Advanced Bayesian Learning

Lecture 6 - Beyond mean-field VI

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University

Fixed form VI

- The independence assumption in mean field VI is restrictive.
- Fixed form VI:
  - Assume parametric form $q_{\lambda}(\boldsymbol{\theta})$ with hyperparameters $\boldsymbol{\lambda}$
  - Optimize $\mathrm{KL}\left[q_{\lambda}(\boldsymbol{\theta})\,||\,p(\boldsymbol{\theta}|\boldsymbol{y})\right]$ wrt $\boldsymbol{\lambda}$.
- As before, we actually optimize lower bound $\mathrm{LB}(\boldsymbol{\lambda})$.
- Ex 1: $q_{\lambda} = N(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)$, $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{L})$, with Cholesky $\Sigma = \boldsymbol{L}\boldsymbol{L}^T$.
- Ex 2: $q_{\lambda} = N(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{a}\boldsymbol{a}^T + \boldsymbol{D})$, $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{a}, \boldsymbol{d})$, where $\boldsymbol{a}$ is a vector, $\boldsymbol{D} = \mathrm{Diag}(\boldsymbol{d})$.
- Ex 3: q_{λ} is a copula, mixture of normals etc etc
- From now on: vectors will not be **bold**.

Fixed form VI - Gradient based optimization

- **Gradient ascent**: optimize $\mathrm{LB}(\lambda)$ wrt λ using step size $a > 0$

$$\text{for } t = 1, 2, \ldots \text{ until convergence do:}$$
$$\lambda^{(t+1)} = \lambda^{(t)} + a \cdot \nabla_\lambda \mathrm{LB}(\lambda^{(t)})$$

- Stop when changes in $\mathrm{LB}(\lambda^{(t)}) < \epsilon$.
- **Lower Bound**

$$\mathrm{LB}(\lambda) = \mathbb{E}_{q_\lambda}\left[\log \frac{p(y|\theta)p(\theta)}{q_\lambda(\theta)}\right] = \int q_\lambda(\theta) h_\lambda(\theta) d\theta$$

$$h_\lambda(\theta) := \log \frac{p(y|\theta)p(\theta)}{q_\lambda(\theta)} = \log p(y|\theta)p(\theta) - \log q_\lambda(\theta).$$

Gradient of LB is an expectation wrt $q_\lambda(\theta)$

$$\nabla_\lambda \mathrm{LB}(\lambda) = \int \nabla_\lambda \left(q_\lambda(\theta) h_\lambda(\theta) \right) d\theta \text{ [interchange } \int \text{ and } \nabla_\lambda]$$
$$= \int \left(\left(\nabla_\lambda q_\lambda(\theta) \right) h_\lambda(\theta) + q_\lambda(\theta) \left(\nabla_\lambda h_\lambda(\theta) \right) \right) d\theta \text{ [product rule]}$$

Since $\nabla_\lambda \log q_\lambda(\theta) = \nabla_\lambda q_\lambda(\theta) / q_\lambda(\theta)$ we have $\nabla_\lambda q_\lambda(\theta) = q_\lambda(\theta) \nabla_\lambda \log q_\lambda(\theta)$

$$\nabla_\lambda h_\lambda(\theta) = \nabla_\lambda \left(\log p(y|\theta) p(\theta) - \log q_\lambda(\theta) \right) = -\nabla_\lambda \log q_\lambda(\theta) = -\frac{\nabla_\lambda q_\lambda(\theta)}{q_\lambda(\theta)}$$

Using that $\int q_\lambda(\theta) \left(\frac{\nabla_\lambda q_\lambda(\theta)}{q_\lambda(\theta)} \right) d\theta = \int \nabla_\lambda q_\lambda(\theta) d\theta = \nabla_\lambda \int q_\lambda(\theta) d\theta = \nabla_\lambda 1 = 0$

$$\nabla_\lambda \mathrm{LB}(\lambda) = \int \left(\left(\nabla_\lambda q_\lambda(\theta) \right) h_\lambda(\theta) + q_\lambda(\theta) \left(\nabla_\lambda h_\lambda(\theta) \right) \right) d\theta$$
$$= \int q_\lambda(\theta) \left(\nabla_\lambda \log q_\lambda(\theta) \right) h_\lambda(\theta) d\theta - \int q_\lambda(\theta) \left(\frac{\nabla_\lambda q_\lambda(\theta)}{q_\lambda(\theta)} \right) d\theta$$
$$= \int q_\lambda(\theta) \left(\nabla_\lambda \log q_\lambda(\theta) \right) h_\lambda(\theta) d\theta$$
$$= \mathbb{E}_{q_\lambda} \left(h_\lambda(\theta) \nabla_\lambda \log q_\lambda(\theta) \right)$$

VI - Stochastic gradient ascent

- **Gradient of LB is an expectation wrt $q_\lambda(\theta)$**

$$\nabla_\lambda \mathrm{LB}(\lambda) = \mathbb{E}_{q_\lambda}\left[\nabla_\lambda \log q_\lambda(\theta) \times h_\lambda(\theta)\right]$$

- **Monte Carlo**: simulate $\theta^{(1)}, \ldots, \theta^{(S)} \sim q_\lambda(\theta)$ and estimate

$$\widehat{\nabla_\lambda \mathrm{LB}(\lambda)} = \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda \log q_\lambda(\theta_s) \times h_\lambda(\theta_s)$$

Algorithm 1: Basic FFVB algorithm

Input: Initial value $\lambda^{(0)}$, tolerance ϵ, step sequence $\{a_t\}_{t=0}^{\infty}$
for $t = 0, 1, \ldots$ *until convergence* **do**

- Generate $\theta_s \sim q_{\lambda^{(t)}}(\theta)$, for $s = 1, \ldots, S$
- Estimate the LB gradient unbiasedly

$$\widehat{\nabla_\lambda \mathrm{LB}(\lambda)} = \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda \log q_\lambda(\theta_s) \times h_\lambda(\theta_s)\big|_{\lambda=\lambda^{(t)}}$$

- Update

$$\lambda^{(t+1)} = \lambda^{(t)} + a_t \cdot \widehat{\nabla_\lambda \mathrm{LB}(\lambda^{(t)})}$$

end
Output: Optimal λ

Monitoring convergence

- **Sufficient conditions for convergence**:

$$a_t > 0, \ \sum_t a_t = \infty \text{ and } \sum_t a_t^2 < \infty$$

- Example:

$$a_t = \begin{cases} \epsilon_0 & \text{if } t \leq \tau \\ \epsilon_0 \frac{\tau}{t} & \text{if } t > \tau \end{cases}$$

- Hard to monitor convergence on $\widehat{\mathrm{LB}(\lambda^{(t)})}$ since it is noisy.
- **Check convergence** on local average:

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

- MNT: $t_W = 20$ or $t_W = 50$ and tolerance $\epsilon = 10^{-5}$ common.

Adaptive learning rate

- **Learning rate** a_t should be small when $\mathbb{V}\left(\nabla_\lambda \widehat{\mathrm{LB}(\lambda^{(t)})}\right)$ is large, otherwise optimizer may backtrack.
- Algorithm above used **same learning rate** a_t for all λ_k.
- But $\mathbb{V}\left(\nabla_{\lambda_k} \widehat{\mathrm{LB}(\lambda^{(t)})}\right)$ may vary with different λ_k.
- Scale gradients with moving average of $\mathbb{V}\left(\nabla_{\lambda_k} \widehat{\mathrm{LB}(\lambda^{(t)})}\right)$
- Let $g_t = \nabla_\lambda \widehat{\mathrm{LB}(\lambda^{(t)})}$ and $v_t = g_t^2$ (elementwise, i.e. $g_t \odot g_t$).

Stochastic gradient ascent with adaptive gradients

Algorithm 2: FFVB algorithm with adaptive gradients

Input: Initial value $\lambda^{(0)}$, tolerance ϵ, step sequence $\{\alpha_t\}_{t=0}^{\infty}$, unbiased gradient estimator g, g_0 and v_0, β_1 and β_2.

Set $\bar{g} \leftarrow g_0$ and $\bar{v} \leftarrow v_0$

for $t = 0, 1, \ldots$ *until convergence* **do**

- $\bar{g} \leftarrow \beta_1 \bar{g} + (1 - \beta_1) g_t$
- $\bar{v} \leftarrow \beta_2 \bar{v} + (1 - \beta_2) g_t^2$ [elementwise]
- $\lambda^{(t+1)} = \lambda^{(t)} + \alpha_t \cdot \bar{g} / \sqrt{\bar{v}}$ [elementwise]

end

Output: Optimal λ

Natural gradient

- Same distance in λ-space can give very different changes in $\mathrm{KL}(q_\lambda \,\|\, p(\theta|y))$ depending on the geometry of $q_\lambda(\theta)$.
- Example: changing the mean of $q_\lambda(\theta)$ can have very different effect on $\mathrm{KL}(q_\lambda \,\|\, p(\theta|y))$ depending on the variance of $q_\lambda(\theta)$.
- The **natural gradient** solves this:

$$\nabla_\lambda \mathrm{LB}(\lambda)^{\mathrm{nat}} = I_F^{-1}(\lambda)\nabla_\lambda \mathrm{LB}(\lambda)$$

  where $I_F(\lambda) = \mathbb{V}_{q_\lambda}\left(\nabla_\lambda \log q_\lambda(\theta)\right)$ is the Fisher Information.
- Compute inverse by iterative conjugate gradient:
  Solve approximately $I_F(\lambda)x = \nabla_\lambda \mathrm{LB}(\lambda)$ for x.
- In exponential families, $\nabla_\lambda \mathrm{LB}(\lambda)^{\mathrm{nat}}$ is simpler than $\nabla_\lambda \mathrm{LB}(\lambda)$, see Blei et al. (2017).

Variance reduction by control variates

- The variance of gradient estimator

$$\widehat{\nabla_\lambda \mathrm{LB}(\lambda)} = \frac{1}{S}\sum_{s=1}^{S} \nabla_\lambda \log q_\lambda(\theta_s) \times h_\lambda(\theta_s)$$

  is often large. Problematic for stochastic gradient ascent.

- Estimator with control variates c_k

$$\widehat{\nabla_{\lambda_k} \mathrm{LB}(\lambda)} = \frac{1}{S}\sum_{s=1}^{S} \nabla_{\lambda_k} \log q_\lambda(\theta_s) \times \left(h_\lambda(\theta_s) - c_k\right)$$

- Unbiased since $\mathbb{E}\left(\nabla_\lambda \log q_\lambda(\theta)\right) = 0$. Lower variance.
- Optimal c_i that minimizes $\mathbb{V}\left(\widehat{\nabla_{\lambda_k} \mathrm{LB}(\lambda)}\right)$ derived in MNT.
- Optimal c_i estimated in gradient ascent. Algorithm 3, MNT.

The reparametrization trick

- Suppose we can generate $\theta \sim q_\lambda(\cdot)$ by generating $\varepsilon \sim p_\varepsilon(\cdot)$ and the deterministic transformation $\theta = g(\lambda, \varepsilon) \sim q_\lambda(\cdot)$.
- Ex: $q_\lambda(\cdot) = N(\mu, \Sigma)$, then $g(\lambda, \varepsilon) = \mu + \Sigma^{1/2}\varepsilon$, $\varepsilon \sim N(0, I)$.
- $\mathrm{LB}(\lambda)$ can be expressed as an expectation wrt $\varepsilon \sim p_\varepsilon(\cdot)$

$$\mathrm{LB}(\lambda) = \mathbb{E}_{q_\lambda}\left(h_\lambda(\theta)\right) = \mathbb{E}_{p_\epsilon}\left(h_\lambda(g(\lambda, \varepsilon))\right)$$

- The **gradient** is an expectation wrt $\varepsilon \sim p_\varepsilon(\cdot)$

$$\nabla \mathrm{LB}(\lambda) = \mathbb{E}_{p_\varepsilon}\left(\nabla_\lambda g(\lambda, \varepsilon) \nabla_\theta h_\lambda(\theta)\right)$$

since $\mathbb{E}_{p_\varepsilon}\left(\nabla_\theta h_\lambda(\theta)\right) = 0$ (MNT).

- **Unbiased estimator** by generating $\varepsilon_1, \ldots, \varepsilon_S \overset{iid}{\sim} p_\varepsilon(\cdot)$

$$\widehat{\nabla_\lambda \mathrm{LB}(\lambda)} = \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda g(\lambda, \varepsilon_s) \nabla_\theta h_\lambda(g(\lambda, \varepsilon_s))$$

- Lower variance since uses gradient information $\nabla_\theta h_\lambda(\theta)$.[1]

[1] Xu et al (2019). Variance Reduction Properties of the Reparameterization Trick. AIStats.