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 limitations including:

- particle depletion
- predict steps can be far from filtered steps?
- high Monte Carlo variability

These can be addressed with some success by various strategies including whatever it's called when you add new particles near current particles if effective number of particles is too small.

Here we propose another approach that we hope will do better than particle filtering. 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 predictions that are far from the true states that come up with particle filtering.

Method Derivation

Let x_t, $t = 1, \ldots, T$ denote an unobserved state vector at each time t. For now, this state has to be continuous but we might be able to discretize later?

Let y_t, $t = 1, \ldots, T$ denote an observed value at each time t.

The details about when we start observing the y_t's relative to the first x_t are unimportant.

For now, we're just going to write down the method to evaluate the likelihood for a fixed set of parameters. This is not explicitly Bayesian or frequentist.

But we could likely differentiate the approximation to the likelihood derived here with respect to parameters θ and plug that into SVGD to estimate the posterior?

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.

Overview of SVGD

SVGD can be used to estimate a (continuous only?) distribution (as a mixture of normals? Is that right?). It requires as inputs a set of initial values for centers of the normals (? or are they just particles?) and a gradient of the density at a particular particle/center point.

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.

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

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