Parameter estimation for Hidden Markov Models

Likelihood inference

Likelihood estimation methods with particle filtering

- Some algorithms
  - Likelihood methods
    - optimisation based
    - gradient based
    - expectation maximisation
  - offline or online
    - we will focus on offline methods
    - only sketch on-line ones to give very basic idea

Maximum Likelihood based methods

- Off-line case: Estimate of θ^* as the maximizing argument of the marginal likelihood of the observed data:

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} \ \ell_T(\theta) \tag{1}$$

where

$$\ell_T(\theta) = \log \ p_\theta(y_{0:T}) \, . \tag{2}$$

- Online case:
  - use a recursive method
  - let θ_n be the estimate of the model parameter after $n-1$ observations
  - update the estimate to θ_{n+1} after receiving the new data y_n.

Offline Maximum Likelihood based methods

Off-line case:

- Estimate of θ^* as:

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} \; \hat{\ell}_T(\theta) \tag{3}$$

  where

$$\hat{\ell}_T(\theta) = \log \widehat{p_\theta(y_{0:T})}.$$

- Can use direct optimisation
  - grid on θ, BFGS, or other popular optimisation methods
- is difficult due to variance of $\hat{p}_\theta(y_{0:T})$

On the Monte Carlo variance of $p_\theta(y_{0:T})$

- Recall, SMC results in unbiased estimation of the marginal likelihood
$$\mathbb{E}_N[\hat{p}_\theta(y_{0:T})] = p_\theta(y_{0:T})$$
- Loosely speaking
$$\hat{p}_\theta(y_{0:T}) = p_\theta(y_{0:T}) + \mathcal{V}$$
with $\mathcal{V}$ some non-trivial zero mean noise depending on T, N and model.
  - recall $\widehat{p}_\theta(y_{0:n})$ has a relative (non-asymptotic) variance that increases linearly with n
- The monte carlo variability is quite an issue for finding maximum over θ

Approximating $\log p_\theta(y_{0:T})$

- Note that
$$\mathbb{E}_N[\hat{p}_\theta(y_{0:T})] = p_\theta(y_{0:T})$$
implies that
$$\mathbb{E}_N[\log \hat{p}_\theta(y_{0:T})] \neq \log p_\theta(y_{0:T})$$
- So $\log \hat{p}_\theta(y_{0:T})$ is a biased estimator.
- Can we correct for the bias?

Approximating $\log p_\theta(y_{0:T})$

- Can use bias correction based on Taylor series

$$\log(Z) = \log Z' + \frac{1}{Z'}(Z - Z') - \frac{1}{2Z'^2}(Z - Z')^2 + \mathcal{O}(Z^3)$$

  Let $Z' = E[Z]$ then ignoring higher order terms

$$\mathbb{E}[\log(Z)] \approx \log \mathbb{E}[Z] - \frac{1}{2\mathbb{E}[Z]^2}\mathbb{V}ar[Z]$$

- What we have is $Z = \widehat{Z} = \hat{p}_\theta(y_{0:T})$ and $Z' = p_\theta(y_{0:T})$

$$\mathbb{E}[\log\ \hat{p}_\theta(y_{0:T})] = \log p_\theta(y_{0:T}) - \frac{\mathbb{V}ar[\hat{p}_\theta(y_{0:T})]}{2p_\theta(y_{0:T})^2}$$

Approximating $\log p_\theta\left(y_{0:T}\right)$

- Bias reduction requires estimating $Var\left[\hat{p}_\theta\left(y_{0:T}\right)\right]$
  - Lee & Whiteley 2018
- Other possibility
  - use multiple runs
- Suppose $\frac{\mathbb{V}ar[\hat{p}_\theta(y_{0:T})]}{2p_\theta(y_{0:T})^2} \approx \frac{(\hat{W}_T - 1)}{2N}$

Approximating $\log p_\theta(y_{0:T})$

- We get then

$$E\left[\log\ \hat{p}_\theta(y_{0:T})\right] = \log\ \hat{p}_\theta(y_{0:T}) - \frac{(\hat{W}-1)}{2N}$$

- So can use

$$\log\ \widehat{p_\theta(y_{0:T})} = \log\ \hat{p}_\theta(y_{0:T}) + \frac{\hat{W}-1}{2N}$$

as a bias reduced estimator for ℓ_T

Optimising $\log p_\theta(y_{0:T})$ w.r.t θ

- Still $\hat{\ell}_T(\theta) = \log \widehat{p_\theta(y_{0:T})}$ will exhibit
  - quite a bit of variance
  - is discontinuous function w.r.t θ
- This can make finding maximum difficult
- Potential remedies:
  - smooth the approximation as a function of θ
  - use a different resampling scheme (Pitt 02, Lee 10)
  - try to reduce the variance with multiple runs

Expectation Maximisation

- Expectation Maximization (EM) algorithm is a very popular alternative procedure for maximizing $\ell_T(\theta)$.
- At iteration $k+1$, we set

$$\theta_{k+1} = \arg\max_{\theta} \; Q(\theta_k, \theta) \tag{4}$$

where

$$Q(\theta_k, \theta) = \int \log p_\theta(x_{0:T}, y_{0:T}) \; p_{\theta_k}(x_{0:T} | y_{0:T}) dx_{0:T}. \tag{5}$$

The sequence $\{\ell_T(\theta_k)\}_{k \geq 0}$ generated by this algorithm is non-decreasing.

Expectation Maximisation

- In particular if $p_\theta(x_{0:T}, y_{0:T})$ belongs to the exponential family, then the EM consists of computing a n_s-dimensional summary statistic like $\mathcal{S}_n^\theta$
- the maximizing argument of $Q(\theta_k, \theta)$ can be characterized explicitly through a suitable function $\Lambda : \mathbb{R}^{n_s} \to \Theta$, i.e.

$$\theta_{k+1} = \Lambda\left(\mathcal{S}_T^{\theta_k}\right). \tag{6}$$

- Particle implementation consists of computing $\mathcal{S}_n^{\theta_k}$

Additive functionals $\mathcal{S}_n^\theta$

- $\mathcal{S}_n^\theta$ is an additive functional

$$\mathcal{S}_n^\theta = \int \left[\sum_{k=0}^{n} s_k\left(x_k, x_{k-1}\right) \right] p_\theta\left(x_{0:n} \middle| y_{0:n}\right) dx_{0:n}, \tag{7}$$

- Theory tells that the asymptotic variance of the SMC estimate

$$\widehat{\mathcal{S}_n^\theta} = \int \left[\sum_{k=0}^{n} s_k\left(x_k, x_{k-1}\right) \right] \widehat{p}_\theta\left(dx_{0:n} \middle| y_{0:n}\right), \tag{8}$$

satisfies

$$\mathbb{V}\left(\widehat{\mathcal{S}_n^\theta}\right) \geq D_\theta \frac{n^2}{N}. \tag{9}$$

even with exponential filter stability.

- This motivates the use of dedicated smoothing algorithms

Gradient ascent

- The log-likelihood may be maximized with the following steepest ascent algorithm: at iteration $k+1$

$$\theta_{k+1} = \theta_k + \gamma_{k+1} \, \nabla_\theta \ell_T(\theta)|_{\theta=\theta_k} \, , \tag{10}$$

- $\{\gamma_k\}_{k\geq 1}$ needs to satisfy $\sum_k \gamma_k = \infty$ and $\sum_k \gamma_k^2 < \infty$.
  - could also use Hessian but omitted for simplicity
- To obtain the *score* vector $\nabla_\theta \ell_T(\theta)$ we can use Fisher's identity Fisher identity

$$\nabla_\theta \log p_\theta(y_{0:n}) = \int \nabla_\theta \log p_\theta(x_{0:n}, y_{0:n}) \, p_\theta(x_{0:n}| \, y_{0:n}) \, dx_{0:n}$$

- The latter is of the form of $\mathcal{S}_n^\theta$ again.

Gradient ascent

We have

$$\begin{aligned}\nabla_\theta \log p_\theta\left(x_{0:n}, y_{0:n}\right) &= \nabla_\theta \log \prod_{p=0}^{n} f_\theta\left(x_p \mid x_{p-1}\right) g_\theta\left(y_p \mid x_p\right) \\ &= \sum_{p=0}^{n} \left(\nabla \log f_\theta\left(x_p \mid x_{p-1}\right) + \nabla \log g_\theta\left(y_p \mid x_p\right)\right)\end{aligned}$$

- Define:

$$s_p(x_{p-1:p}) = \nabla \log f_\theta\left(x_p \mid x_{p-1}\right) + \nabla \log g_\theta\left(y_p \mid x_p\right).$$

- $\nabla_\theta \log p_\theta(y_{0:n})$ is of the form of $\mathcal{S}_n^\theta$ again.

Smoothing algorithms

- We are essentially interested in designing better particle approximations for $\{p_\theta(x_n | y_{0:T})\}_{n=0}^{T}$
- Some popular approaches
  - fixed lag smoothing
  - forward filtering backward sampling
  - forward filtering backward smoothing

Discussion

- Both FFBSa and FFBSm have computational cost is prop. to $N^2 T$ operations in total
- Assuming exponential forgetting of HMM:
  - $\mathcal{S}_n^\theta$ based on the fixed-lag approximation has an asymptotic variance with rate n/N with a non-vanishing (as $N \to \infty$) bias proportional to n and a constant decreasing exponentially fast with L.
  - The asymptotic bias and variance of the particle estimate of $\mathcal{S}_n^\theta$ computed using FFBSa/m satisfy:

$$\left| \mathbb{E}\left(\widehat{\mathcal{S}}_n^\theta\right) - \mathcal{S}_n^\theta \right| \leq F_\theta \frac{n}{N}, \quad \mathbb{V}\left(\widehat{\mathcal{S}}_n^\theta\right) \leq H_\theta \frac{n}{N}. \tag{11}$$

Discussion

- To compute $\widehat{\mathcal{S}}_n^\theta$ one can implement with cost $N^2 T$
  1. simple particle filter with N^2 particles
  2. FFBS particle filter with N particles
- Then
  - Case 1: suffers from path degeneracy
    - bias of order T/N^2
    - variance at least of order T^2/N^2
  - Case 2: more expensive
    - bias of order T/N
    - variance of order T/N

Numerical example

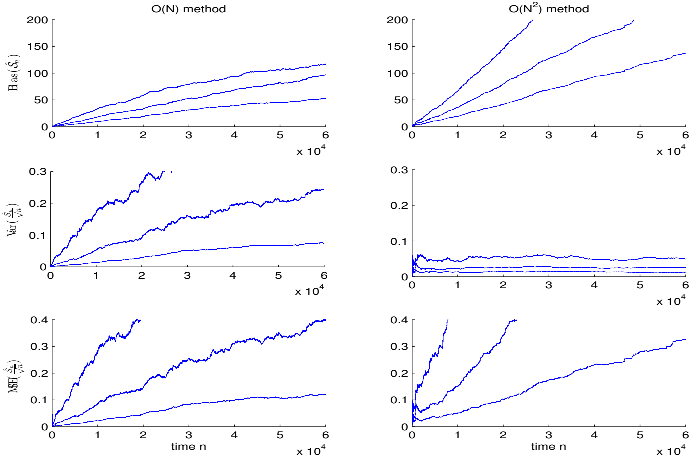

Figure: Estimating smoothed additive functionals: Empirical bias of the estimate of $\mathcal{S}_n^\theta$ (top panel), empirical variance (middle panel) and mean squared error (bottom panel) for the estimate of $\mathcal{S}_n^\theta/\sqrt{n}$.

Numerical example

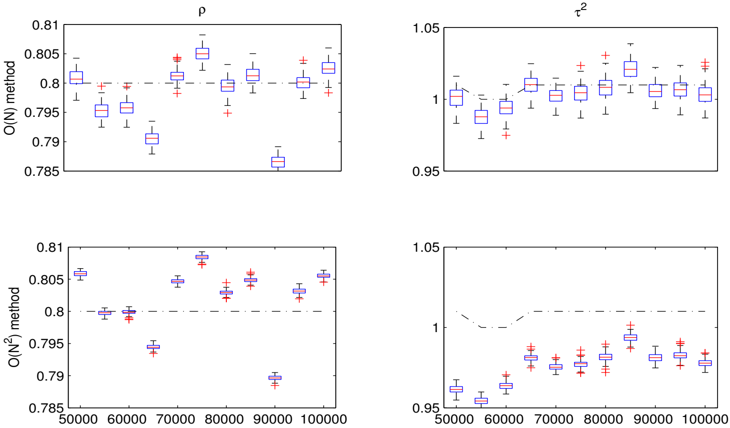

Figure: EM: Boxplots of $\hat{\theta}_n$ for $n \geq 5 \times 10^4$ using 100 realizations of the algorithms.

On-line methods

- On-line/ Forwards only extensions for EM and gradient methods do exist.
  - Poyiadjis, Doucet, Singh 11 Particle approximations of the score and observed information matrix...
  - Cappe 09 Online sequential Monte Carlo EM algorithm
  - Del Moral, Doucet, Singh 09 Forward Smoothing using Sequential Monte Carlo
  - Olsson and Westerborn 17 Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm

On-line methods

- For gradient method:

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \nabla \log p_{\theta_{\mathbf{0}:n}}(y_n | y_{0:n-1})$$

where $\nabla \log p_{\theta_{\mathbf{0}:n}}(y_n | y_{0:n-1})$ is defined as

$$\nabla \log p_{\theta_{\mathbf{0}:n}}(y_n | y_{0:n-1}) = \nabla \log p_{\theta_{\mathbf{0}:n-\mathbf{1}}, \boldsymbol{\theta}_n}(y_{0:n}) - \nabla \log p_{\theta_{\mathbf{0}:n-\mathbf{1}}}(y_{0:n-1}),$$

On-line methods

- The notation $\nabla \log p_{\theta_{0:n}}(y_{0:n})$ corresponds to a 'time-varying' score
  - which is computed with a filter using the parameter θ_p at time $p < n$.
- Using Fisher's identity to compute this 'time-varying' score, then we have for $1 \leq p \leq n$

$$s_p(x_{p-1:p}) = \nabla \log f_{\theta_p}\left(x_p \middle| x_{p-1}\right) + \nabla \log g_{\theta_p}\left(y_p \middle| x_p\right).$$

On-line methods

- In offline EM maximisation can be rewritten as

$$\theta_{k+1} = \Lambda\left(T^{-1}\mathcal{S}_T^{\theta_k}\right).$$

- So for on-line EM can use Robbins-Monro averaging

$$\begin{aligned}
&\mathcal{S}_{\theta_{0:n}} = \gamma_{n+1} \int s_n\left(x_{n-1:n}\right) p_{\theta_{0:n}}(x_{n-1}, x_n | y_{0:n}) dx_{n-1:n} \\
&+ (1-\gamma_{n+1}) \textstyle\sum_{k=0}^{n} \left(\displaystyle\prod_{i=k+2}^{n} (1-\gamma_i)\right) \gamma_{k+1} \\
&\times \int s_k\left(x_{k-1:k}\right) p_{\theta_{0:k}}(x_{k-1:k} | y_{0:k}) dx_{k-1:k},
\end{aligned}$$

- Then use standard maximization step is used as in the batch version:

$$\theta_{n+1} = \Lambda\left(\mathcal{S}_{\theta_{0:n}}\right).$$

- There is also a forward only implementation of FFBSm (Del Moral et. al. 2009)

Discussion

- On-line and offline parameter estimation drops down to computing smoothed integrals of additive functions
- Fair comparisons
  - either use standard algorithm (with $\mathcal{O}(N)$ cost) or dedicated smoothing algorithms (with $\mathcal{O}(N^2)$ cost)
- With the exception of on-line gradient methods when the same computational cost is used:
  - the first choice suffers from the variance
  - the second suffers from the bias
  - both give similar MSE
- PaRiS implements $\mathcal{O}(N^2)$ methods with less computational cost

Discussion

- Parameter estimation for HMMs is a challenging and exciting topic
- We have seen effective methods for:
  - low dimensional θ, X_n, Y_n
- We have not covered:
  - SMC^2, Particle Gibbs, Long/tall data, high dimensions, ...
- Review:
  - https://arxiv.org/pdf/1412.8695.pdf