Robust Synthetic Likelihood Inference with Variational Bayes

Megan: The main goal of this research is to make VBSL more efficient and robust. But first, let's understand VBSL.

1 Derivation

Let y_{obs} be the observed data, and let $\mathcal{M}$ be a parametric model, with model parameter θ, for explaining y_{obs}. Let $p(y_{\text{obs}}|\theta)$ be the likelihood function, and $p(\theta)$ be the prior; the task is to approximate the posterior distribution

$$p(\theta|y_{\text{obs}}) \propto p(\theta)p(y_{\text{obs}}|\theta).$$

Consider the ABC problem in which the likelihood function $p(y_{\text{obs}}|\theta)$ is intractable but it is possible to generate data from the model. That is, given any value of the model parameter θ, we can generate data $y = y(\theta)$ from $\mathcal{M}$.

In the ABC literature, it is often desirable to work with a set of lower-dimensional summary statistics s_{obs} of y_{obs}; then, we work with the likelihood $p(s_{\text{obs}}|\theta)$ instead of $p(y_{\text{obs}}|\theta)$. The synthetic likelihood method further assumes that

$$p(s_{\text{obs}}|\theta) = N_d\big(s_{\text{obs}}; \mu(\theta), \Sigma(\theta)\big)$$

with d the length of s_{obs}.

In VB, we need to estimate the gradient of the lower bound

$$\begin{aligned} \nabla_\lambda \text{LB}(\lambda) &= \mathbb{E}_{q_\lambda}\left[\nabla_\lambda \log q_\lambda(\theta) \times \big(h_\lambda(\theta) - c\big)\right] && (1) \\ &= \mathbb{E}_{q_\lambda}\left[\nabla_\lambda \log q_\lambda(\theta) \times \big(\log p(\theta) + \log p(s_{\text{obs}}|\theta) - \log q_\lambda(\theta) - c\big)\right]. && (2) \end{aligned}$$

with c the vector of control variates. As the term $\log p(s_{\text{obs}}|\theta)$ is intractable, the VBSL paper suggests to replace this term by an unbiased estimator $\hat{\ell}_N(s_{\text{obs}}|\theta)$ as follows. Let $y_1,...,y_N$ be N datasets generated from model $\mathcal{M}$ given parameter θ, and let $s_1,...,s_N$ be the corresponding summary statistics. Compute the sample mean and sample variance

$$\hat{\mu}(\theta) = \frac{1}{N}\sum s_i,\ \hat{\Sigma}(\theta) = \frac{1}{N}\sum (s_i - \hat{\mu}(\theta))(s_i - \hat{\mu}(\theta))^\top.$$

Then, an unbiased estimator of $\log p(s_{\text{obs}}|\theta)$ is (ignore irrelavant constants that are independent of θ)

$$\hat{\ell}_N(s_{\text{obs}}|\theta) = -\frac{1}{2}\log \hat{\Sigma}(\theta) - \frac{N-d-2}{2(N-1)}(s_{\text{obs}} - \hat{\mu}(\theta))^\top \hat{\Sigma}(\theta)^{-1}(s_{\text{obs}} - \hat{\mu}(\theta)) \quad (3)$$

(2) becomes

$$\begin{aligned}\nabla_\lambda \mathrm{LB}(\lambda) &= \mathbb{E}_{\theta\sim q_\lambda}\Big[\nabla_\lambda \log q_\lambda(\theta)\times\big(\hat{h}_\lambda(\theta)-c\big)\Big]\\ &= \mathbb{E}_{\theta\sim q_\lambda}\Big[\nabla_\lambda \log q_\lambda(\theta)\times\big(\log p(\theta)+\hat{\ell}_N(s_{\mathrm{obs}}|\theta)-\log q_\lambda(\theta)-c\big)\Big]. \qquad (4)\end{aligned}$$

The control variates $c=(c_1,...,c_D)$, with D the length of λ, are

$$c_i=\mathrm{cov}\Big(\nabla_{\lambda_i}[\log q_\lambda(\theta)]\hat{h}_\lambda(\theta),\nabla_{\lambda_i}[\log q_\lambda(\theta)]\Big)\Big/\mathbb{V}\Big(\nabla_{\lambda_i}[\log q_\lambda(\theta)]\Big). \qquad (5)$$

2 VBSL algorithm

Let us use Cholesky Gaussian VB where $q_\lambda(\theta)=N(\mu,\Sigma)$ with $\Sigma^{-1}=CC^\top$ and C a lower triangular matrix. The variational parameter λ is

$$\lambda=\begin{pmatrix}\mu\\ \mathrm{vech}(C)\end{pmatrix},$$

and

$$\log q_\lambda(\theta)=-\frac{d}{2}\log(2\pi)+\log|C|-\frac{1}{2}(\theta-\mu)^\top CC^\top(\theta-\mu),$$

$$\nabla_\lambda \log q_\lambda(\theta)=\begin{pmatrix}CC^\top(\theta-\mu)\\ \mathrm{vech}(\cancel{\mathrm{diag}}(C^{-1})-(\theta-\mu)(\theta-\mu)^\top C))\end{pmatrix}$$

diff dimensions cant take cov

The algorithm below provides a detailed pseudo-code implementation of this CGVB approach that uses the control variate for variance reduction and moving average adaptive learning.

Algorithm 1 (VBSL algorithm)**. Input**: *Initial* $\lambda^{(0)}=(\mu^{(0)},C^{(0)})$, *adaptive learning weights* $\beta_1,\beta_2\in(0,1)$, *fixed learning rate* ϵ_0, *threshold* τ, *rolling window size* t_W *and maximum patience* P. **Model-specific requirement**: *synthetic likelihood estimate* (3).

- *Initialization*
  - *Generate* $\theta_s\sim q_{\lambda^{(0)}}(\theta)$, $s=1,...,S$.
  - *Compute the unbiased estimate of the LB gradient*

$$\widehat{\nabla_\lambda LB}(\lambda^{(0)}):=\frac{1}{S}\sum_{s=1}^{S}\nabla_\lambda \log q_\lambda(\theta_s)\times\hat{h}_\lambda(\theta_s)|_{\lambda=\lambda^{(0)}}.$$

  - *Set* $g_0:=\widehat{\nabla_\lambda LB}(\lambda^{(0)})$, $v_0:=(g_0)^2$, $\bar{g}:=g_0$, $\bar{v}:=v_0$.
  - *Estimate the vector of control variates* c *as in* (5) *using the samples* $\{\theta_s,s=1,...,S\}$.
  - *Set* $t=0$, *patience*$=0$ *and* `stop=false`.
- *While* `stop=false`*:*
  - *Calculate* $\mu^{(t)}$ *and* $C^{(t)}$ *from* $\lambda^{(t)}$. *Generate* $\theta_s\sim q_{\lambda^{(t)}}(\theta)$, $s=1,...,S$.

- *Compute the unbiased estimate of the LB gradient*[1]

$$g_t := \widehat{\nabla_\lambda LB}(\lambda^{(t)}) = \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda \log q_\lambda(\theta_s) \circ \big(h_\lambda(\theta_s) - c\big)|_{\lambda=\lambda^{(t)}}.$$

- *Estimate the new control variate vector c as in* (5) *using the samples* $\{\theta_s, s=1,...,S\}$.
- *Compute* $v_t = (g_t)^2$ *and*

$$\bar{g} = \beta_1 \bar{g} + (1-\beta_1) g_t,\ \bar{v} = \beta_2 \bar{v} + (1-\beta_2) v_t.$$

- *Compute* $\alpha_t = \min(\epsilon_0, \epsilon_0 \frac{\tau}{t})$ *and update*

$$\lambda^{(t+1)} = \lambda^{(t)} + \alpha_t \bar{g}/\sqrt{\bar{v}}$$

- *Compute the lower bound estimate*

$$\widehat{LB}(\lambda^{(t)}) := \frac{1}{S}\sum_{s=1}^{S} \hat{h}_{\lambda^{(t)}}(\theta_s).$$

- *If* $t \geq t_W$*: compute the moving averaged lower bound*

$$\overline{LB}_{t-t_W+1} = \frac{1}{t_W}\sum_{k=1}^{t_W} \widehat{LB}(\lambda^{(t-k+1)}),$$

  and if $\overline{LB}_{t-t_W+1} \geq \max(\overline{\mathrm{LB}})$ *patience = 0; else patience:=patience+1.*
- *If patience* $\geq P$, ***stop=true***.
- *Set* $t := t+1$.

3 Tasks

Implement the VBSL algorithm above and repeat the α-stable model example (Section 5.2 in the VBSL paper).

[1] The term $\nabla_\lambda \log q_\lambda(\theta_s) \circ \big(h_\lambda(\theta_s) - c\big)$ should be understood component-wise, i.e. it is the vector whose ith element is $\nabla_{\lambda_i} \log q_\lambda(\theta_s) \times \big(\hat{h}_\lambda(\theta_s) - c_i\big)$.