Seminar on Advanced Topics in Statistical Learning

Mou Minghao

SSE, CUHK(SZ)

July 27, 2022

Outline

EM Algorithm

K-Means Clustering

Gaussian Mixed Model

Hidden Markov Model

Kalman Filter

Particle Filter

Bayesian Inference for Latent Hawkes Process

Reparameterization Gradients through Acceptance-Rejection Sampling

Variational Sequential Monte Carlo

References

Expectation Maximization

Expectation Maximization Algorithm is an iterative algorithm which is used to calculate the maximum likelihood estimation or maximum a posterior in parametric probabilistic models with hidden variables.
It can be decomposed into two steps

1. Expectation: estimate the parameters based on the data and the model then calculate the expectation
2. Maximization: find the parameter which maximizes the likelihood function

EM Algorithm

Given the data $\mathcal{X}$, assume independence of samples, we want to fit the model $p(x;\theta)$, the log likelihood function is given as

$$
\begin{aligned}
L(\theta) &:= \sum_{i=1}^{n} \log p(x_i;\theta) \\
&= \sum_{i=1}^{n} \log \sum_{z} p(x_i, z;\theta) \\
&= \sum_{i=1}^{n} \log \sum_{z} Q_i(z) \frac{p(x_i, z;\theta)}{Q_i(z)} \\
&\geq \sum_{i=1}^{n} \sum_{z} Q_i(z) \log \frac{p(x_i, z;\theta)}{Q_i(z)} \\
&:= J(z, Q;\theta)
\end{aligned}
\tag{1}
$$

EM Algorithm

In the last inequality, we used Jensen's inequality.
Since the objective is to maximize the log likelihood and we notice that the last quantity in (1) gives a lower bound to the log likelihood. Hence, we try to maximize the lower bound.
In Jensen's inequality, the equality holds iff $X = \mathbb{E}(X)$, consider

$$\frac{p(x_i, z; \theta)}{Q_i(z)} = c \tag{2}$$

Sum both sides of 2 over z and use $\sum_z Q_i(z) = 1$, we have

$$\sum_z p(x_i, z; \theta) = c \tag{3}$$

Thus,

$$Q_i(z) = \frac{p(x_i, z; \theta)}{\sum_z p(x_i, z, \theta)} = p(z|x_i; \theta) \tag{4}$$

which is actually the *posterior distribution* of z.

EM Algorithm

Algorithm (Expectation Maximization)

```
Input: $\theta \leftarrow \theta_0$
For $j = 1, 2, ..., N$, do
    1. Expectation Step:
        1.1 $Q_i(z) \leftarrow p(z|x_i;\theta), \forall i$
        1.2 compute $J(z, Q; \theta) = \sum_{i=1}^{n} \sum_{z} Q_i(z) \frac{p(x_i, z; \theta)}{Q_i(z)}$
    2. Maximization Step:
        2.1 $\theta \leftarrow \text{arg}\max_\theta J(z, Q; \theta)$
```

Remark

Idea: *first fix θ, tune $Q(z)$ so that the lower bound $J(z, Q; \theta)$ equals $L(\theta)$. Fix $Q(z)$, find the θ which maximizes $J(z, Q; \theta) = L(\theta)$. Repeat this procedure until θ converges (i.e. $\|\theta_{i+1} - \theta_i\| < \epsilon$)*

Latent Hawkes Process[1]

While Poisson processes are foundational models for spatiotemporal data, many real-world systems violate the assumption of independent intervals. Hawkes processes remedy this shortcoming by allowing events to influence the future rate.

Definition (Conditional Intensity Function)

$$\lambda_v(t, y|\mathcal{H}_t, \theta) = b_v(t, y; \theta) + \sum_{n=1}^{N} f_{v_n \to v}(t, y; t_n, y_n; \theta)\mathbb{I}[t > t_n] \quad (5)$$

Given this conditional intensity function, the log likelihood of a set of events decomposes into two terms: the negative integrated rate and the sum of instantaneous log rates,

[1] The content is from [Linderman et al., 2017]

Latent Hawkes Process

$$\begin{aligned}&\log p(\{v_n, t_n, y_n\}\,|\theta) \\ &= -\sum_{v=1}^{V} \int_0^T \int_{\mathcal{Y}} \lambda_v(t, y|\mathcal{H}_t, \theta) dtdy + \sum_{n=1}^{N} \log \lambda_{v_n}(t_n, y_n|\mathcal{H}_{t_n}, \theta)\end{aligned} \tag{6}$$

in the partially-observed case, we must perform joint inference of both the model parameters and the latent data.

Learning and inference in latent Hawkes processes is fundamentally a latent variable problem. As such, we start with an *expectation-maximization* algorithm that alternates between

1. **Inference**: computing expected log likelihoods

$$\mathcal{L}(\theta) := \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\theta_{curr})}[\log p(\boldsymbol{z}, \boldsymbol{x}|\theta)]$$

2. **Learning**: taking gradients with respect to the model parameters $\nabla_\theta \mathcal{L}$

Data-driven Sequential Monte Carlo

The unique challenge of latent Hawkes processes is that the latent variables take the form of a marked point process. Specifically, since the number of latent events is undetermined, we propose a variety of methods for performing inference over sets of unknown cardinality.

It leverages the autoregressive nature of Hawkes processes (the instantaneous rate is only a function of preceding events) to sequentially propose and resample particles.

Data-driven Sequential Monte Carlo

1. define a scaffold $\{s_i\}_{i=1}^{I}$ that partitions the time range $[0, T]$ into I disjoint intervals
2. the value of the p-th particle in the i-th interval $z_i^{(p)} := \left\{(v_n^{(p)}, t_n^{(p)}, y_n^{(p)}) : s_{i-1} < t_n^{(p)} \leq s_i\right\}$ is a set of latent events. We only propose latent events for vertices whose data is missing in that interval
3. We generate a candidate set of latent events for interval i by sampling a proposal distribution $z_i^{(p)} \sim r(z_i | x_{1:i}, z_{1:i-1}^{(p)}, \theta)$, and weighting the newly updated particles with the function,

$$w(z_{1:i}^{(p)}) = \frac{p(x_{1:i}, z_{1:i}^{(p)} | \theta)}{p(x_{1:i-1}, z_{1:i-1}^{(p)} | \theta) r(z_i^{(p)} | x_{1:i}, z_{1:i-1}^{(p)}, \theta)} \quad (7)$$

   the high dimensionality of the marks calls for delicate choices of the proposal distribution to control the variance of the SMC estimates. To this end, we utilize data-driven proposals, leveraging our intuition that latent marks are often similar to observed marks.

Rao-Blackwellized Sequential Monte Carlo

note that in the un-marked case, inference of the latent times is relatively simple, and standard SMC works well. This motivates a Rao-Blackwellized approach, in which we marginalize the latent y_n and infer only the timestamps t_n and vertices v_n. The weights are then given by $w(\tilde{z}_{1:i}^{(p)})$, where $\tilde{z}_{1:i}^{(p)}$ denotes the particles without marks

the weights now need the marginal likelihood $p(x_{1:i}, \tilde{z}_{1:i}^{(p)}, \theta)$, we have

$$p(x_{1:i}, \tilde{z}_{1:i}^{(p)} | \theta) = \frac{p(x_{1:i}, z_{1:i}^{(p)}, \theta)}{p(y_{1:i}^{(p)} | x_{1:i}, \tilde{z}_{1:i}, \theta)} \tag{8}$$

However, we do not know $p(y_{1:i}^{(p)} | x_{1:i}, \tilde{z}_{1:i}, \theta)$. Thus, we perform VI on it. Specifically, we optimize a parametric variational distribution $q(y_{1:i}^{(p)}; \eta) \approx p(y_{1:i}^{(p)} | x_{1:i}, \tilde{z}_{1:i}, \theta)$, its ELBO is

$$ELBO(p, q) = \mathbb{E}_q[\log p(x_{1:i}, z_{1:i}^{(p)}, \theta)] - \mathbb{H}[q(y_{1:i}^{(p)}; \eta)] \tag{9}$$

Rao-Blackwellized Sequential Monte Carlo

We just maximize (9), it can be achieved by applying Coordinate Ascent or Gradient Ascent algorithms.

Remark

This approximation biases our SMC estimates, but the Rao-Blackwellization should reduce its variance. In other words, we trade bias in variational approximation for lower variance due to Rao-Blackwellization.

Data-driven Sequential Monte Carlo: Algorithm Summary

Algorithm (Variational Inference)

```
VariationalInference(p,q)
Input: target $p(y_{1:i}|x_{1:i}, \tilde{z}_{1:i}, \theta)$, variational family $q(y_{1:i}; \eta)$
Output: $q(y_{1:i}; \eta^*)$
$\eta^* = arg \max_\eta ELBO(p, q)$ as given in (9)
End
```

Data-driven Sequential Monte Carlo: Algorithm Summary

Algorithm (Data-driven Rao-Blackwellized SMC)

```
Input: model $p(x_{1:i}, z_{1:i}, \theta)$, variational family $q(y_{1:i}; \eta)$, proposal distribution $r(z_i|x_{1:i}, z_{1:i-1}^{(p)}, \theta)$, number of partitions $I$, number of particles $M$
Output: variation parameter $\eta^*$, model parameter $\theta^*$
$q(y_{1:i}, \eta^*) \leftarrow \textsc{VarationalInference}(p, q)$
$\theta_0 \leftarrow 0$
For $i = 1, 2, ..., N$ do
    For $p = 1, 2, ..., M$ do
  1. sample $z_1^{(p)} \sim r(z_1|x_1, \theta)$, $w(\tilde{z}_1^{(p)}) \leftarrow \frac{p(x_1, z_1^{(p)}, \theta)}{q(y_1^{(p)}; \eta^*) r(\tilde{z}_1^{(p)}|x_1, \theta)}$
    For $i = 2, 3, ..., I$ do
        For $p = 1, 2, ..., M$ do
  1. sample $z_i^{(p)} \sim r(z_i|x_{1:i}, z_{1:i-1}^{(p)}, \theta)$
  2. $w(\tilde{z}_{1:i}^{(p)}) \leftarrow \frac{p(x_{1:i}, z_{1:i}^{(p)}|\theta) q(y_{1:i-1}^{(p)}; \eta^*)}{p(x_{1:i-1}, z_{1:i-1}^{(p)}|\theta) q(y_{1:i}^{(p)}; \eta^*) r(z_i^{(p)}|x_{1:i}, \tilde{z}_{1:i-1}^{(p)}, \theta)}$
Base on the samples $\tilde{z}_{1:i}^{(p)}$, we can approximate $p(\tilde{z}_{1:i}|x_{1:i}, \theta)$
```

Data-driven Sequential Monte Carlo: Algorithm Summary

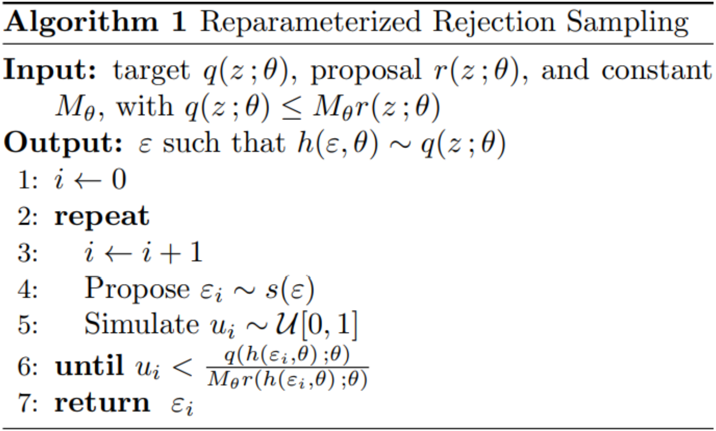
Algorithm (Data-driven Rao-Blackwellized SMC, ctd.)

```
    $\mathcal{L}(\theta) \leftarrow \mathbb{E}_{p(\tilde{z}_{1:i}|x_{1:i},\theta_{i-1})}[\log p(x_{1:i}, \tilde{z}_{1:i}, \theta)]$
    $\theta_i \leftarrow arg \max_\theta \mathcal{L}(\theta)$
    If $\theta$ converges do
        Output $\theta_i$ as $\theta^*$
End
Resampling and standardization
```

RSVI

- We can view the rejection sampler as a complicated deterministic mapping of a (random) number of simple random variables such as uniforms and normals
- However, this mapping is in general not differentiable
- Our insight is that we can overcome this problem by instead considering only the marginal over the accepted sample, analytically integrating out the accept-reject variable
- This is continuous under mild assumptions, enabling us to greatly extend the class of variational families amenable to reparameterization

Reparametrized Rejection Sampling



Algorithm 1 Reparameterized Rejection Sampling

```
Input: target $q(z\,;\theta)$, proposal $r(z\,;\theta)$, and constant
    $M_\theta$, with $q(z\,;\theta) \leq M_\theta r(z\,;\theta)$
Output: $\varepsilon$ such that $h(\varepsilon, \theta) \sim q(z\,;\theta)$
$i \leftarrow 0$
repeat
    $i \leftarrow i + 1$
    Propose $\varepsilon_i \sim s(\varepsilon)$
    Simulate $u_i \sim \mathcal{U}[0, 1]$
until $u_i < \frac{q(h(\varepsilon_i, \theta)\,;\theta)}{M_\theta r(h(\varepsilon_i, \theta)\,;\theta)}$
return $\varepsilon_i$
```

The RRS in VI

We now use reparameterized rejection sampling to develop a novel Monte Carlo estimator of the gradient of the ELBO $\mathcal{L}(\theta) := \mathbb{E}_q[\log p(x, z)] + \mathbb{H}[q(z)]$

Proposition

Let $\pi(\epsilon; \theta)$ be the distribution of the accepted samples in the RRS sampling, we have

$$\pi(\epsilon; \theta) = s(\epsilon) \frac{q(h(\epsilon, \theta); \theta)}{r(h(\epsilon, \theta); \theta)} \tag{10}$$

proof.

□

RSVI

Proposition

Let f be any measurable function and $\epsilon \sim \pi(\epsilon, \theta)$, then

$$\mathbb{E}_{\pi}[f(h(\epsilon, \theta))] = \mathbb{E}_{q}[f(z)] \tag{11}$$

proof.

□

RSVI

By the above proposition, we have ELBO

$$\mathcal{L}(\theta) = \mathbb{E}_{\pi}[f(h(\epsilon, \theta))] + \mathbb{H}[q(z; \theta)] \tag{12}$$

compute the gradient of $\mathbb{E}_q[f(z)]$

$$\nabla_\theta \mathbb{E}_q[f(z)] = \mathbb{E}_{\pi}[\nabla_\theta f(h(\epsilon, \theta))] + \mathbb{E}_{\pi}[f(h(\epsilon, \theta))\nabla_\theta \log \frac{q(h(\epsilon, \theta))}{r(h(\epsilon, \theta), \theta)}] \tag{13}$$

the first term on RHS is g_{rep}, the second term is g_{cor}. Thus,

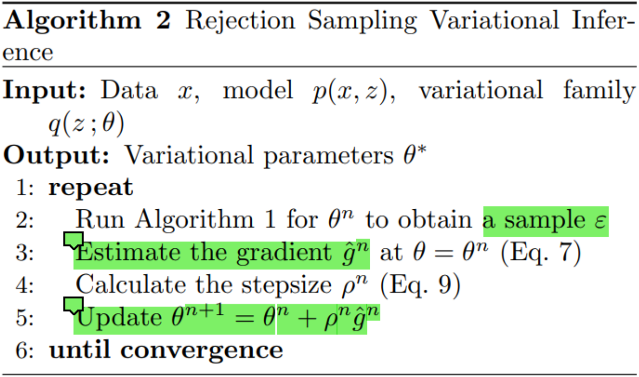
$$\nabla_\theta \mathcal{L}(\theta) = g_{rep} + g_{cor} + \mathbb{H}[q(z; \theta)] \tag{14}$$

and thus we can build an unbiased one-sample Monte Carlo estimator

$$\begin{aligned} \hat{g}_{rep} &= \nabla_z f(z)|_{z=h(\epsilon,\theta)} \nabla_\theta h(\epsilon, \theta) \\ \hat{g}_{cor} &= f(h(\epsilon, \theta))\nabla_\theta \log \frac{q(h(\epsilon, \theta), \theta)}{r(h(\epsilon, \theta), \theta)} \end{aligned} \tag{15}$$

RSVI-Full Algorithm

use (14) to obtain a monte-carlo estimate of the gradient of ELBO and then use this estimate $\hat{g}$ to do SGD, the step size ρ^n is adopted from [Kucukelbir et al., 2015]



Algorithm 2 Rejection Sampling Variational Inference

```
Input: Data $x$, model $p(x, z)$, variational family
    $q(z\,;\theta)$
Output: Variational parameters $\theta^*$
repeat
    Run Algorithm 1 for $\theta^n$ to obtain a sample $\varepsilon$
    Estimate the gradient $\hat{g}^n$ at $\theta = \theta^n$ (Eq. 7)
    Calculate the stepsize $\rho^n$ (Eq. 9)
    Update $\theta^{n+1} = \theta^n + \rho^n \hat{g}^n$
until convergence
```

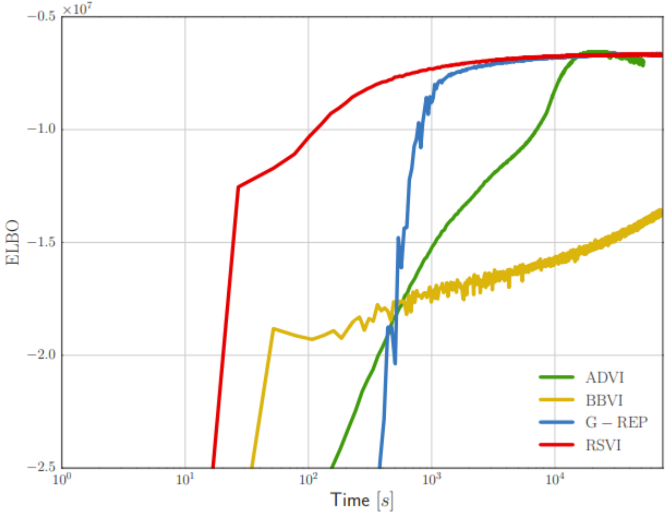
Examples of Acceptance-Rejection Reparameterization

- for *Gamma*$(\alpha, 1)$ with $\alpha \geq 1$, we can use

$$z = h(\epsilon, \alpha) := (\alpha - 1/3)(1 + \epsilon/\sqrt{9\alpha - 3})^3, \tag{16}$$

  where $\epsilon \sim s(\epsilon) := \mathcal{N}(0, 1)$
- when $\beta \neq 1$, divide z by β
- the acceptance rate of this method is very high (around 0.95 when $\alpha = 1$) and as $\alpha \to \infty$, the acceptance rate approaches 1 (i.e. $\pi(\epsilon, \theta) \to s(\epsilon)$)
- for $\alpha < 1$, note that $z = u^{1/\alpha}\tilde{z} \sim Gamma(\alpha, \beta)$ and $\tilde{z} \sim Gamma(\alpha + 1, \beta)$, we just do rejection sampling for $\tilde{z}$
- **Shape Augmentation:** inspired by the trick above, if we want to compute ELBO of *Gamma*$(\alpha, 1)$, we can consider $z = \tilde{z}\prod_{i=1}^{B} u_i^{1/(\alpha+i-1)}$, then $\tilde{z} \sim Gamma(\alpha + B, 1)$ [can be proved by induction], where $u_i \sim_{i.i.d.} \mathcal{U}[0, 1]$. Thus, we do rejection sampling on $\tilde{z}$ to leverage the high acceptance rate

Experiments



Variational Inference

Remark (Key Idea)

transfer an inference problem to an optimization problem

1. probabilistic model $p(x, z)$
2. want to find $p(z|x) = p(x, z)/p(x)$ (inference); analytically intractable → approximate inference
3. approximate $p(z|x)$ by $\{q(z; \theta)\} \rightarrow \min_\theta KL(q(z; \theta)||p(z|x))$, which is equivalent to $\max_\theta \mathbb{E}_q[\log p(x, z)] + \mathbb{H}[q(z; \theta)]$ (ELBO)
4. when the model is conjugate (the complete conditionals have conjugacy property) → coordinate ascent
5. Otherwise, do gradient ascent, the gradient involves complex integral → Monte-carlo method

Variational Inference

Definition (Score Function Estimator / log-derivative trick[2])

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{q(z;\theta)}[\log p(x,z) \nabla_\theta \log q(z;\theta)] + \nabla_\theta \mathbb{H}[q(z;\theta)] \tag{17}$$

Definition (Reparametrization trick[3])

Require: z is continuous; $h(\epsilon, \theta) \sim q(z;\theta)$, where $\epsilon \sim s(\epsilon)$ (independent of the variational parameter θ); h is differentiable w.r.t. θ

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{s(\epsilon)}[\nabla_z \log p(x, h(\epsilon,\theta)) \nabla_\theta h(\epsilon,\theta)] + \nabla_\theta \mathbb{H}[q(z;\theta)] \tag{18}$$

[2] [Williams, 1992]

[3] [Kingma and Welling, 2013]

Sequential Monte Carlo

1. SMC is a sampling method designed to approximate $p(x_{1:t}|y_{1:t}), t = 1, 2, ..., T$
2. $p(x_{1:t}|y_{1:t}) \approx \hat{p}(x_{1:t}|y_{1:t}) := \sum_{i=1}^{N} \frac{w_t^i}{\sum_l w_t^l} \delta_{x_{1:t}^l}$
3. at $t = 1$, standard importance sampling $x_1^i \sim r(x_1)$
4. for each step $t > 1$, start each step by resampling ancestor variables $a_{t-1}^i \in \{1, 2, ..., N\}$ with probability proportional to w_{t-1}^j
5. next propose new values and then append them to the trajectories

Algorithm (SMC)

1. *initialization*
2. $a_{t-1}^i \sim \mathit{Categorical}(\boldsymbol{w}_{t-1} / \sum_l w_{t-1}^l)$
3. $x_t^i \sim r(x_t | x_{t-1}^{a_{t-1}^i})$
4. $x_{1:t}^i := (x_{1:t-1}^{a_{t-1}^i}, x_t^i)$
5. $w_t^i = f(x_t^i | x_{t-1}^{a_{t-1}^i}) g(y_t | x_t^i) / r(x_t^i | x_{t-1}^{a_{t-1}^i})$

VSMC[4]

1. $p(x_{1:t}, y_{1:t})$: a sequence of probablistic models
2. want to compute $p(x_{1:t}|y_{1:t})$
3. in examples like HMM and SSM, $p(x_{1:T}, y_{1:t}) = f(x_1)\prod_{t=2}^{T} f(x_t|x_t - 1)\prod_{t=1}^{T} g(y_t|x_t)$, f is the prior on x and g is the observation distribution
4. Usually, the computation is intractable, we might apply VI or SMC. [Naesseth et al., 2018] combine these two ideas
5. This paper focus on SSM, but VSMC can be applied to any sequence of probabilistic models like SMC

[4] Naesseth, Christian, et al. "Variational sequential monte carlo." International conference on artificial intelligence and statistics. PMLR, 2018.

VSMC

Remark

SMC also yields an unbiased estimate of the marginal likelihood,

$$\hat{p}(y_{1:T}) = \prod_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} w_t^i \tag{19}$$

This estimate will play an important role in the VSMC objective.

VSMC

The proposal distribution $r(x_t|x_{t-1})$ is the key design choice

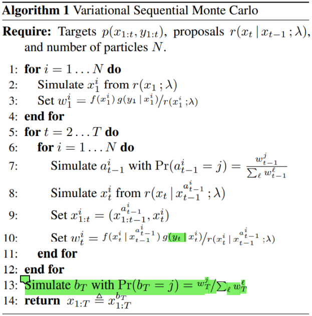
1. if $r = f$, this is known as bootstrap particle filter
2. poor approximation if the number of particles is small and x_t is high-dimensional
3. Variational SMC addresses this shortcoming; it learns parameterized proposal distributions for efficient inference

Remark (Pipeline of this paper)

1. *how to sample from the VSMC family*
2. *the distribution of this family*
3. *a tractable objective for optimization*
4. *learn model parameter by variational EM*

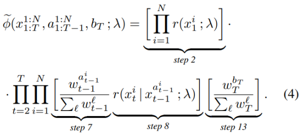
VSMC

- to sample from the VSMC family, we run SMC (the proposal $r(x_t|x_{t-1};\lambda)$ is parametrized by the variational parameter λ) and then sample once from the empirical approximation



Algorithm 1 Variational Sequential Monte Carlo

```
Require: Targets $p(x_{1:t}, y_{1:t})$, proposals $r(x_t \,|\, x_{t-1}\,;\lambda)$,
    and number of particles $N$.
 1: for $i = 1 \ldots N$ do
 2:   Simulate $x_1^i$ from $r(x_1\,;\lambda)$
 3:   Set $w_1^i = f(x_1^i)\, g(y_1 \,|\, x_1^i) / r(x_1^i\,;\lambda)$
 4: end for
 5: for $t = 2 \ldots T$ do
 6:   for $i = 1 \ldots N$ do
 7:     Simulate $a_{t-1}^i$ with $\Pr(a_{t-1}^i = j) = \frac{w_{t-1}^j}{\sum_\ell w_{t-1}^\ell}$
 8:     Simulate $x_t^i$ from $r(x_t \,|\, x_{t-1}^{a_{t-1}^i}\,;\lambda)$
 9:     Set $x_{1:t}^i = (x_{1:t-1}^{a_{t-1}^i}, x_t^i)$
10:     Set $w_t^i = f(x_t^i \,|\, x_{t-1}^{a_{t-1}^i})\, g(y_t \,|\, x_t^i) / r(x_t^i \,|\, x_{t-1}^{a_{t-1}^i}\,;\lambda)$
11:   end for
12: end for
13: Simulate $b_T$ with $\Pr(b_T = j) = w_T^j / \sum_\ell w_T^\ell$
14: return $x_{1:T} \triangleq x_{1:T}^{b_T}$
```



$$\widetilde{\phi}(x_{1:T}^{1:N}, a_{1:T-1}^{1:N}, b_T\,;\lambda) = \underbrace{\left[\prod_{i=1}^{N} r(x_1^i\,;\lambda)\right]}_{\textit{step 2}} \cdot$$

$$\cdot \prod_{t=2}^{T} \prod_{i=1}^{N} \underbrace{\left[\frac{w_{t-1}^{a_{t-1}^i}}{\sum_\ell w_{t-1}^\ell}\right.}_{\textit{step 7}} \underbrace{\left. r(x_t^i \,|\, x_{t-1}^{a_{t-1}^i}\,;\lambda)\right]}_{\textit{step 8}} \underbrace{\left[\frac{w_T^{b_T}}{\sum_\ell w_T^\ell}\right]}_{\textit{step 13}} . \qquad (4)$$

VSMC

- In this joint, the final output sample is defined by extracting the b_T-th trajectory $x_{1:T} = x_{1:T}^{b_T}$
- Note that the data $y_{1:T}$ enter via the weights and (optionally) the proposal distribution
- Let $b_t := a_t^{b_{t+1}}$ for $t \leq T - 1$ denote the ancestors for the trajectory $x_{1:T}$ returned by algorithm 1
- let $\neg b_{1:T}$ be all particle indices not equal to $(b_1, b_2, ..., b_T)$

$$\widetilde{\phi}(x_{1:T}^{1:N}, a_{1:T-1}^{1:N}, b_T\,; \lambda) = \underbrace{\left[\prod_{i=1}^{N} r(x_1^i\,; \lambda)\right]}_{\textit{step 2}} \cdot \\ \cdot \prod_{t=2}^{T}\prod_{i=1}^{N} \underbrace{\left[\frac{w_{t-1}^{a_{t-1}^i}}{\sum_\ell w_{t-1}^\ell}\right.}_{\textit{step 7}} \underbrace{\left. r(x_t^i \,|\, x_{t-1}^{a_{t-1}^i}\,; \lambda)\right]}_{\textit{step 8}} \underbrace{\left[\frac{w_T^{b_T}}{\sum_\ell w_T^\ell}\right]}_{\textit{step 13}}. \qquad (4)$$

VSMC

Proposition

The VSMC approximation on $x_{1:T}$ is

$$q(x_{1:T}|y_{1:T};\lambda) = p(x_{1:T}, y_{1:T})\mathbb{E}_{\tilde{\Phi}(x_{1:T}^{\neg b_{1:T}}, a_{1:T-1}^{\neg b_{1:T-1}};\lambda)}[\hat{p}(y_{1:T})^{-1}] \tag{20}$$

proof.

□

VSMC

- While we can estimate the expectation in (20) with Monte Carlo, it yields a biased estimate of $\log q(x_{1:T}|y_{1:T};\lambda)$ and ELBO
- $q(x_{1:T}|y_{1:T};\lambda)$ is hard to evaluate pointwisely, so we cannot directly maximize the ELBO
- to derive a tractable objective, a lower bound to the ELBO that is also amenable to stochastic optimization is proposed

Definition (Surrogate ELBO)

$$\tilde{\mathcal{L}}(\lambda) := \mathbb{E}_{\tilde{\Phi}(x_{1:T}^{1:N}, a_{1:T-1}^{1:N};\lambda)}\left[\log \hat{p}(y_{1:T})\right] \tag{21}$$

it satisfies

$$\log p(y_{1:T}) \geq \mathcal{L}(\lambda) \geq \tilde{\mathcal{L}}(\lambda) \tag{22}$$

proof.

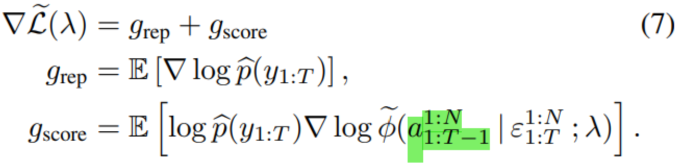
□

VSMC

- The surrogate ELBO is the expected SMC log-marginal likelihood estimate. We can estimate it unbiasedly as a byproduct of sampling from the VSMC variational approximation
- We run the algorithm and use the estimate to perform stochastic optimization of the surrogate ELBO

SGD

- assume the proposals $r(x_t|x_{t-1};\lambda)$ are reparameterizable, $x_t \sim h(x_{t-1}, \epsilon_t; \lambda), \epsilon_t \sim s(\epsilon_t)$
- With this assumption, rewrite the gradient of $\mathcal{L}(\lambda)$ by using the reparameterization trick[5]

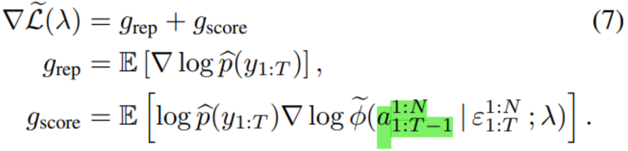


$$\begin{aligned} \nabla \widetilde{\mathcal{L}}(\lambda) &= g_{\text{rep}} + g_{\text{score}} \qquad (7) \\ g_{\text{rep}} &= \mathbb{E}\left[\nabla \log \widehat{p}(y_{1:T})\right], \\ g_{\text{score}} &= \mathbb{E}\left[\log \widehat{p}(y_{1:T}) \nabla \log \widetilde{\phi}(a^{1:N}_{1:T-1} \,|\, \varepsilon^{1:N}_{1:T}\,; \lambda)\right]. \end{aligned}$$

- The ancestor variables a are discrete and cannot be reparameterized—this can lead to high variance in the score function term

[5] [Kingma and Welling, 2013]

SGD

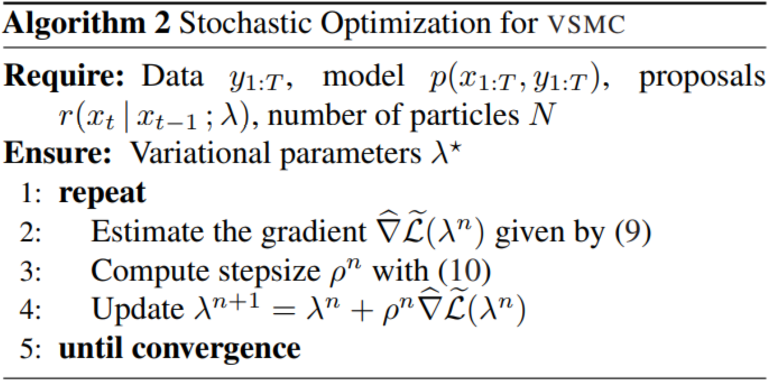


$$
\begin{aligned}
\nabla \widetilde{\mathcal{L}}(\lambda) &= g_{\text{rep}} + g_{\text{score}} \qquad (7)\\
g_{\text{rep}} &= \mathbb{E}\left[\nabla \log \widehat{p}(y_{1:T})\right],\\
g_{\text{score}} &= \mathbb{E}\left[\log \widehat{p}(y_{1:T}) \nabla \log \widetilde{\phi}(a_{1:T-1}^{1:N} \mid \varepsilon_{1:T}^{1:N}\, ; \lambda)\right].
\end{aligned}
$$

- We found that ignoring the score function term g_{score} leads to faster convergence and very little difference in final ELBO. This corresponds to approximating the gradient of $\tilde{\mathcal{L}}(\lambda)$ by

$$\nabla \tilde{\mathcal{L}}(\theta) \approx \mathbb{E}[\nabla \log \hat{p}(y_{1:T})] = g_{rep} \qquad (23)$$

Full Algorithm



Algorithm 2 Stochastic Optimization for VSMC

```
Require: Data $y_{1:T}$, model $p(x_{1:T}, y_{1:T})$, proposals $r(x_t \mid x_{t-1}\,; \lambda)$, number of particles $N$
Ensure: Variational parameters $\lambda^\star$
repeat
    Estimate the gradient $\widehat{\nabla}\widetilde{\mathcal{L}}(\lambda^n)$ given by (9)
    Compute stepsize $\rho^n$ with (10)
    Update $\lambda^{n+1} = \lambda^n + \rho^n \widehat{\nabla}\widetilde{\mathcal{L}}(\lambda^n)$
until convergence
```

Variation Expectation Maximization

- add a outer loop to do EM
- initially, randomly assign a θ_0, then it reduces to apply the algorithm proposed by this paper to get the posterior
- apply EM to update θ
- repeat

Perspectives on VSMC

- $N = 1 \rightarrow$ structured variational approximation, no resampling, the variational distribution is exactly the proposal
- $T = 1 \rightarrow$ importance sampling, the importance is the weight $w^i = f(x^i)g(y|x^i)/r(x^i;\lambda)$

Theoretic Results

The normalization constant estimate of the SMC sampler $\hat{p}(y_{1:T})$ is unbiased[6]. Together with Jensen's inequality, it implies that the surrogate ELBO $\tilde{\mathcal{L}}(\lambda)$ is a lower bound of $\log p(y_{1:T})$

$$\tilde{\mathcal{L}}(\lambda) = \mathbb{E}[\log \hat{p}(y_{1:T})] \leq \log \mathbb{E}[\hat{p}(y_{1:T})] = \log p(y_{1:T}) \tag{24}$$

Theorem ([Moral and Formulae, 2004])

If $\log \hat{p}(y_{1:T}) := \log \prod_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} w_t^i$ *is uniformly intergrable, then as* $N \to \infty$

$$\tilde{\mathcal{L}}(\lambda) = \mathcal{L}(\lambda) = \log p(y_{1:T}) \tag{25}$$

Remark

This fact means that the gap disappears and the distribution returned by VSMC $q(x_{1:T};\lambda)$ *will tend to the true posterior* $p(x_{1:T}|y_{1:T})$*. A bound is*

$$KL(q(x_{1:T};\lambda)||p(x_{1:T}|y_{1:T})) \leq \frac{c(\lambda)}{N} \tag{26}$$

where $c(\lambda)$ *is a constant depends on* λ

[6] [Moral and Formulae, 2004]

Theoretric Results

[Bérard et al., 2014] show a central limit theorem for the SMC approximation $\log \hat{p}(y_{1:T}) - \log p(y_{1:T})$ with $N = bT$, where $b > 0$ as $T \to \infty$

Theorem

Under the same condition as in the work [Bérard et al., 2014], assume $\log \hat{p}(y_{1:T})$ *is uniformly integrable, then*

$$KL(q(x_{1:T};\lambda)||p(x_{1:T}|y_{1:T})) \leq -\mathbb{E}\left[\log \frac{\hat{p}(y_{1:T})}{p(y_{1:T})}\right] \xrightarrow{T\to\infty} \frac{\sigma^2(\lambda)}{2b}, \quad (27)$$

where $0 < \sigma^2(\lambda) < \infty$

Remark

This implies that we can make the variational approximation arbitrarily accurate by setting $N \propto T$*, even as* $T \to \infty$

Experiments

stochastic volatility model[7] is a common model in financial econometrics

$$\begin{aligned} x_t =& \mu + \phi(x_{t-1} - \mu) + v_t, \\ y_t =& \beta \exp \frac{x_t}{2} e_t \end{aligned} \tag{28}$$

where $v_t \sim \mathcal{N}(0, Q), e_t \sim \mathcal{N}(0, I), x_1 \sim \mathcal{N}(\mu, Q)$ and $\theta = (\mu, \phi, Q, \beta)$

- computing $\log p(y_{1:T}; \theta)$ and its gradient is intractable
- we use *VEM* to learn θ
- the proposal chosen is $r(x_t|x_{t-1}; \lambda, \theta) \propto f(x_t|x_{t-1}; \theta)\mathcal{N}(x_t; y_t; \Sigma_t)$

Data:10 years of monthly returns (9/2007 to 8/2017) for the exchange rate of 22 international currencies w.r.t US dollars

[7] [Asai et al., 2006]

Experiments

Table 2: ELBO for the stochastic volatility model with $T = 119$ on exchange rate data. We compare VSMC (this paper) with IWAE and structured VI.

	Method	ELBO
	Structured VI	6905.1
$N = 4$	IWAE	6911.2
	VSMC	**6921.6**
$N = 8$	IWAE	6912.4
	VSMC	**6935.8**
$N = 16$	IWAE	6913.3
	VSMC	**6936.6**

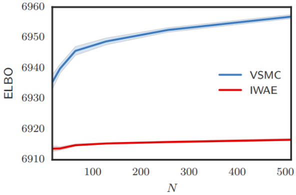


Figure 4: The estimated ELBO for VSMC (this paper) and IWAE , with confidence bands, as a function of the number of particles N for fix $\theta^\star$, $\lambda^\star$.

Experiments

Deep Markov Model[8]:

$$\begin{aligned} x_t =& \mu_\theta(x_{t-1}) + \exp(\sigma_\theta(x_{t-1})/2)v_t \\ y_t \sim& Poisson(\exp(\eta_\theta(x_t))) \end{aligned} \tag{29}$$

where $v_t \sim \mathcal{N}(0, I), x_0 = 0$ and μ, σ, η are NN parametrized by θ

The proposal is used:

$$r(x_t|x_{t-1}, y_t; \lambda) \propto \mathcal{N}(x_t; \mu^x_\lambda(x_{t-1}), \exp(\sigma^x_\lambda(x_{t-1}))) \times \mathcal{N}(x_t; \mu^y_\lambda(y_t), \exp(\sigma^y_\lambda(y_t))) \tag{30}$$

where $\mu^x, \mu^y, \sigma^x, \sigma^y$ are NNs parametrized by λ

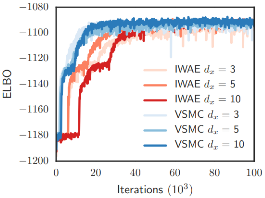


Figure 5: The estimated ELBO of the neural population test data as a function of iterations for VSMC (this paper) and IWAE, for $d_x = \{3, 5, 10\}$ and $T = 21$.

[8] [Dinh et al., 2014]

Dirac Measure δ

Definition

Let X be a nonempty set. Let $\mathscr{P}(X)$ denote the power set of X. Then $(X, \mathscr{P}(X))$ is a measurable space. Let $x \in X$, then Dirac Measure concentrated at x is $\delta_x : \mathscr{P}(X) \to \{0, 1\}$ defined by

$$\delta_x(E) = \begin{cases} 1 & \textit{if } x \in E \\ 0 & \textit{if } x \notin E \end{cases} \tag{31}$$

Uniform Integrability[9]

Two of the most important modes of convergence in probability theory are convergence a.s. and convergence in norm. neither mode of convergence implies the other. However, if we impose an additional condition on the sequence of variables, convergence a.s. will imply convergence in norm.

As usual, our starting point is a random experiment modeled by a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, Ω is the set of outcomes, $\mathscr{F}$ is the σ-algebra of events, and $\mathbb{P}$ is the probability measure. Recall that for $k \geq 1$, $L^k(\Omega)$ is the vector space of random variables with $\mathbb{E}[|X|^k] < \infty$, endowed with the norm $\|X\|_k := [\mathbb{E}[|X|^k]]^{1/k}$.

Remark

$$\mathbb{E}[X; A] := \mathbb{E}[X\mathbb{I}_A] = \int_A X d\mathbb{P} \tag{32}$$

[9] https://www.randomservices.org/random/expect/Uniform.html

Uniform Integrability

Definition (Uniform Integrability)

A sequence of RVs $\{X_i | i \in I\}$ *is said to be uniformly integrable if for* $\forall \epsilon > 0$, $\exists \delta > 0$ *such that for* $\forall i \in I$,

$$\mathbb{E}\left[|X_i|; |X_i| > \delta\right] < \epsilon \tag{33}$$

Equivalently, $\mathbb{E}[|X_i|; |X_i| > \delta] \to 0$ *as* $x \to \infty$ *uniformly in* $i \in I$

Proposition

The collection $\{X_i\}_I$ *is uniformly integrable if and only if the following conditions hold:*

1. $\{\mathbb{E}[|X_i|]\}$ *is uniformly bounded*
2. $\forall \epsilon > 0$, $\exists \delta > 0$ *such that if* $A \in \mathscr{F}$ *and* $\mathbb{P}(A) < \delta$, *then* $\mathbb{E}[|X_i|; A] < \epsilon$ *for all* i

Uniform Integrability

Theorem (uniform integrability theorem)

If $\{X_i\}_{\mathbb{N}}$ is uniformly integrable and $X_n \to X$ as $n \to \infty$ in probability, then $X_n \to X$ as $n \to \infty$ in norm.

References I

Asai, M., McAleer, M., and Yu, J. (2006).
Multivariate stochastic volatility: A review.
Econometric Reviews, 25(2-3):145–175.

Bérard, J., Del Moral, P., and Doucet, A. (2014).
A lognormal central limit theorem for particle approximations of normalizing constants.
Electronic Journal of Probability, 19:1–28.

Dinh, L., Krueger, D., and Bengio, Y. (2014).
Nice: Non-linear independent components estimation.
arXiv preprint arXiv:1410.8516.

Kingma, D. P. and Welling, M. (2013).
Auto-encoding variational bayes.
arXiv preprint arXiv:1312.6114.

Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. (2015).
Automatic variational inference in stan.
Advances in neural information processing systems, 28.

Linderman, S. W., Wang, Y., and Blei, D. M. (2017).
Bayesian inference for latent hawkes processes.
Advances in Neural Information Processing Systems.

Moral, P. and Formulae, F.-K. (2004).
Genealogical and interacting particle systems with applications.
In *Feynman-Kac Formulae*. Springer.

Naesseth, C., Linderman, S., Ranganath, R., and Blei, D. (2018).
Variational sequential monte carlo.
In *International conference on artificial intelligence and statistics*, pages 968–977. PMLR.

References II

Williams, R. J. (1992).
Simple statistical gradient-following algorithms for connectionist reinforcement learning.
Machine learning, 8(3):229–256.