Approximate Bayesian Computation and Particle Filters

Dennis Prangle

Reading University

5th February 2014

Introduction

Talk is mostly a literature review

A few comments on my own ongoing research

See Jasra "Approximate Bayesian Computation for a Class of Time Series Models" arXiv (2014) for a thorough overview

Background: ABC and state space models

Likelihood methods

Statistical inference often based on the likelihood

Probability (density) of data y_{obs} given parameters θ

Maximum likelihood: choose θ which maximises this

Bayes: use likelihood and prior to form posterior parameter distribution

Both rely on many evaluations of the likelihood for different θ

Likelihood-free methods

For complex models likelihood evaluation can be too expensive/impossible (examples later)

But often quick to simulate data given parameters

Motivates likelihood-free methods based on simulation

General idea:

- Simulate data y_{sim} from many θs
- Select θs giving $y_{\text{sim}} \approx y_{\text{obs}}$

These methods typically give approximate results

Approximate Bayesian Computation (ABC)

Puts likelihood-free idea into roughly Bayesian framework

Simple algorithm is to repeat these steps N times

- Draw θ from prior
- Draw $y_{\text{sim}}|\theta$ from model
- Accept θ if $d(y_{\text{sim}}, y_{\text{obs}}) \leq \epsilon$

Output θs are sample from approximation to the posterior

Approximation quality improves as $\epsilon \to 0$

But sample size decreases

Choice of ϵ is a trade-off

Curse of dimensionality / summary statistics

Quality of results worsens for high dimensional data

Can replace data y with low dimensional summary statistics $S(y)$

i.e. accept if $d(S(y_{\text{sim}}), S(y_{\text{obs}})) \leq \epsilon$

Some research in how to choose S well

e.g. Fearnhead and Prangle (2012)

But this adds further layers of approximation and tuning

For state space models better approaches possible

State space models

Assume there is a latent Markov chain $X_1, X_2, \ldots, X_T$

Observations are $Y_1, Y_2, \ldots, Y_T$

Y_i is conditionally independent of everything given X_i

Informally: Y_i depends only on X_i

Only Y_is observed. X_is known as hidden/latent states.

Model sometimes referred to as a latent or hidden Markov model.

State space models

Often there is an evolution density:

$$\pi(X_{t+1} = x_{t+1} | X_t = x_t, \theta) = g(x_{t+1} | x_t, \theta)$$

And an observation density:

$$\pi(Y_t = y_t | X_t = x_t, \theta) = h(y_t, | x_t, \theta)$$

(Dependence on t possible as well)

Standard particle filter requires tractable observation density

State space model inference goals

- Parameter inference: $\theta | y_1, y_2, \ldots, y_t$ (learn parameters)
- Filtering: $x_t | y_1, y_2, \ldots, y_t$ (learn current state)
- Smoothing: $x_1, x_2, \ldots, x_t | y_1, y_2, \ldots, y_t$ (learn historic states)
- Prediction: $x_{t+1} | y_1, y_2, \ldots, y_t$ (learn future state)

Main example: alpha-stable model

α-stable distribution is a model for heavy-tailed data

Often used in finance

Key parameter is α

Valid range is $0 < \alpha \leq 2$

$\alpha = 2$ is normal distribution

Smaller α gives heavier tails

e.g. $\alpha = 1$ gives Cauchy distribution

For most α values the density does not have a closed form

Distribution also has location and scale parameters

Main example: alpha-stable model

Latent states and observation both scalars

Some Markov model for latent states

Observation Y_i is X_i plus α-stable draw

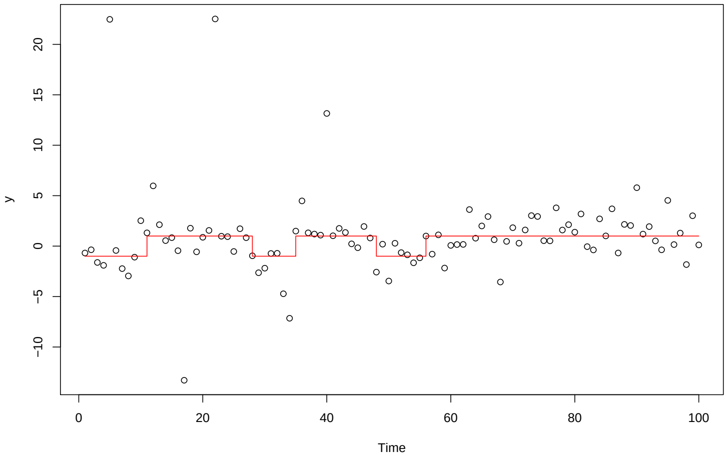
Application: daily log returns of a stock

X_i is "state of economy"

Y_i is X_i plus short term variation - sometimes very large!

Simplest case is X_i constant - iid data

Main example: plot



Other examples

(X_t, Y_t) together form a Markov process

X_t is unobserved

Y_t is observed exactly

This can be put into state space form

Observation density often not tractable

Several applications e.g. chemical reactions, infectious diseases, population dynamics

ABC filtering

ABC particle filter

Input: parameters θ, bandwidth $\epsilon > 0$, number of particles N, data $y_1^{\text{obs}}, \ldots, y_T^{\text{obs}}$

1. Initialise: let $t = 1$.
2. Sample $x^{(i)}$ values from prior for $1 \leq i \leq N$
3. Simulate observations $y^{(i)}$ from $\pi(y|x^{(i)}, \theta)$
4. Accept/reject

   $$\text{Let } w^{(i)} = \begin{cases} 1 & \text{for } ||y_t^{\text{obs}} - y^{(i)}||_2 \leq \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$
5. Increment t, resample and propagate particles.
   Return to step 3.

Variations

- Alternative distance metric
- Smooth weights
- Summary statistics/data transformations
- More later

Filtering/smoothing output

Consider computing some function of states (e.g. mean)

Can compare algorithm output estimate with true value

Filtering problem: error is $O(\epsilon)$, not affected by amount of data

Smoothing problem: error is $O(T\epsilon)$

Strong assumptions required

Error can be controlled by increasing computational effort

Likelihood estimate

Recall likelihood is density of observations given θ

Can get an estimate from particle filter

In ABC this is product of acceptance frequencies (and constant)

i.e. $C(\epsilon)\frac{N_{\text{acc}}^{(1)}}{N}\frac{N_{\text{acc}}^{(2)}}{N}\ldots$

Converges to the true value for $N \to \infty, \epsilon \to \infty$ appropriately (Lebesgue differentiation theorem)

Degeneracy

A problem with ABC particle filtering is degeneracy

Zero acceptances at some iteration means algorithm cannot continue

Smooth weights not much help

Possible solution later

Parameter inference

Strategy

Get likelihood estimates at various θs using ABC PF

Use to approximate maximum likelihood estimator

Or to construct posterior distribution of θ

Bayesian approach

Use an MCMC algorithm on θ

For each θ proposed, make an associated likelihood estimate $\hat{L}(\theta)$

The stationary distribution is

$$\propto \text{prior}(\theta) \times E[\hat{L}(\theta)|\theta]$$

This is posterior distribution if $\hat{L}(\theta)$ unbiased

Not true for ABC so get an approximate posterior

Maximum likelihood approach

Stochastic gradient ascent algorithms

Given θ estimate gradient of log likelihood

This can be done from ABC output (various papers)

Move along direction of greatest increase

Several papers on implementation details

This maximises $E[\hat{L}(\theta)|\theta]$

Comparison of methods

MLE approach faster

MLE approach can be updated quickly as new data arrive

Bayesian approach gives uncertainty quantification

Both currently require considerable tuning and assumptions

Approximation quality

Both approaches effectively use an approximate likelihood

$$L_\epsilon(\theta) = E[\hat{L}(\theta)|\theta]$$

With MLE θ^*_ϵ

Typically not equal to correct MLE, θ^*

But as $\epsilon \to 0$, $\theta^*_\epsilon \to \theta^*$

Consistency

Consider $\epsilon > 0$ and $T \to \infty$

i.e. large amount of data

Under some conditions true MLE converges to true parameter values ("consistency")

And θ^*_ϵ converges to a nearby value

Noisy ABC: consistency

Noisy ABC adds iid noise to the observations before performing ABC inference

From a particular noise observation (see next slide)

Achieves consistency: $\lim_{T\to\infty} \theta^*_\epsilon$ is correct even for $\epsilon > 0$!

With enough data this approx method learns true parameters

But adding noise increases variance

Noisy ABC: theory

ABC likelihood can be shown to be likelihood of a perturbed model

Perturbation is adding measurement error

Error distribution is uniform with radius ϵ

Noisy ABC adds iid noise from same dist so that observations are now "model+noise"

And the perturbed model is in fact correct

This gives consistency

Avoiding degeneracy

Alive particle filter: motivation

Consider an iteration of ABC PF with data y_i

N simulations performed

Degeneracy if all rejected

Alive PF uses adaptive N

Keeps simulating until M acceptances

(M prespecified)

Avoids degeneracy, has random run-time

Alive particle filter: technical details

Care is needed in alive PF to get good likelihood estimate properties

But details not needed in this talk

See Jasra et al (2013)

Efficiency comparison

Work in progress!

Comparing number of simulations required

For standard ABC PF, this is number needed to avoid degeneracy with given prob

Standard ABC PF cost is at best $O(T \log T)$

Alive PF cost is at best $O(T)$

Efficiency comparison: poor conditions

Both PFs have much higher cost in following situation

Tails of observations are heavy in comparison with tail of proposals

Efficiency comparison: improving performance

- Transforming data to avoid heavy tails helpful
- Modify alive PF to quit early once we know likelihood estimate is low

Paper later in year hopefully!

Conclusion

Summary 1

- ABC allows inference based on model simulations only
- Results are approximate
- Summary statistics often necessary for high dim data, but add further approximation
- For simple state space models ABC particle filter can avoid summary stats
- Analyses each data point in sequence

Summary 2

Filtering, smoothing, parameter inference discussed

Theory on convergence for $\epsilon \to 0$

Noisy ABC allow consistent estimates

Alive PF avoid degeneracy problems

Future directions

Tuning - in particular ϵ

Improving noisy ABC (e.g. multiple noise realisations)

Faster algorithms

More general theory - lots of assumptions currently needed

Higher dimensional data

Choosing between models

Alternative likelihood-free approaches (e.g. via expectation propagation)