Learning Static Parameters in Stochastic Processes

Bharath Ramsundar

December 14, 2012

1 Introduction

Consider a Markovian stochastic process X_T evolving (perhaps nonlinearly) over time variable T. We may observe this process only through observations Y_T. This process is parameterized by vector of parameters Θ, which is typically unknown and must be learned from the sequence $\{Y_T\}$. Learning the parameter Θ is critical to gaining understanding of this class of stochastic processes.

Nonlinear stochastic processes arise in a variety of situations. For example, understanding the time evolution of stock value through a standard stochastic volatility model requires the ability to track the nonlinear evolution of price X_T. Such problems are typically solved by Sequential Monte Carlo (SMC) Methods, such as particle filtering, but only when parameter vector Θ is provided to the Monte Carlo Algorithm. Standard algorithms such as particle filtering encounter difficulties when Θ is unknown.

Consequently, a large literature has developed focusing on algorithms to learn Θ in this setting. For example, researchers from statistics and econometrics have introduced the Particle MCMC Algorithm [1] which mixes Markov Chain Monte Carlo (MCMC) methods with Sequential Monte Carlo algorithms in order to learn parameters Θ. Unfortunately, such algorithms tend to be slow, and often require at least quadratic time complexity [3].

In this project, we introduce a new algorithm that learns parameter Θ while performing inference. Namely, we modify the Decayed MCMC Filtering algorithm [4] to learn static parameters. We then perform empirical analysis showing the robustness of this algorithm in handling static parameters and also prove preliminary correctness results.

Direction and guidance for this research were provided by Professor Stuart Russell of UC Berkeley.

2 Nonlinear State Space Model

The following toy nonlinear stochastic problem [2] is routinely used to evaluate algorithms for their ability to handle nonlinear time evolution.

$$
\begin{aligned}
X_n &= \frac{X_{n-1}}{2} + 25\frac{X_{n-1}}{1+X_{n-1}^2} + 8\cos(1.2n) + V_n \\
Y_n &= \frac{X_n^2}{20} + W_n
\end{aligned}
$$

$X_1 \sim \mathcal{N}(0,5)$, the V_n are IID drawn from $\mathcal{N}(0,\sigma_v^2)$, and the W_n are IID drawn from $\mathcal{N}(0,\sigma_w^2)$ ($\mathcal{N}(m,\sigma^2)$ denotes the Gaussian distribution of mean m and variance σ^2 and IID means independent and identically distributed). We set parameter vecter $\Theta = (\sigma_v, \sigma_w)$. Henceforth, we maintain the convention in figures that blue lines are observations, green lines are true states, and red lines are the products of inference.

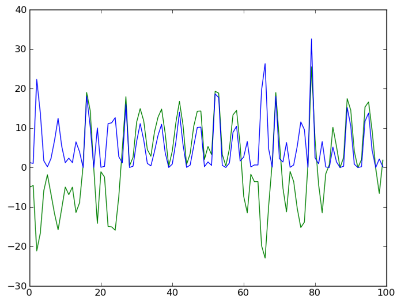

Figure 1: Nonlinear State Space Model, $\sigma_w = \sigma_v = 3$

3 Prior Literature on Static Parameter Learning

In this section we review various algorithms created for Static Parameter Learning Problems.

3.1 Particle Filtering

The Particle Filter does not learn Static Parameters, but effectively draws samples from nonlinear processes given such parameters. We implemented a simple particle filter to draw samples from the nonlinear state space model. Figure 2 shows that the particle filter can almost infer the true state.

3.2 Particle Markov Chain Monte Carlo

The PMCMC algorithm [1] is an extension of the Markov Chain Monte Carlo (MCMC) framework to handle nonlinearities. The PMCMC extends the reach of this framework by using a particle filter to sample from nonlinear distributions. We implemented PMCMC and used it to calculate distributions for parameters σ_v, σ_w. See Figure 3 for details.

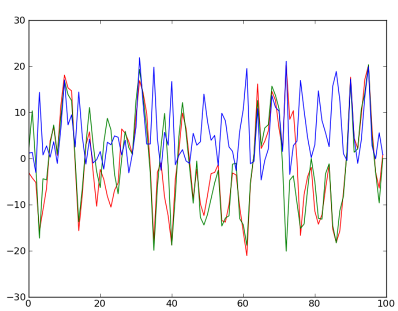

Figure 2: Nonlinear State Space Model Particle Filtering, 50 particles, 100 time steps

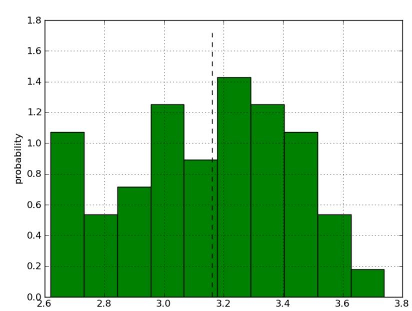

Figure 3: PMCMC Parameter Estimation Histogram

Algorithm 1: DMCMC: Decayed MCMC Filtering

Input: $g(t)$: Decay Function; S: Total Number of Time Steps;
K: Gibbs Moves Per Time Step;
Output: Approximate Filtering Distributions $\widetilde{\mathcal{D}}_s$

for $s = 1$ **to** S **do**
  for $i = 1$ **to** K **do**
    Choose t from $g(s)$;
    sample X_t from $P(X_t \mid X_{t-1}, X_{t+1}, Y_t)$;
    increment count for X_s in $\widetilde{\mathcal{D}}_s$;
return $\widetilde{\mathcal{D}}_1, \ldots, \widetilde{\mathcal{D}}_T$;

4 SDMCMC: Decayed MCMC with Static Parameters

In this section we define a variant of the Decayed MCMC Filtering algorithm (introduced in [4]) for the static parameter estimation problem.

4.1 Decayed MCMC Algorithm

We start by giving a brief overview of Decayed MCMC Filtering. Assume as before hidden state variables $X_1, \ldots, X_T$ and evidence variables $Y_1, \ldots, Y_T$. Decayed MCMC creates a markov chain to target the distribution $X_T \mid Y_1, \ldots Y_T$ through Gibbs updates. To save time, the algorithm spends progressively less time updating past elements and focuses on recent history. The rate of this decay is given by function $g(t)$ on window $[0, T]$ which controls sampling. For example $g(T) = \frac{1}{T}$ gives the standard Gibbs Sampling Methodology. The result achieved in [4] is that for

$$g_{\alpha,\delta}(t) = \alpha(T - t + 1)^{-(1+\delta)}$$

Algorithm 1 converges to the true filtering distribution for X_T in time not dependent on T.

4.2 Decayed MCMC for Static Parameters

MCMC techniques have long been used in the statistical literature to estimate static parameters. We consequently propose Algorithm (2) as a method to dynamically learn static parameters while filtering.

4.2.1 Empirical Results

Empirical results on the nonlinear state space model shows that Algorithm 2 works well in practice. However, the effectiveness of Algorithm 2 becomes most clear when we emphasize the fact that this inference is performed without prior knowledge of $\Theta = (\sigma_v, \sigma_w)$. For comparison, we perform inference in the Particle Filter with Θ initialized according to P_Θ. From Figures 4 and 5, we see that the particle filter noticeably diverges from the true state, while SDMCMC achieves near perfect accuracy.

In fact, with growing sequence size S, Algorithm 2 does not lose accuracy, while the particle filter does. Figure 6 compares the L^2 distances of the inferred and true solutions for SDMCMC and

Algorithm 2: SDMCMC: Decayed MCMC Filtering for Static Parameters

Input: $g(t)$: Decay Function; S: Total Number of Time Steps;
K: Gibbs Moves Per Time Step;
Output: Approximate Filtering Distributions $\widetilde{\mathcal{D}}_s$, parameter Θ

Sample Θ from prior P_Θ;
for $s = 1$ **to** S **do**
  for $i = 1$ **to** K **do**
    Choose t from $g(s)$;
    Sample u from $U([0, 1])$;
    if $u < \frac{1}{S}$ **then**
      Resample Θ from $P(\Theta \mid X_1, \ldots, X_s)$;
    else
      sample X_t from $P(X_t \mid X_{t-1}, X_{t+1}, Y_t, \Theta)$;
      increment count for X_s in $\widetilde{\mathcal{D}}_s$;
  Sample X_{s+1} from $P(X_{s+1} \mid X_s, Y_{s+1}, \Theta)$;
return $\widetilde{\mathcal{D}}_1, \ldots, \widetilde{\mathcal{D}}_T, \Theta$;

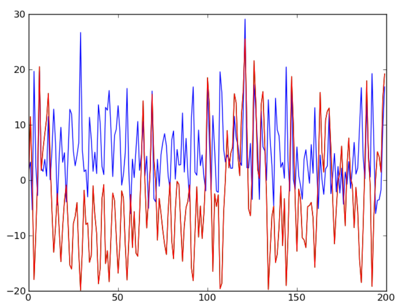

Figure 4: Decayed MCMC, $K = 3$

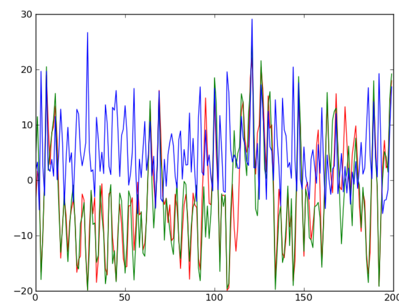

Figure 5: Particle Filter, 50 particles

SMC (Particle Filter) methods for $S = 200$ timesteps. This figure indicates that the L^2 distance is divergent for growing T with SMC methods but is convergent for SDMCMC.

Finally, we consider the parameter learning capabilities of SDMCMC. Figures 7 and 8 show the histograms of σ_v, σ_w considered in a run of Algorithm 2. Although the distribution does not center around the true parameters $\sigma_v = \sigma_w = 3$, it is close. The diagrams suggest that there might be some bias. Further analysis is required to clarify this point.

4.2.2 Preliminary Mathematical Correctness Analysis

We present some preliminary mathematical analysis of Algorithm 2.

Theorem 4.1. *For all $s < S$ in the outer **for** loop of Algorithm 2, the inner **for** loop defines a Markov process with stationary distribution $X_1, \ldots, X_s, \Theta \mid Y_1, \ldots, Y_s$.*

Proof. It suffices to show that the distribution $X_1, \ldots, X_s, \Theta \mid Y_1, \ldots, Y_s$ is invariant under an action of the inner **for** loop. To do so, we will consider the two cases of the **if** condition separately. Suppose $u < 1/S$. Note that the marginal distribution (summing out Θ) on $X_1, \ldots X_s$ is exactly

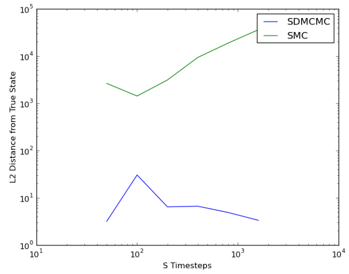

Figure 6: Log-Log Comparison of L^2 distance for SDMCMC and SMC

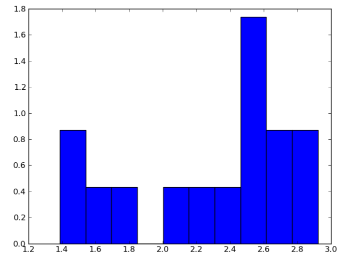

Figure 7: σ_v Histogram, True $\sigma_v = 3$

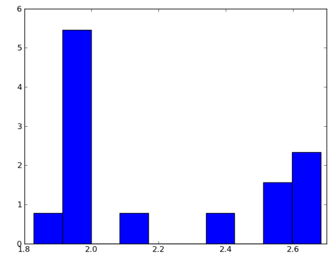

Figure 8: σ_w Histogram, True $\sigma_w = 3$

the conditional $X_1, \ldots, X_s \mid Y_1, \ldots, Y_s$. Algorithm 2 samples $\Theta \sim P(\Theta \mid X_1, \ldots, X_s)$. It follows that the joint distribution after the sample is

$$\begin{aligned} P(\Theta \mid X_1, \ldots, X_s) P(X_1, \ldots, X_s \mid Y_1, \ldots, Y_s) &= P(\Theta \mid X_1, \ldots, X_s, Y_1, \ldots, Y_s) P(X_1, \ldots, X_s \mid Y_1, \ldots, Y_s) \\ &= P(\Theta, X_1, \ldots, X_s \mid Y_1, \ldots, Y_s) \end{aligned}$$

Now suppose that $u \geq 1/S$. Suppose that i is sampled according to decay $g(s)$. Then the marginal distribution (summing out X_i) is

$$\Theta, X_1, \ldots, \hat{X}_i, \ldots, X_s \sim P(\Theta, X_1, \ldots, \hat{X}_i, \ldots, X_s \mid Y_1, \ldots, Y_s)$$

The hat signifies exclusion. Conditional independence shows that the joint distribution after the Gibbs sample is then

$$\begin{aligned} & P(X_i \mid \Theta, X_{i-1}, X_{i+1}, Y_i) P(\Theta, X_1, \ldots, \hat{X}_i, \ldots X_s \mid Y_1, \ldots, Y_s) \\ & \quad = P(X_i \mid \Theta, X_1, \ldots, \hat{X}_i, \ldots, X_s, Y_1, \ldots, Y_s) P(\Theta, X_1, \ldots, \hat{X}_i, \ldots, X_s \mid Y_1, \ldots, Y_s) \\ & \quad = P(\Theta, X_1, \ldots, X_s \mid Y_1, \ldots, Y_s) \end{aligned}$$

$\square$

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society*, 2010.

[2] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113. IET, 1993.

[3] N. Kantas. An overview of sequential monte carlo methods for parameter estimation in general state-space models. In *Proc. IFAC Symposium on System Identification (SYSID)*, 2009.

[4] B. Marthi, H. Pasula, S. Russell, and Y. Peres. Decayed mcmc filtering. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 319–326. Morgan Kaufmann Publishers Inc., 2002.