Towards Automatic Bayesian Model Comparison: A Sequential Monte Carlo Approach

Yan Zhou*, **Adam M. Johansen****, John A. D. Aston†

∗ NUS, Singapore; ∗∗ University of Warwick; † University of Cambridge

a.m.johansen@warwick.ac.uk

www2.warwick.ac.uk/fac/sci/statistics/staff/academic/johansen/talks/

JSM: August 2nd, 2016

Outline

- Goals
- Background
  - (Bayesian) Model Comparison
  - Sequential Monte Carlo (SMC)
- One SMC Algorithm
- Adapative SMC
  - Sequence of Distributions
  - Proposal Distributions
- Illustrative Examples
  - A Gaussian Mixture Model
  - A Positron Emission Tomography
- Conclusions

Goals

Automatic Bayesian Model Comparison

- Robust approximation of marginal likelihood (evidence);
- or Bayes factors;
- with minimal application-specific tuning.

Caveats: *Towards* Automatic Bayesian Model Comparison

- We don't consider philosophical issues or prior specification.
- Performance is undoubtedly *improved* by customization.
- Sufficiently difficult problems will *require* customization.

Goals

Automatic Bayesian Model Comparison

- Robust approximation of marginal likelihood (evidence);
- or Bayes factors;
- with minimal application-specific tuning.

Caveats: *Towards* Automatic Bayesian Model Comparison

- We don't consider philosophical issues or prior specification.
- Performance is undoubtedly *improved* by customization.
- Sufficiently difficult problems will *require* customization.

Bayesian Model Comparison

- Here we consider a finite collection of candidates, $\mathcal{K}$
- Prior over models: $\pi(k) = \mathbb{P}(M = k)$
- Model k prior: $\pi(\theta_k | M = k)$
- Model k likelihood: $p(\mathbf{y}|\theta_k, M = k)$
- Evidence:

$$p(\mathbf{y}|M = k) = \int p(\mathbf{y}|\theta_k, M = k)\pi(\theta_k|M = k)\pi(k)d\theta_k$$

- Posterior probabilities:

$$\mathbb{P}(M = k|\mathbf{y}) = \frac{\pi(k)p(\mathbf{y}|M = k)}{\sum_{k' \in \mathcal{K}} \pi(k')p(\mathbf{y}|M = k')}$$

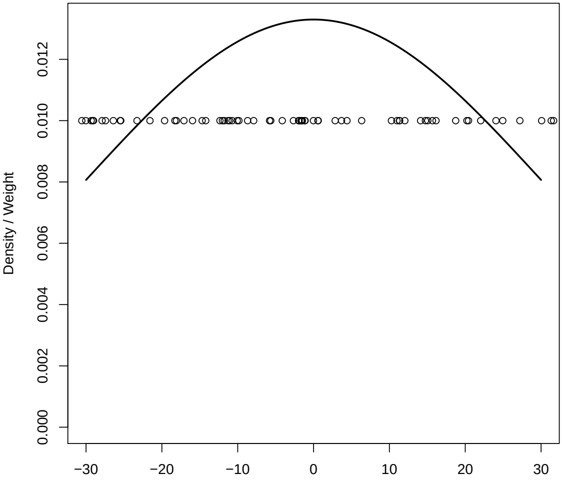
- Bayes Factors:

$$B_{k,k'} = \mathbb{P}(M = k|\mathbf{y})/\mathbb{P}(M = k'|\mathbf{y})$$

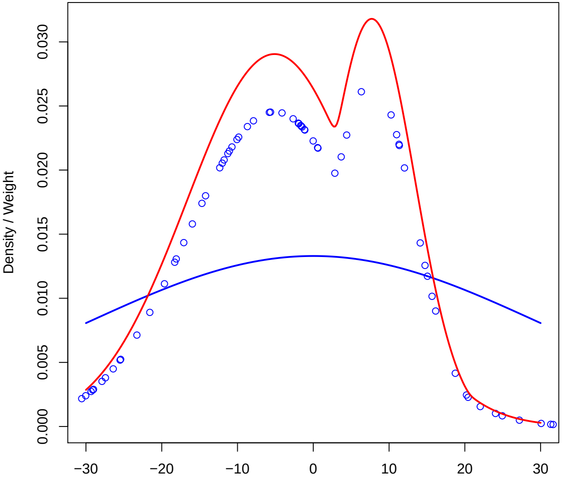
Sequential Monte Carlo Samplers [2]

- *Very* general sampling framework.
- We focus on a special case:
  - Given $\pi_0, \ldots, \pi_T$ where $\pi_t = \gamma_t / Z_t$ and Z_t is unknown,
  - iteratively, *weight*, *resample* and *move* a population of samples, to obtain
  - an unbiased estimate of Z_T / Z_0 and a "properly weighted" sample targetting π_T.
  - Example: $\pi_0 =$ prior and $\pi_T =$ posterior.
- Now reasonably well characterized theoretically, e.g.:
  - SLLN;
  - $\sqrt{N}$-CLT.
- Potentially more robust than standard MCMC approaches.
- Amenable to adaptation.

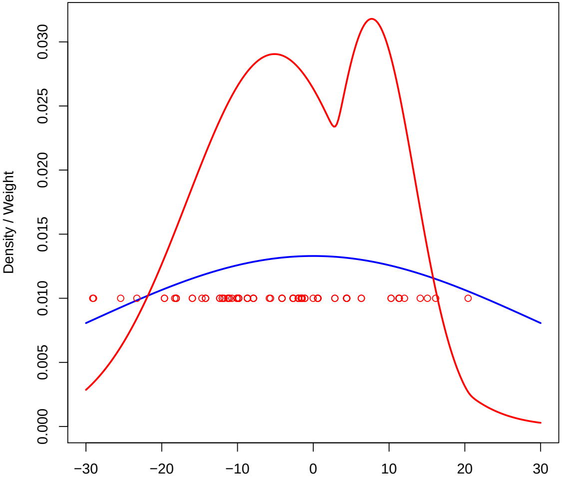
Simple Illustration of SMC I



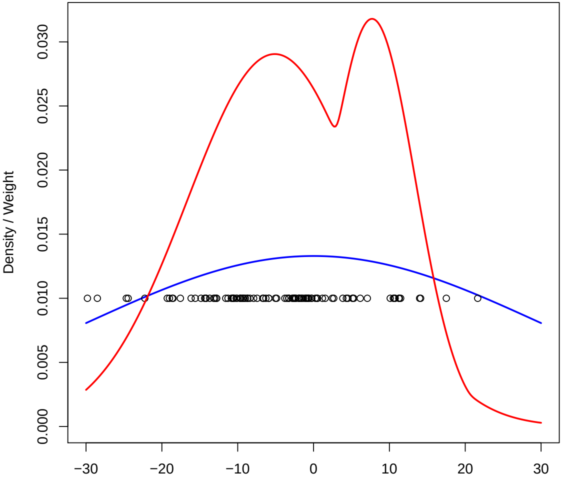
Simple Illustration of SMC II



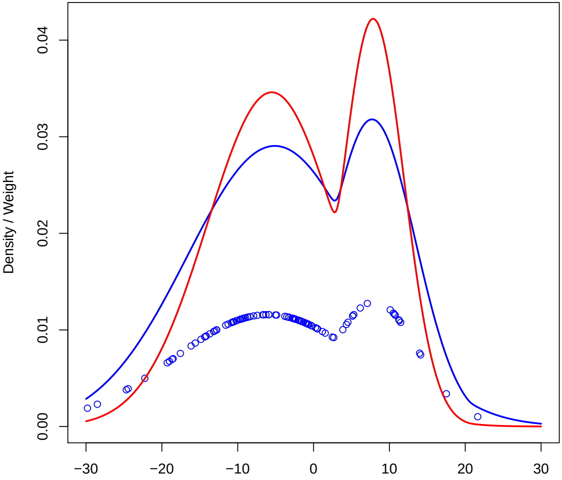
Simple Illustration of SMC III



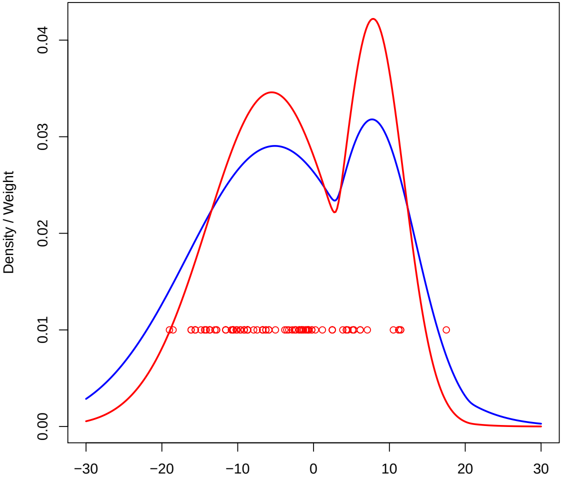
Simple Illustration of SMC IV



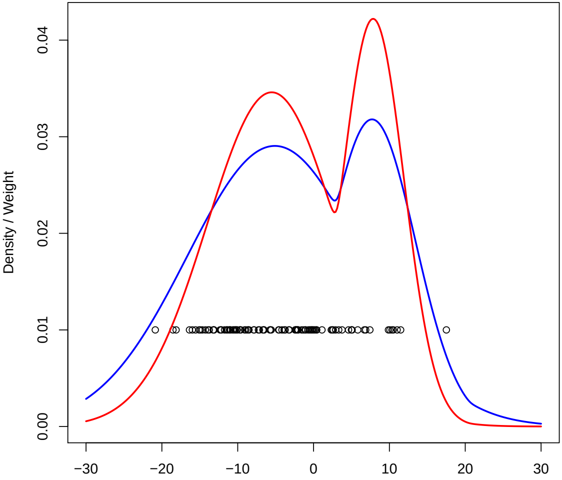
Simple Illustration of SMC V



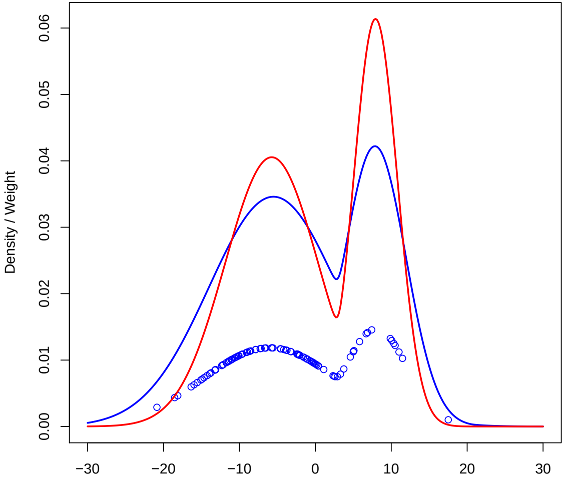
Simple Illustration of SMC VI



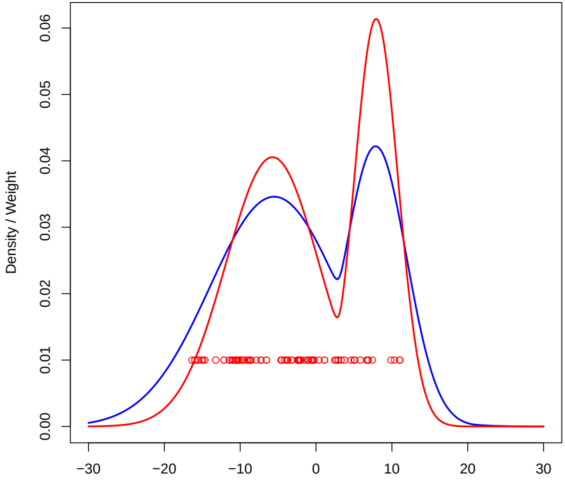
Simple Illustration of SMC VII



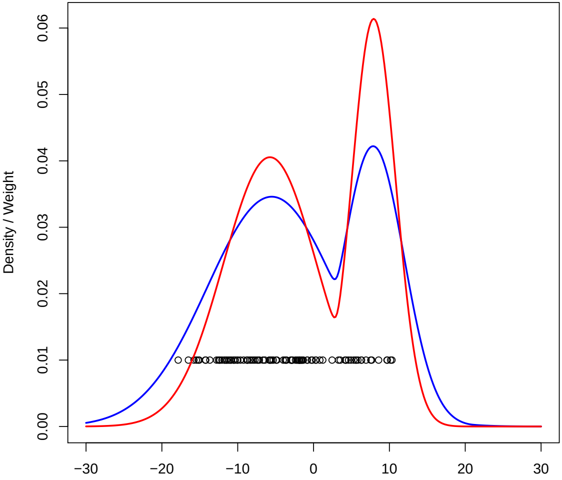
Simple Illustration of SMC VIII



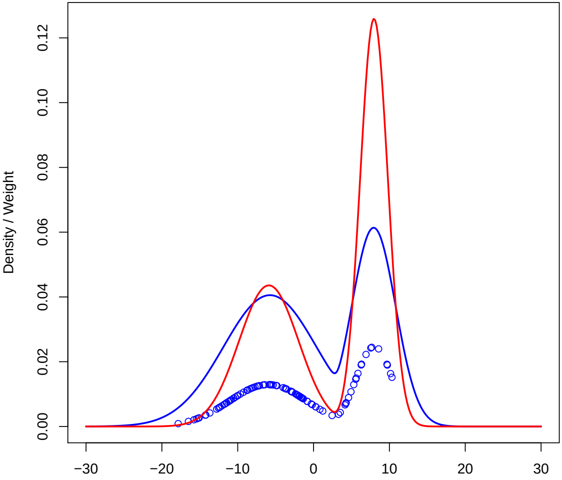
Simple Illustration of SMC IX



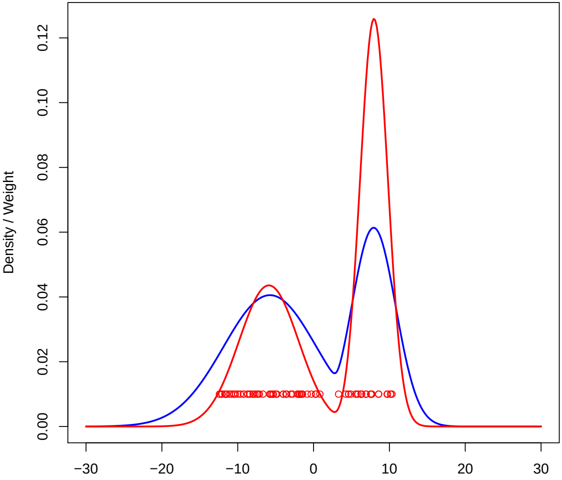
Simple Illustration of SMC X



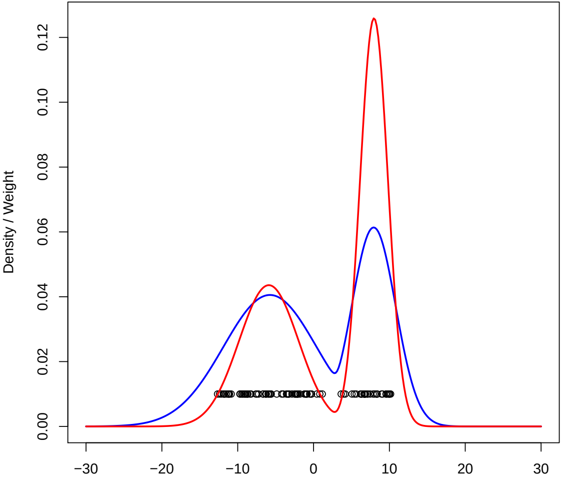
Simple Illustration of SMC XI



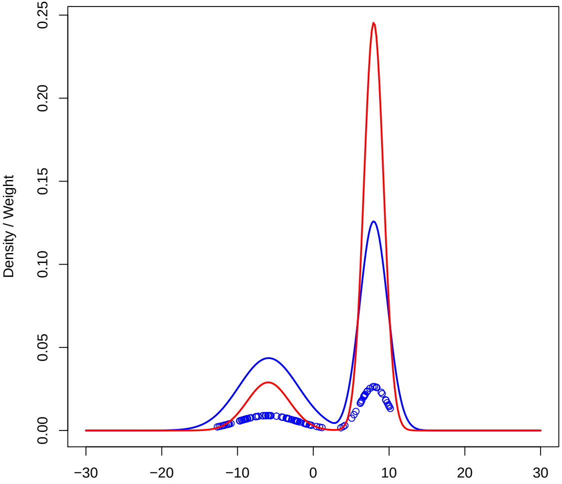
Simple Illustration of SMC XII



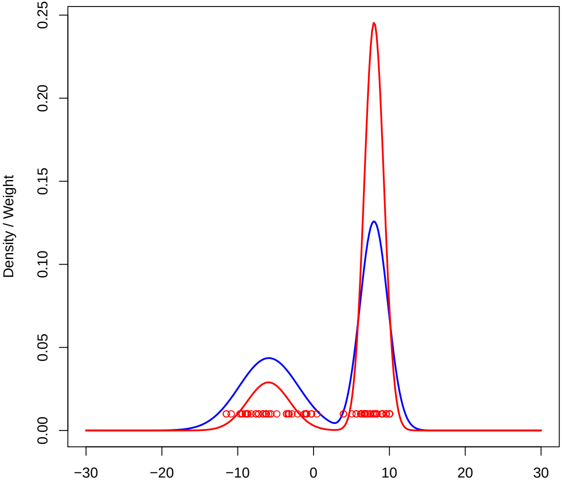
Simple Illustration of SMC XIII



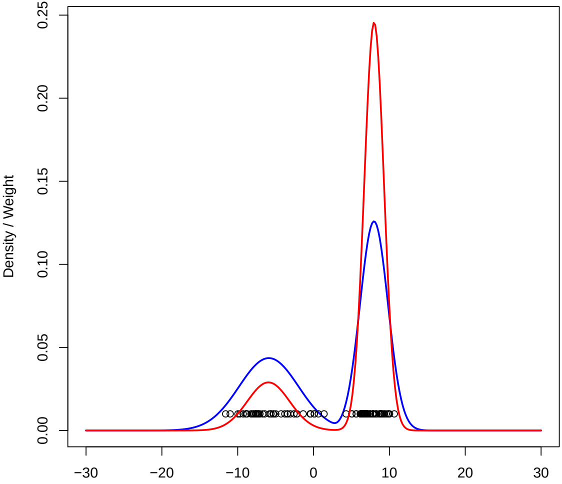
Simple Illustration of SMC XIV



Simple Illustration of SMC XV



Simple Illustration of SMC XVI



The Basic Algorithm [SMC2-DS] — For each model, $k \in \mathcal{K}$

Initialisation: Set $t \leftarrow 0$.

Sample $\theta_0^{(k,i)} \sim \pi(\cdot|M_k)$.

Set $W_0^{(k,i)} = 1/N$.

Iteration: Set $t \leftarrow t+1$.

Weight $W_t^{(k,i)} \propto W_{t-1}^{(k,i)} p(\mathbf{y}|\theta_{t-1}^{(k,i)}, M_k)^{\alpha(t/T_k)-\alpha([t-1]/T_k)}$.

Apply resampling if necessary.

Sample $\theta_t^{(k,i)} \sim K_t(\cdot|\theta_{t-1}^{(k,i)})$, a $\pi_t^{(k)}$-invariant kernel.

Repeat the *Iteration* step *until* $t = T_k$.

Where:

- $\alpha : [0,1] \mapsto [0,1]$ is an increasing bijection
- $\pi_t^{(k)}(\theta) \propto \pi(\theta|M_k) \cdot p(\mathbf{y}|\theta, M_k)^{\alpha(t/T_k)}$
- An unbiased estimate of $p(\mathbf{y}|M_k) = \int p(\mathbf{y}|\theta_k, M_k) p(\theta_k|M_k) d\theta_k$ is a byproduct.

Some Related Alternatives

Many other approaches are possible:

- Mimic reversible jump using one (or more) SMC samplers.
- Approximate Bayes factors directly.
- Use *path sampling / thermodynamic integration* as an alternative estimator of the normalizing constant.

and there are some competing strategies, particularly:

- Reversible Jump MCMC [3]
- Annealed Importance Sampling [6]
- Population MCMC (parallel tempering), e.g., [1]

Adaptation: MCMC Kernels

- Like MCMC we can adapt the proposal kernels used.
- Unlike MCMC:
  - We have historical information.
  - We do not depend upon ergodicity.
- Strategy employed here, roughly speaking:
  - Estimate variance and each target distribution; rescale appropriately to obtain proposal for next iteration.

Adaptation: Sequence of Distributions

- But, what should T or $\pi_1, \ldots, \pi_{T-1}$ be?
- Weights at time t depend on samples at $t-1$ and π_t
- so, we can choose π_t based on $(W^i_{t-1}, \theta^i_{t-1})_{i=1}^N$.
- Heuristically, want $||\pi_t - \pi_{t-1}||$ to be similar for all t.
- The χ^2-divergence is a natural criterion for importance sampling:

$$d_{\chi^2}(\pi_{t-1}, \pi_t) = \int \left(\frac{\pi_t(\theta)}{\pi_{t-1}(\theta)} \right)^2 \pi_{t-1}(\theta) d\theta - 1$$

- and can be approximate using an N-sample from π_{t-1}

$$\widehat{d_{\chi^2}}(\pi_{t-1}, \pi_t) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\pi_t(\theta^i)}{\pi_{t-1}(\theta^i)} \right)^2 - 1.$$

Conditional Effective Sample Size (CESS)

- "Exact ESS" of an N-sample from π_{t-1} targetting π_t is [4]:

$$\text{Exact ESS} = \frac{N}{1 + \text{var}_{\pi_{t-1}}(\frac{d\pi_t}{d\pi_{t-1}})} \tag{1}$$

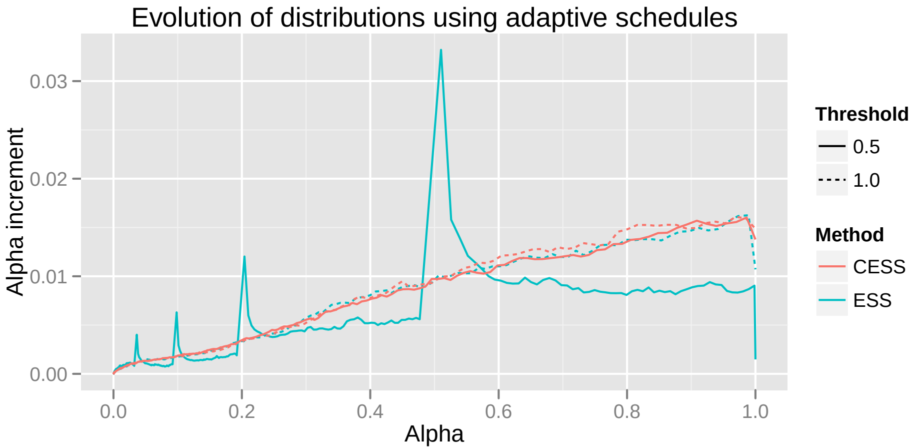
- approximated by replacing $1 + \text{var}_{\pi_{t-1}}(\frac{d\pi_t}{d\pi_{t-1}})$ with the empirical mean squared normalised importance weights:

$$\text{ESS} = N \Big/ \left(\frac{\sum_{i=1}^{N} (w_t^i)^2}{(\sum_{j=1}^{N} w_t^i)^2} \right) = \frac{N}{\sum_{i=1}^{N} (W_t^i)^2}$$

- the CESS is closely related:

$$\frac{N}{\sum_{i=1}^{N} W_{t-1}^i (\frac{d\pi_t}{d\pi_{t-1}}(X_{t-1}^i))^2} \approx \frac{N}{\sum_{i=1}^{N} W_{t-1}^i (\frac{w_t^i}{\sum_{j=1}^{N} W_{t-1}^j w_t^j})^2} =: CESS.$$

CESS/ESS in Specifying Distribution Seqeunces



Example: Gaussian Mixture Model

- Data $\mathbf{y} = (y_1, \ldots, y_n)$ are iid

$$y_i|\theta_r \sim \sum_{j=1}^{r} \omega_j \mathcal{N}(\mu_j, \lambda_j^{-1})$$

- Parameters $\theta_r = (\mu_{1:r}, \lambda_{1:r}, \omega_{1:r})$ and r is the number of components. The priors are taken to be the same for all components: $\mu_j \sim \mathcal{N}(\xi, \kappa^{-1})$, $\lambda_j \sim \mathcal{G}(\nu, \chi)$ and $\omega_{1:r} \sim \mathcal{D}(\rho)$
- Kernel: composition of MH kernels:

  $\mu_{1:r}$ using a Normal random walk proposal.
  $log(\lambda_{1:r})$ using a Normal random walk.
  $\omega_{1:r}$ using a Normal random walk on logit scale.
  Scales tuned to yield approximately constant acceptance rates.

GMM Results

Simulating 100 observations from a four components model with $\mu_{1:4} = (-3, 0, 3, 6)$, and $\lambda_j = 2$, $\omega_j = 0.25$, $j = 1, \ldots, 4$.

Basic Algorithms

	Algorithms						
Quantity	SMC2-DS	SMC2-PS	SMC3-DS	SMC3-PS	AIS-DS	AIS-PS	PMCMC
$\log B_{4,5}$	2.15	2.15	2.16	2.21	2.16	2.17	2.63
sd	0.25	**0.22**	0.61	0.62	1.12	1.10	0.41

Adaptive proposals: SMC2 achieves essentially identical performance without tuning.

Adaptive distributions: using CESS SMC2 sd fell by around 20% relative to the best manual tuning.

Example: Positron Emission Tomography

An m-compartmental model has generative form:

$$y_j = C_T(t_j; \phi_{1:m}, \theta_{1:m}) + \sqrt{\frac{C_T(t_j; \phi_{1:m}, \theta_{1:m})}{t_j - t_{j-1}}}\varepsilon_j \qquad (2)$$

$$C_T(t_j; \phi_{1:m}, \theta_{1:m}) = \sum_{i=1}^{m} \phi_i \int_0^{t_j} C_P(s) e^{-\theta_i(t_j - s)} ds \qquad (3)$$

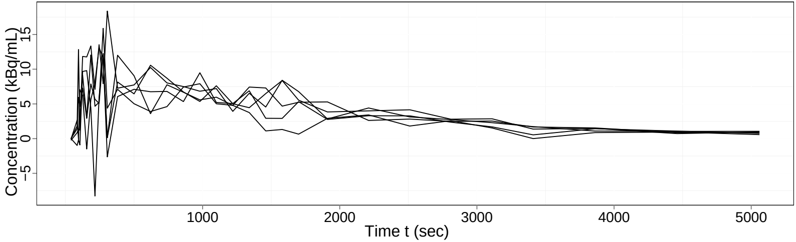
where t_j is the measurement time of y_j, $\varepsilon_j \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$ is additive measurement error and input function C_P is (treated as) known; parameters $\phi_1, \theta_1, \ldots, \phi_m, \theta_m$ characterize the model dynamics.

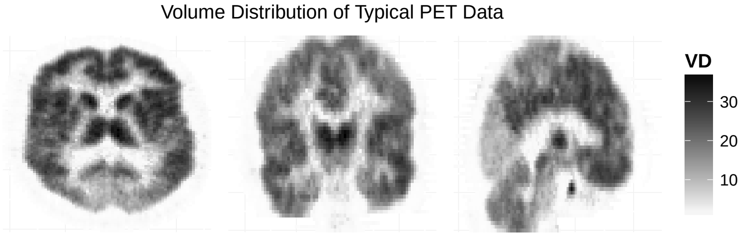
Proposal scales			Manual			Adaptive
Annealing scheme			Prior (5)	Posterior (5)	Adaptive	
T	N	Algorithm	Marginal likelihood estimates ($\log p(y \mid M_k) \pm$ sd)			
500	30	PMCMC	-39.1 ± 0.56	-926.8 ± 376.99		
500	192	SMC2-DS	-39.2 ± 0.25	-39.7 ± 1.06	-39.2 ± 0.18	-39.1 ± 0.12
		SMC2-PS	-39.2 ± 0.25	-91.3 ± 21.69	-39.2 ± 0.18	-39.1 ± 0.13
100	960	SMC2-DS	-39.3 ± 0.36	-40.6 ± 1.41	-39.2 ± 0.31	-39.2 ± 0.19
		SMC2-PS	-39.3 ± 0.35	302.1 ± 46.29	-39.3 ± 0.31	-39.2 ± 0.18
5000	30	PMCMC	-39.3 ± 0.21	-917.6 ± 129.54		
5000	192	SMC2-DS	-39.2 ± 0.09	-39.2 ± 0.20	-39.2 ± 0.08	-39.1 ± 0.04
		SMC2-PS	-39.2 ± 0.09	-43.8 ± 2.13	-39.2 ± 0.08	-39.1 ± 0.04
1000	960	SMC2-DS	-39.2 ± 0.08	-39.2 ± 0.31	-39.2 ± 0.07	-39.2 ± 0.03
		SMC2-PS	-39.2 ± 0.08	-65.7 ± 5.54	-39.2 ± 0.07	-39.2 ± 0.03

Proposal scales			Manual			Adaptive
Annealing scheme			Prior (5)	Posterior (5)	Adaptive	
T	N	Algorithm	Bayes factor estimates ($\log B_{2,1} \pm$ sd)			
500	30	PMCMC	1.7 ± 0.62	-70.9 ± 525.79		
500	192	SMC2-DS	1.6 ± 0.27	1.3 ± 1.13	1.6 ± 0.20	1.6 ± 0.15
		SMC2-PS	1.6 ± 0.27	-3.9 ± 30.02	1.6 ± 0.20	1.6 ± 0.15
100	960	SMC2-DS	1.6 ± 0.37	0.5 ± 1.55	1.6 ± 0.34	1.6 ± 0.21
		SMC2-PS	1.6 ± 0.37	-13.1 ± 66.30	1.6 ± 0.33	1.6 ± 0.21
5000	30	PMCMC	1.6 ± 0.24	-60.3 ± 198.10		
5000	192	SMC2-DS	1.6 ± 0.10	1.6 ± 0.23	1.6 ± 0.09	1.6 ± 0.05
		SMC2-PS	1.6 ± 0.10	1.3 ± 2.98	1.6 ± 0.09	1.6 ± 0.05
1000	960	SMC2-DS	1.6 ± 0.09	1.6 ± 0.33	1.6 ± 0.08	1.6 ± 0.04
		SMC2-PS	1.6 ± 0.09	-0.2 ± 6.63	1.6 ± 0.08	1.6 ± 0.04

Real data from an opioid receptor study

Turning $> 200,000$ measured time series into estimates in 2 hours:





Conclusions

- SMC provides a flexible and powerful framework for estimating (ratios of) normalising constants.
- Adaptation of proposals, distribution sequences is easy and effective.
- Empirically it outperforms the state of the art for comparison of finite collections of models in the examples considered.
- Allows application to very large numbers of data sets without fine-tuning.
- Flexible library facilitates fast C++ implementation [7].
- We can go much further. . . e.g. [5].

References

B. Calderhead and M. A. Girolami.
Estimating Bayes factors via thermodynamic integration and population mcmc.
Computational Statistics and Data Analysis, 53:4028–4045, 2009.

P. Del Moral, A. Doucet, and A. Jasra.
Sequential Monte Carlo samplers.
Journal of the Royal Statistical Society B, 63(3):411–436, 2006.

P. J. Green.
Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination.
Biometrika, 82:711–732, 1995.

A. Kong, J. S. Liu, and W. H. Wong.
Sequential imputations and Bayesian missing data problems.
Journal of the American Statistical Association, 89(425):278–288, March 1994.

F. Lindsten, A. M. Johansen, C. A. Næsseth, B. Kirkpatrick, T. Schön, J. A. D. Aston, and A. Bouchard-Côté.
Divide and conquer with sequential Monte Carlo samplers.
Technical Report 1406.4993, ArXiv Mathematics e-prints, 2014.

R. M. Neal.
Annealed importance sampling.
Statistics and Computing, 11:125–139, 2001.

Y. Zhou.
vSMC: Parallel sequential Monte Carlo in C++.
Mathematics e-print 1306.5583, ArXiv, 2013.

Y. Zhou, A. M. Johansen, and J. A. D. Aston.
Towards automatic model comparison: An adaptive sequential Monte Carlo approach.
Journal of Computational and Graphical Statistics, 2015.
In press.