Bayesian Estimation of DSGE Models[1]
Chapter 9: Particle Filters & MH Samplers

Ed Herbst[1] Frank Schorfheide[2]

[1]Federal Reserve Board

[2]University of Pennsylvania

February 9, 2016

[1]The views expressed in this paper are those of the authors and do not necessarily reflect the views of the Federal Reserve Board of Governors or the Federal Reserve System.

Embedding PF Likelihoods into Posterior Samplers

- Likelihood functions for nonlinear DSGE models can be approximated by the PF.
- We will now embed the likelihood approximation into a posterior sampler: PFMH Algorithm (a special case of PMCMC).
- The book also discusses SMC^2.

Embedding PF Likelihoods into Posterior Samplers

- Distinguish between:
  - $\{p(Y|\theta), p(\theta|Y), p(Y)\}$, which are related according to:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}, \quad p(Y) = \int p(Y|\theta)p(\theta)d\theta$$

  - $\{\hat{p}(Y|\theta), \hat{p}(\theta|Y), \hat{p}(Y)\}$, which are related according to:

$$\hat{p}(\theta|Y) = \frac{\hat{p}(Y|\theta)p(\theta)}{\hat{p}(Y)}, \quad \hat{p}(Y) = \int \hat{p}(Y|\theta)p(\theta)d\theta.$$

- Surprising result (Andrieu, Docet, and Holenstein, 2010): under certain conditions we can replace $p(Y|\theta)$ by $\hat{p}(Y|\theta)$ and still obtain draws from $p(\theta|Y)$.

PFMH Algorithm

For $i = 1$ to N:

1. Draw ϑ from a density $q(\vartheta|\theta^{i-1})$.
2. Set $\theta^i = \vartheta$ with probability

$$\alpha(\vartheta|\theta^{i-1}) = \min\left\{1,\ \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{i-1})}{\hat{p}(Y|\theta^{i-1})p(\theta^{i-1})/q(\theta^{i-1}|\vartheta)}\right\}$$

   and $\theta^i = \theta^{i-1}$ otherwise. The likelihood approximation $\hat{p}(Y|\vartheta)$ is computed using a particle filter.

Why Does the PFMH Work?

- At each iteration the filter generates draws $\tilde{s}_t^j$ from the proposal distribution $g_t(\cdot|s_{t-1}^j)$.
- Let $\tilde{S}_t = \left(\tilde{s}_t^1, \ldots, \tilde{s}_t^M\right)'$ and denote the entire sequence of draws by $\tilde{S}_{1:T}^{1:M}$.
- Selection step: define a random variable A_t^j that contains this ancestry information. For instance, suppose that during the resampling particle $j = 1$ was assigned the value $\tilde{s}_t^{10}$ then $A_t^1 = 10$. Let $A_t = \left(A_t^1, \ldots, A_t^N\right)$ and use $A_{1:T}$ to denote the sequence of A_t's.
- PFMH operates on an enlarged probability space: θ, $\tilde{S}_{1:T}$ and $A_{1:T}$.

Why Does the PFMH Work?

- Use $U_{1:T}$ to denote random vectors for $\tilde{S}_{1:T}$ and $A_{1:T}$. $U_{1:T}$ is an array of *iid* uniform random numbers.
- The transformation of $U_{1:T}$ into $(\tilde{S}_{1:T}, A_{1:T})$ typically depends on θ and $Y_{1:T}$, because the proposal distribution $g_t(\tilde{s}_t|s^j_{t-1})$ depends both on the current observation y_t as well as the parameter vector θ.
- E.g., implementation of conditionally-optimal PF requires sampling from a $N(\bar{s}^j_{t|t}, P_{t|t})$ distribution for each particle j. Can be done using a prob integral transform of uniform random variables.
- We can express the particle filter approximation of the likelihood function as

$$\hat{p}(Y_{1:T}|\theta) = g(Y_{1:T}|\theta, U_{1:T}).$$

where

$$U_{1:T} \sim p(U_{1:T}) = \prod_{t=1}^{T} p(U_t).$$

Why Does the PFMH Work?

- Define the joint distribution

$$p_g\big(Y_{1:T}, \theta, U_{1:T}\big) = g(Y_{1:T}|\theta, U_{1:T})p\big(U_{1:T}\big)p(\theta).$$

- The PFMH algorithm samples from the joint posterior

$$p_g\big(\theta, U_{1:T}\big|Y_{1:T}\big) \propto g(Y|\theta, U_{1:T})p\big(U_{1:T}\big)p(\theta)$$

  and discards the draws of $\big(U_{1:T}\big)$.

- For this procedure to be valid, it needs to be the case that PF approximation is unbiased:

$$\mathbb{E}[\hat{p}(Y_{1:T}|\theta)] = \int g(Y_{1:T}|\theta, U_{1:T})p\big(U_{1:T}\big)d\theta = p(Y_{1:T}|\theta).$$

Why Does the PFMH Work?

- We can express acceptance probability directly in terms of $\hat{p}(Y_{1:T}|\theta)$.
- Need to generate a proposed draw for both θ and $U_{1:T}$: ϑ and $U^*_{1:T}$.
- The proposal distribution for $(\vartheta, U^*_{1:T})$ in the MH algorithm is given by $q(\vartheta|\theta^{(i-1)})p(U^*_{1:T})$.
- No need to keep track of the draws $(U^*_{1:T})$.
- MH acceptance probability:

$$
\begin{aligned}
\alpha(\vartheta|\theta^{i-1}) &= \min \left\{ 1, \frac{\frac{g(Y|\vartheta, U^*)p(U^*)p(\vartheta)}{q(\vartheta|\theta^{(i-1)})p(U^*)}}{\frac{g(Y|\theta^{(i-1)}, U^{(i-1)})p(U^{(i-1)})p(\theta^{(i-1)})}{q(\theta^{(i-1)}|\theta^*)p(U^{(i-1)})}} \right\} \\
&= \min \left\{ 1, \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{(i-1)})}{\hat{p}(Y|\theta^{(i-1)})p(\theta^{(i-1)})/q(\theta^{(i-1)}|\vartheta)} \right\}.
\end{aligned}
$$

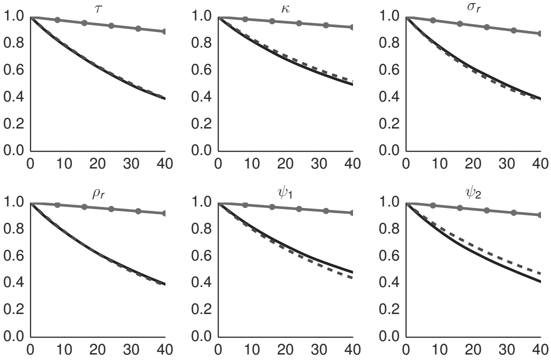
Small-Scale DSGE: Accuracy of MH Approximations

- Results are based on $N_{run} = 20$ runs of the PF-RWMH-V algorithm.
- Each run of the algorithm generates $N = 100,000$ draws and the first $N_0 = 50,000$ are discarded.
- The likelihood function is computed with the Kalman filter (KF), bootstrap particle filter (BS-PF, $M = 40,000$) or conditionally-optimal particle filter (CO-PF, $M = 400$).
- "Pooled" means that we are pooling the draws from the $N_{run} = 20$ runs to compute posterior statistics.

Small-Scale DSGE: Accuracy of MH Approximations

	Posterior Mean (Pooled)			Inefficiency Factors			Std Dev of Means		
	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF
τ	2.63	2.62	2.64	66.17	126.76	1360.22	0.020	0.028	0.091
κ	0.82	0.81	0.82	128.00	97.11	1887.37	0.007	0.006	0.026
ψ_1	1.88	1.88	1.87	113.46	159.53	749.22	0.011	0.013	0.029
ψ_2	0.64	0.64	0.63	61.28	56.10	681.85	0.011	0.010	0.036
ρ_r	0.75	0.75	0.75	108.46	134.01	1535.34	0.002	0.002	0.007
ρ_g	0.98	0.98	0.98	94.10	88.48	1613.77	0.001	0.001	0.002
ρ_z	0.88	0.88	0.88	124.24	118.74	1518.66	0.001	0.001	0.005
$r^{(A)}$	0.44	0.44	0.44	148.46	151.81	1115.74	0.016	0.016	0.044
$\pi^{(A)}$	3.32	3.33	3.32	152.08	141.62	1057.90	0.017	0.016	0.045
$\gamma^{(Q)}$	0.59	0.59	0.59	106.68	142.37	899.34	0.006	0.007	0.018
σ_r	0.24	0.24	0.24	35.21	179.15	1105.99	0.001	0.002	0.004
σ_g	0.68	0.68	0.67	98.22	64.18	1490.81	0.003	0.002	0.011
σ_z	0.32	0.32	0.32	84.77	61.55	575.90	0.001	0.001	0.003
$\ln \hat{p}(Y)$	-357.14	-357.17	-358.32				0.040	0.038	0.949

Autocorrelation of PFMH Draws



Notes: The figure depicts autocorrelation functions computed from the output of the 1 Block RWMH-V algorithm based on the Kalman filter (solid), the conditionally-optimal particle filter (dashed) and the bootstrap particle filter (solid with dots).

SW Model: Accuracy of MH Approximations

- Results are based on $N_{run} = 20$ runs of the PF-RWMH-V algorithm.
- Each run of the algorithm generates $N = 10,000$ draws.
- The likelihood function is computed with the Kalman filter (KF) or conditionally-optimal particle filter (CO-PF).
- "Pooled" means that we are pooling the draws from the $N_{run} = 20$ runs to compute posterior statistics. The CO-PF uses $M = 40,000$ particles to compute the likelihood.

SW Model: Accuracy of MH Approximations

	Post. Mean (Pooled)		Ineff. Factors		Std Dev of Means	
	KF	CO-PF	KF	CO-PF	KF	CO-PF
$(100\beta^{-1} - 1)$	0.14	0.14	172.58	3732.90	0.007	0.034
$\bar{\pi}$	0.73	0.74	185.99	4343.83	0.016	0.079
$\bar{l}$	0.51	0.37	174.39	3133.89	0.130	0.552
α	0.19	0.20	149.77	5244.47	0.003	0.015
σ_c	1.49	1.45	86.27	3557.81	0.013	0.086
Φ	1.47	1.45	134.34	4930.55	0.009	0.056
φ	5.34	5.35	138.54	3210.16	0.131	0.628
h	0.70	0.72	277.64	3058.26	0.008	0.027
ξ_w	0.75	0.75	343.89	2594.43	0.012	0.034
σ_l	2.28	2.31	162.09	4426.89	0.091	0.477
ξ_p	0.72	0.72	182.47	6777.88	0.008	0.051
ι_w	0.54	0.53	241.80	4984.35	0.016	0.073
ι_p	0.48	0.50	205.27	5487.34	0.015	0.078
ψ	0.45	0.44	248.15	3598.14	0.020	0.078
r_π	2.09	2.09	98.32	3302.07	0.020	0.116
ρ	0.80	0.80	241.63	4896.54	0.006	0.025
r_y	0.13	0.13	243.85	4755.65	0.005	0.023
$r_{\Delta y}$	0.21	0.21	101.94	5324.19	0.003	0.022

SW Model: Accuracy of MH Approximations

	Post. Mean (Pooled)		Ineff. Factors		Std Dev of Means	
	KF	CO-PF	KF	CO-PF	KF	CO-PF
ρ_a	0.96	0.96	153.46	1358.87	0.002	0.005
ρ_b	0.22	0.21	325.98	4468.10	0.018	0.068
ρ_g	0.97	0.97	57.08	2687.56	0.002	0.011
ρ_i	0.71	0.70	219.11	4735.33	0.009	0.044
ρ_r	0.54	0.54	194.73	4184.04	0.020	0.094
ρ_p	0.80	0.81	338.69	2527.79	0.022	0.061
ρ_w	0.94	0.94	135.83	4851.01	0.003	0.019
ρ_{ga}	0.41	0.37	196.38	5621.86	0.025	0.133
μ_p	0.66	0.66	300.29	3552.33	0.025	0.087
μ_w	0.82	0.81	218.43	5074.31	0.011	0.052
σ_a	0.34	0.34	128.00	5096.75	0.005	0.034
σ_b	0.24	0.24	186.13	3494.71	0.004	0.016
σ_g	0.51	0.49	208.14	2945.02	0.006	0.021
σ_i	0.43	0.44	115.42	6093.72	0.006	0.043
σ_r	0.14	0.14	193.37	3408.01	0.004	0.016
σ_p	0.13	0.13	194.22	4587.76	0.003	0.013
σ_w	0.22	0.22	211.80	2256.19	0.004	0.012
$\ln \hat{p}(Y)$	-964	-1018			0.298	9.139

Computational Considerations

- We implement the PFMH algorithm on a single machine, utilizing up to twelve cores.
- For the small-scale DSGE model it takes 30:20:33 [hh:mm:ss] hours to generate 100,000 parameter draws using the bootstrap PF with 40,000 particles. Under the conditionally-optimal filter we only use 400 particles, which reduces the run time to 00:39:20 minutes.
- For the SW model it took 05:14:20:00 [dd:hh:mm:ss] days to generate 10,000 draws using the conditionally-optimal PF with 40,000 particles.