Sequential Stein Variational Gradient Descent for Time Series Model Estimation

Gibson, Reich, and Ray in some order

December 3, 2017

Introduction

Particle filtering suffers from two main practical disadvantages. The first is particle depletion, where the number of effective particles with non-neglibile weight becomes too small. This has the effect of concentrating the mass around a small number of particles, leading to poor estimates of the target distribution. The second is the running time of the algorithm. A cursory analysis reveals that each particle is updated once per time step in the time series, and once per re-sampling step, to mitigate the issue above. If we imagine the order of particles is close to the length of the time series, we see that run-time is $O(n^3)$.

We propose another approach that we hope will do better than particle filtering in practice. In this approach, Stein Variational Gradient Descent (SVGD) is used to sequentially estimate the distribution of state variables in each time step, conditional on observed data up through that time. This method should overcome problems with particle depletion and excessive run-times for long time-series.

Overview of SVGD

Stein Variational Gradient Descent can be used to estimate a continuous distribution by a set of samples. By iteratively transporting samples from an initial distribution in the direction of the likelihood, we are able to generate a sample from the target distribution. The usefullness of this approximation is apparent in Bayesian statistics, where the usually intractable normalizing constant disappears in the gradient. The particles are subject to the following gradient ascent procedure.

$$x_t^{(i)} \leftarrow x_{t-1}^{(i)} + \frac{1}{n}\sum_{j=1}^{n}[k(x_j, x_{t-1}^{(i)}) * \nabla log\ p(x_j) + \nabla k(x_j, , x_{t-1}^{(i)})]$$

Sequential Stein Variational Gradient Descent

Suppose we are given a time series $Y_1, Y_2, ..., Y_t$ for $Y \in \mathbb{R}$. We model the sequence as a state-space model parameterized by an observation density $p(y_t|x_t)$ and a transition density $p(x_t|x_{t-1})$ Figure 1.

We are interested in the filtering distribution $p(x_1, ..., x_n|y_1, ..., y_n)$ which by Bayes formula is

$$p(x_1, ..., x_n|y_1, ..., y_n) = \frac{p(y_1, ..., y_n|x_1, ..., x_n)p(x_1, ..., x_n)}{Z}$$

.

Because computing the normalizing constant Z is intractable for many choices of $p(y_t|x_t)$ and $p(x_t|x_{t-1})$, we must resort to monte carlo algorithms. The classic approach that incorporates the sequential nature of the data is given by the particle filtering algorithm. Particle filtering approximates the filtering density using sequential importance sampling. We instead focus on the following recursion.

$$p(x_t|y_{1:t}) = \int p(x_{0:t}|y_{1:t}) d_{x_0:t-1}$$

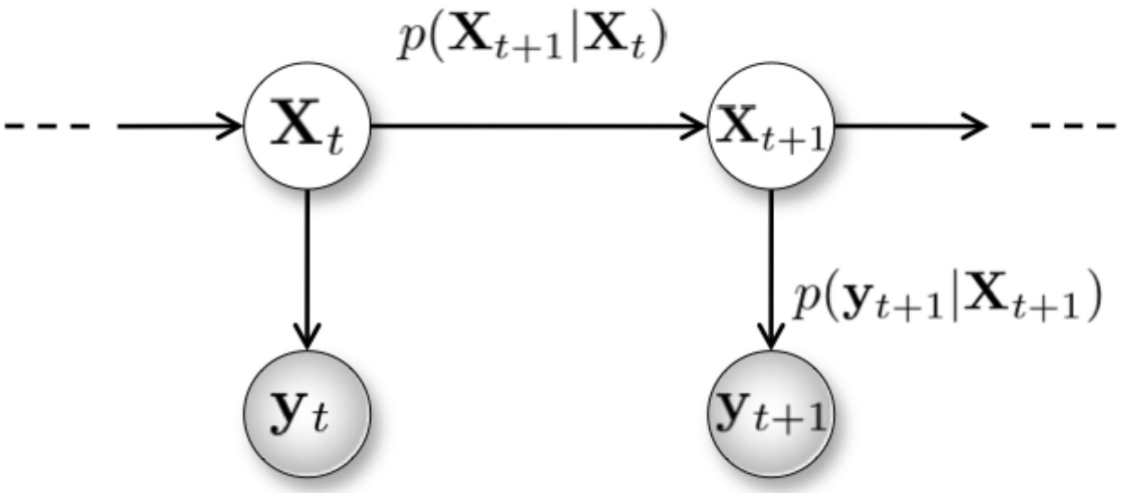
$p(\mathbf{X}_{t+1}|\mathbf{X}_t)$
$\mathbf{X}_t$
$\mathbf{X}_{t+1}$
$p(\mathbf{y}_{t+1}|\mathbf{X}_{t+1})$
$\mathbf{y}_t$
$\mathbf{y}_{t+1}$

Figure 1: Caption for the picture.

$$= \frac{p(y_t|x_t)}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t} p(x_t|y_{1:t-1})$$

$$\propto p(y_t|x_t)p(x_t|y_{1:t-1})$$
$$\propto p(y_t|x_t)p(x_t|y_{1:t-1})$$
$$\propto p(y_t|x_t) \int_{x_{t-1}} p(x_t, x_{t-1}|y_{1:t-1}) d_{x_{t-1}}$$

$$\propto p(y_t|x_t) \int_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}) d_{x_{t-1}}$$

which we can approximate recursively as

$$\propto p(y_t|x_t) \frac{1}{n} \sum_{i=1}^{n} p(x_t|x_{t-1}^{(i)})$$

(proof in apendix A)

Model Structure

States:

- $X_1 \sim g_1(x_1; \xi)$
- $X_t|X_{t-1} \sim g(x_t|x_{t-1}; \xi)$ for all $t = 2, \ldots, T$

Observations:

- $Y_t|X_t \sim h(y_t|x_t;\zeta)$

Here, $g_1(\cdot)$ and $g(\cdot)$ are appropriately defined probability density functions depending on parameters ξ and $h(\cdot)$ is an appropriately defined probability density function or probability mass function depending on parameters ζ.

Define $\theta = (\xi, \zeta)$ to be the full set of model parameters.

Filtering

There are two types of filtering:

1. sample of particles $x_{1:T}^{(k)} \sim f(x_{1:T}|y_{1:T})$
2. sample of particles $x_t^{(k)} \sim f(x_t|y_{1:t})$ for each $t = 1, \ldots, T$

Let's look at the second one. Assume we have a sample $x_{t-1}^{(k)} \sim f(x_{t-1}|y_{1:t-1})$

$$\begin{aligned}
p(x_t|y_{1:t}) &= \frac{f(x_t, y_t|y_{1:t-1})}{f(y_t|y_{1:t-1})} \\
&\propto f(x_t, y_t|y_{1:t-1}) \\
&= f(y_t|x_t) f(x_t|y_{1:t-1}) \\
&= f(y_t|x_t) \int f(x_t, x_{t-1}|y_{1:t-1}) dx_{t-1} \\
&= f(y_t|x_t) \int f(x_t|x_{t-1}) f(x_{t-1}|y_{1:t-1}) dx_{t-1} \\
&\approx f(y_t|x_t) \sum_{x_{t-1}^{(k)}} f(x_t|x_{t-1}^{(k)})
\end{aligned}$$

So $\log\{p(x_t|y_{1:t})\}$ is approximately proportional to $\log\{f(y_t|x_t)\} + \log\{\sum_{x_{t-1}^{(k)}} f(x_t|x_{t-1}^{(k)})\}$

Evaluating the Likelihood via Filtering

Our goal (for now) is to evaluate the likelihood function

$$\begin{aligned}
L(\theta|y_{1:T}) &= f(y_{1:T};\theta) \\
&= f(y_1;\theta) \prod_{t=2}^{T} f(y_t|y_{1:t-1};\theta) \\
&= \int_{x_1} f(y_1, x_1;\theta) dx_1 \prod_{t=2}^{T} \int_{x_t} f(y_t, x_t|y_{1:t-1};\zeta) dx_t \\
&= \int_{x_1} f(y_1|x_1;\zeta) f(x_1;\xi) dx_1 \prod_{t=2}^{T} \int_{x_t} f(y_t|x_t, y_{1:t-1};\zeta) f(x_t|y_{1:t-1};\xi) dx_t \\
&= \int_{x_1} f(y_1|x_1;\zeta) f(x_1;\xi) dx_1 \prod_{t=2}^{T} \int_{x_t} f(y_t|x_t;\zeta) f(x_t|y_{1:t-1};\xi) dx_t \\
&\approx \sum_{x_1^{(k)}} f(y_1|x_1^{(k)};\zeta) \prod_{t=2}^{T} \sum_{x_{t|t-1}^{(k)}} f(y_t|x_{t|t-1}^{(k)};\zeta), \text{ where}
\end{aligned}$$

$$x_1^{(k)} \sim f(x_1;\xi) \text{ and } x_{t|t-1}^{(k)} \sim f(x_t|y_{1:t-1};\xi)$$

Note that if we have a sample $x_{t-1|t-1}^{(k)} \sim f(x_{t-1}|y_{1:t-1};\xi)$, we can obtain a sample $x_{t|t-1}^{(k)} \sim f(x_t|y_{1:t-1};\xi)$ from the transition density.

We will apply SVGD to iteratively obtain samples from the updated distributions $x_{t|t}^{(k)} \sim f(x_t|y_{1:t};\xi)$ starting from samples $x_{t-1|t-1}^{(k)} \sim f(x_{t-1}|y_{1:t-1};\xi)$ at the previous time step. To do this, we need to obtain the derivative of the log of the density we want to estimate with respect to x_t.

$$\begin{aligned}
\frac{d}{dx_t}\log\{f(x_t|y_{1:t};\xi)\} &= \frac{d}{dx_t}\log\left\{\frac{f(x_t|y_{1:t-1})f(y_t|x_t,y_{1:t-1})}{f(y_t|y_{t:t-1})}\right\} \\
&= \frac{d}{dx_t}\left[\log\left\{f(x_t|y_{1:t-1})\right\} + \log\left\{f(y_t|x_t)\right\} - \log\left\{f(y_t|y_{t:t-1})\right\}\right] \\
&= \frac{d}{dx_t}\log\left\{\int_{x_{t-1}} f(x_t|x_{t-1},y_{1:t-1};\xi)f(x_{t-1}|y_{1:t-1};\xi)dx_{t-1}\right\} + \frac{d}{dx_t}\log\left\{f(y_t|x_t)\right\} \\
&= \frac{\frac{d}{dx_t}\int_{x_{t-1}} f(x_t|x_{t-1};\xi)f(x_{t-1}|y_{1:t-1};\xi)dx_{t-1}}{\int_{x_{t-1}} f(x_t|x_{t-1};\xi)f(x_{t-1}|y_{1:t-1};\xi)dx_{t-1}} + \frac{\frac{d}{dx_t}f(y_t|x_t)}{f(y_t|x_t)} \\
&\approx \frac{\frac{d}{dx_t}\sum_{x_{t-1|t-1}^{(k)}} f(x_t|x_{t-1};\xi)}{\sum_{x_{t-1|t-1}^{(k)}} f(x_t|x_{t-1};\xi)} + \frac{\frac{d}{dx_t}f(y_t|x_t)}{f(y_t|x_t)} \\
&= \frac{\sum_{x_{t-1|t-1}^{(k)}} \frac{d}{dx_t}f(x_t|x_{t-1};\xi)}{\sum_{x_{t-1|t-1}^{(k)}} f(x_t|x_{t-1};\xi)} + \frac{\frac{d}{dx_t}f(y_t|x_t)}{f(y_t|x_t)}
\end{aligned}$$

Simulation Studies

We will do several simulation studies, divided into 2 groups:

1. illustrating scenarios in which common particle filtering methods struggle, but SSVGD has better chances
   a. bad initalization
   b. normal-poisson filtering, seasonal model
   c. one other, more complex?
2. demonstrating accuracy (compare to true states, exactly computed filtered states, and likelihood)
   a. Kalman filter
   b. something nonlinear where we can do exact computations (brute force for short time series??)?

1. a. Bad Initialization

Simulation Study Design

Paragraph with data generating process, written with math.

Paragraph describing settings for simulation, e.g. number of simulation runs, length of time series generated, etc.

Paragraph describing different methods in comparison. 2 PF implementations, one non-linear KF implementation, and SSVGD

Results

Paragraph, referencing one figure and one table, summarizing results

1. b. Normal-Poisson seasonal model

Simulation Study Design

Paragraph with data generating process, written with math.

Paragraph describing settings for simulation, e.g. number of simulation runs, length of time series generated, etc.

Paragraph describing different methods in comparison. 2 PF implementations, one non-linear KF implementation, and SSVGD

Results

Paragraph, referencing one figure and one table, summarizing results

1. c. One other, more complex?

Simulation Study Design

Paragraph with data generating process, written with math.

Paragraph describing settings for simulation, e.g. number of simulation runs, length of time series generated, etc.

Paragraph describing different methods in comparison. 2 PF implementations, one non-linear KF implementation, and SSVGD

Results

Paragraph, referencing one figure and one table, summarizing results

2. a. Kalman Filter

Simulation Study Design

Paragraph with data generating process, written with math.

Paragraph describing settings for simulation, e.g. number of simulation runs, length of time series generated, etc.

Paragraph describing different methods in comparison. 2 PF implementations, one non-linear KF implementation, and SSVGD

Results

Paragraph, referencing one figure and one table, summarizing results

2. b. something nonlinear where we can do exact computations (brute force for short time series??)?

Simulation Study Design

Paragraph with data generating process, written with math.

Paragraph describing settings for simulation, e.g. number of simulation runs, length of time series generated, etc.

Paragraph describing different methods in comparison. 2 PF implementations, one non-linear KF implementation, and SSVGD

Results

Paragraph, referencing one figure and one table, summarizing results

Application

Example model with real data. fairly real model, but not thaaaaaat complex.

Discussion

Bibliography