all R possible states for $j = 1, ..., N$. This yields RN particles at time t: $\{\beta_{t,j,k}\}, k = 1, ..., R$.
- Calculate new weights $q_{t,j,k} = q_{t-1,j} P_{il} L(y_t | \beta_{t,j,k})$.
- Resample from a population of RN particles down to N particles to get $(\beta_{t,j}, q_{t,j})$ $j = 1, ..., N$.

What is the optimal way to select N out of RN particles?

Optimal resampling

- We have $q = \{q_j\}_{j=1}^M$: discrete probability mass function ($M = RN$).
- We want to find $Q = \{Q_j\}_{j=1}^M$, a stochastic approximation of q such that

1. $\mathbb{E}(Q_j) = q_j$.
2. the support of Q has no more than $N < M$ points.
3. $\mathbb{E}[\sum_j (Q_j - q_j)^2]$ is minimized.

Theorem 1

Theorem 1. Let $N < M$ and let c be the unique root of $\Sigma_{j=1}^{M} \min(cq_j, 1) = N$. If an approximation Q can be found, satisfying the conditions

(a) $E(Q_j) = q_j, j = 1, \ldots, M$, and
(b) the support of Q has no more than N points, with marginal distributions given by

$$Q_j = \begin{cases} q_j/p_j, & \text{with probability } p_j, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $p_j = \min(cq_j, 1)$, for $j = 1, \ldots, M$, then Q minimizes the expected squared error loss, $E\,\Sigma_{j=1}^{M} (Q_j - q_j)^2$, under the conditions stated.

Proof of Theorem 1

Appendix A: Proof of theorem 1

The condition that, with probability 1, at most N of the Q_js are non-zero implies that marginally Q_j is equal to some random variable C_j with probability p_j and is 0 otherwise. Further, it implies that $\Sigma_{j=1}^{M} p_j \leqslant N$, and since

$$E\{(Q_j - q_j)^2\} = p_j\{E(C_j) - q_j\}^2 + p_j \operatorname{var}(C_j) + (1 - p_j)q_j^2$$

the expected squared error loss is minimized by the choice $C_j = c_j$, a constant, for each j. Further, the unbiasedness condition $E(Q_j) = q_j$ implies that $c_j = q_j/p_j$.

With this choice of Q, we have

$$E\left\{\sum_{j=1}^{M}(Q_j - q_j)^2\right\} = \sum_{j=1}^{M} q_j^2(1/p_j - 1).$$

Minimizing this is equivalent to minimizing $\Sigma_{j=1}^{M} q_j^2/p_j$, subject to $\Sigma_{j=1}^{M} p_j \leqslant N$. This is achieved by expression (3).

Proof of Theorem 1: Objective function

Our goal is to minimize

$$\mathbb{E}\left[\sum_{j=1}^{M}(Q_j - q_j)^2\right]$$

such that

- $\mathbb{E}[Q_j] = q_j$
- At most N of Q_j are non-zero (selected).

Proof of Theorem 1: Define Q_j

At most N of Q_j are non-zero, so that means we need to select N of them with some probability.

$$Q_j = \begin{cases} C_j & \text{with probability } p_j \\ 0 & \text{otherwise,} \end{cases}$$

for some random variable C_j to be determined.

Proof of Theorem 1: Define Q_j

We have the condition that at most N Q_j's are not 0:

$$\sum_{j}^{M} 1[Q_j \neq 0] \leq N.$$

Now, the expectation of the indicator variable $1[Q_j \neq 0]$ is:

$$\mathbb{E}[1[Q_j \neq 0]] = p_j.$$

Sum over $j = 1, ..., M$, yields the expected number of particles selected:

$$\mathbb{E}[\sum_{j=1}^{M} 1[Q_j \neq 0]] = \sum_{j=1}^{M} p_j \leq N.$$

Proof of Theorem 1: why C_j should be deterministic

By tower property,

$$\mathbb{E}[(Q_j - q_j)^2] = p_j \mathbb{E}[(C_j - q_j)^2] + (1 - p_j)(0 - q_j)^2.$$

Let $C_j = \mu_j + \epsilon_j$, with ϵ_j is random variance with mean 0 and some variance σ^2.

Then, bias-variance decomposition yields:

$$\mathbb{E}[(C_j - q_j)^2] = (\mathbb{E}[C_j] - q_j)^2 + \mathbb{E}(\epsilon_j^2).$$

The expected squared loss can only be increased by variance of C_j.

Note: $var(C_j) = var(\epsilon_j^2) = \mathbb{E}(\epsilon_j^2)$.

Proof of Theorem 1: why C_j should be deterministic

Let's suppose C_j is deterministic. Then,

$$\mathbb{E}[(Q_j - q_j)^2] = p_j(C_j - q_j)^2 + (1 - p_j)q_j^2.$$

If C_j is not deterministic,

So,

$$\mathbb{E}[(Q_j - q_j)^2] = p_j((\mu_j - q_j)^2 + var(\epsilon_j)) + (1 - p_j)q_j^2.$$

So the sqaured error is minimized by assuming C_j is deterministic.

Proof of Theorem 1: unbiasedness yields C_j

We want unbiasedness: $\mathbb{E}[Q_j] = p_j C_j = q_j$.

This implies that $C_j = q_j / p_j$.

So we got our C_j.

Proof of Theorem 1: objective function

Plug C_j back in the squared error:

$$\mathbb{E}[(Q_j - q_j)^2] = p_j(q_j/p_j - q_j)^2 + (1 - p_j)q_j^2 \tag{1}$$

$$= q_j^2/p_j - q_j^2. \tag{2}$$

Proof of Theorem 1: objective function

Finally, our objective function is given by:

$$\sum_{j=1}^{M} \mathbb{E}[(Q_j - q_j)^2] = \sum_{j=1}^{M} (q_j^2/p_j - q_j^2). \tag{3}$$

Since q_j^2 is fixed, we need to optimize p_j with constraints:

- $0 \leq p_j \leq 1$,
- $\sum_{j=1}^{M} p_j \leq N$.

Proof of Theorem 1: "water-filling" argument

We want:

$$\text{argmin}_{p_1,\ldots,p_M} \sum_{j=1}^{M} q_j^2/p_j. \tag{4}$$

Since $p_j \leq 1$, for q_j large, we want to keep the term $q_j^2/p_j = q_j^2$, i.e., $p_j = 1$.

But of course, the condition that at most N Q_j is non-zero stops us from setting all $p_j = 1$.

We prioritize the ones with large q_j (weights).

Proof of Theorem 1: "water-filling" argument

The number of particles who will be selected with probability $p_j = 1$ is determined by solving for $c > 0$:

$$\sum_{j=1}^{M} \min(cq_j, 1) \leq N.$$

- Those particles whose weights are such that $cq_j \geq 1$, get $p_j = 1$ (selected).
- Otherwise, $p_j = cq_j < 1$ and they have chance to be selected according to p_j.

Proof of Theorem 1: check unbiasedness

For particles whose $p_j = 1$, $\mathbb{E}[Q_j] = \mathbb{E}[q_j/p_j] = q_j$.

For particles whose $p_j < 1$, $\mathbb{E}[Q_j] = p_j C_j + (1 - p_j)0 = q_j$ since $C_j = q_j/p_j$.

Algorithm

1. Normalize the weights $q^*_{t,j} = q_{t,j} / \sum_{j=1}^{M} q_{t,j}$. Solve for $c > 0$: $N = \sum_j \min(c q^*_{t,j}, 1)$.

Algorithm

1. Normalize the weights $q^*_{t,j} = q_{t,j} / \sum_{j=1}^{M} q_{t,j}$. Solve for $c > 0$: $N = \sum_j \min(cq^*_{t,j}, 1)$.
2. If $q^*_{t,j} \geq 1/c$, then place particle j in set 1 (selected). Otherwise, place particle in set 2. Let L be the size of set 1.

Algorithm

1. Normalize the weights $q^*_{t,j} = q_{t,j} / \sum_{j=1}^{M} q_{t,j}$. Solve for $c > 0$: $N = \sum_j \min(c q^*_{t,j}, 1)$.
2. If $q^*_{t,j} \geq 1/c$, then place particle j in set 1 (selected). Otherwise, place particle in set 2. Let L be the size of set 1.
3. Use stratified sampling to resample $N - L$ particles from set 2, proportional to the weight $q^*_{t,j}$.

Algorithm

1. Normalize the weights $q^*_{t,j} = q_{t,j} / \sum_{j=1}^{M} q_{t,j}$. Solve for $c > 0$: $N = \sum_j \min(cq^*_{t,j}, 1)$.
2. If $q^*_{t,j} \geq 1/c$, then place particle j in set 1 (selected). Otherwise, place particle in set 2. Let L be the size of set 1.
3. Use stratified sampling to resample $N - L$ particles from set 2, proportional to the weight $q^*_{t,j}$.
4. The L particles in set 1 maintain its weight $q^*_{t,j}$. The remaining $N - L$ particles receive weight $1/c$.

Results

- See Figure 2.

Results

Table 1. Comparison of performance of three particle filter algorithms (see the text for details) at analysing the well-log data†

Algorithm	*Absolute error*	*Square error*
New particle filter	3.66	0.49
Mixture Kalman filter	73.6	19.0
Basic particle filter	143	88.7

Results

Table 2. Comparison of performance of three particle filter algorithms (see the text for details) at analysing the well-log data†

Algorithm	*False positive results*	*Missed changepoints*
New particle filter	0	0
Mixture Kalman filter	4.5	1.1
Basic particle filter	25.5	0.03