Mattias Villani
Department of Statistics
Stockholm University

Advanced Bayesian Learning

Computer Lab 3 - Variational Inference

The labs are the only examination, so you should do the labs **individually**.
You can use any programming language you prefer, but do **submit the code**.
Submit a readable report in **PDF** (no Word documents!) or a **JuPyteR notebook**

---

1. Consider the autoregressive process of first order

$$y_t = \mu + \phi(y_{t-1} - \mu) + \varepsilon_t, \quad \text{where } \varepsilon_t \overset{iid}{\sim} N(0, \sigma^2)$$

for $t = 1, \ldots, T$ and μ is the unconditional mean of the process $\mathbb{E}(y_t) = \mu$. Assume the priors $\mu \sim N(0, \sigma_\mu^2)$, $\phi|\sigma^2 \sim N(0, \sigma^2/\kappa_0)$ and $\sigma^2 \sim \text{Inv}-\chi^2(\nu_0, \sigma_0^2)$. For simplicity we do not impose the stationarity restriction $-1 < \phi < 1$, but instead set $\sigma_0^2 = 1$, $\kappa_0 = 8$ and $\nu_0 = 4$, so the process is stationary with prior probability ≈ 0.95 [Note that the marginal prior for ϕ is $t_{\nu_0}(0, \sigma_0^2/\kappa_0)$ so these values gives $\Pr(-1 < \phi < 1) \approx 0.95$].

   (a) Code up the Gibbs sampler in Section 9.4 in the Bayesian Learning textbook and simulate from the posterior $p(\mu, \phi, \sigma^2|\boldsymbol{y})$, where $\boldsymbol{y}$ is a vector with the time series in the file `timeseries.csv`. Set $\sigma_\mu = 2$ and use $y_0 = 0$ as initial value when constructing the lag y_{t-1}.

   (b) Derive a mean-field variational approximation for the posterior $p(\mu, \phi, \sigma^2|\boldsymbol{y})$, code it up, and compare the mean-field VI approximation to the results in 1a).

   (c) Use rstan i R/Python, Turing.jl in Julia, or some other probabilistic programming language to find a variational approximation of $p(\mu, \phi, \sigma^2|\boldsymbol{y})$ for the above AR process on the same data `timeseries.csv`. First find a mean-field variational approximation of $p(\mu, \phi, \sigma^2|\boldsymbol{y})$ and compare with your home-cooked version in 1b). Then do a variational approximation using a multivariate normal approximation $q(\mu, \phi, \sigma^2) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with a full positive definite covariance matrix $\boldsymbol{\Sigma}$. Compare the accuracy of the two approximations and the computing times. [Hint: here are the rstan documention for AR processes, which is easily transferred to Turing.jl if you prefer that language]

Good luck! Remember that sometimes an approximate answer is the correct answer!