Notes on Bayesian Learning Rule

Daniel Iong

1 Bayesian Objective & Bayesian Learning Rule

- The Bayesian objective is an extension of the usual empirical risk minimization objective.

  Empirical risk minimization:

$$\theta_* = \arg\min_{\theta} \underbrace{\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i)) + R(\theta)}_{\bar{\ell}(\theta)}$$

  Bayesian objective:

$$q_*(\theta) = \arg\min_{q(\theta)\in\mathcal{Q}} \mathbb{E}_q\Big[\sum_{i=1}^{N} \ell(y_i, f_\theta(x_i))\Big] + \mathbb{D}_{KL}[q(\theta)||p(\theta)] \tag{1}$$

  - The prior $p(\theta)$ is related to the regularizer: $p(\theta) \propto \exp(-R(\theta))$.
  - If $\exp(-\ell(y_i, f_\theta(x_i))) \propto p(y_i|f_\theta(x_i))$, then $q_*(\theta)$ is the posterior distribution for θ
  - $\mathbb{D}_{KL}[q(\theta)||p(\theta)] = \mathbb{E}_q[\bar{\ell}(\theta)] - \mathcal{H}(q)$, where $\mathcal{H}(q) = \mathbb{E}_q[-\log q(\theta)]$ is the **entropy**.

- The class of distributions $\mathcal{Q}$ to be optimized over is assumed to be the set of a regular and minimial exponential family:

$$q(\theta) = h(\theta)\exp\big[\langle\lambda, T(\theta)\rangle - A(\lambda)\big]$$

  - $\mu = \mathbb{E}_q[T(\theta)] = \nabla_\lambda A(\lambda)$

- The **Bayesian learning rule** (BLR) is given by

$$\lambda_{t+1} \leftarrow \lambda_t - \rho_t \tilde{\nabla}_\lambda \underbrace{\big[\mathbb{E}_q[\bar{\ell}(\theta)] - \mathcal{H}(q)\big]}_{\mathbb{D}_{KL}[q(\theta)||p(\theta)]}$$

  - $\rho_t > 0$ is a sequence of learning rates.
  - The **natural gradients** are defined as

$$\tilde{\nabla}_\lambda \mathbb{E}_{q_t}(\cdot) = F(\lambda_t)^{-1}\Big[\nabla_\lambda \mathbb{E}_q(\cdot)|_{\lambda=\lambda_t}\Big]$$

    * By chain rule, $\nabla_\lambda \mathbb{E}_q(\cdot)|_{\lambda=\lambda_t} = \nabla_\mu E_q(\cdot)|_{\mu=\nabla_\lambda A(\lambda_t)} \nabla_\lambda \mu$. Therefore, they can be expressed as

$$\tilde{\nabla}_\lambda \mathbb{E}_{q_t}(\cdot) = \nabla_\mu \mathbb{E}_q(\cdot)|_{\mu=\nabla_\lambda A(\lambda_t)}$$

- **Optimality condition**: A solution q_* to eq. (1) satisfies

$$\nabla_\mu \mathbb{E}_q[\bar{l}(\theta)] = \nabla_\mu \mathcal{H}(q_*)$$

  - It can be shown that $\nabla_\mu \mathcal{H}(q) = -\lambda$ (i.e. the natural parameter of $q_*(\theta)$ is the natural gradient of the expected negative-loss).

- **Main ideas**:

  - Many well-known learning algorithms (and its variants) can be directly derived from the BLR (after choosing an appropriate $\mathcal{Q}$ and natural-gradient approximation).
  - Natural gradients retrieve essential higher-order information about the loss landscape.

2 Motivation for using natural-gradients

Each iteration of gradient-descent solves the following optimization problem:

$$\lambda_{t+1} \leftarrow \arg\min_{\lambda} \langle \nabla_\lambda \mathcal{L}(\lambda_t), \lambda \rangle + \frac{1}{2\rho_t} ||\lambda - \lambda_t||_2^2$$

- This implicitly penalizes changes in parameters.
- The parameters λ_t parameterize a distribution, but distance between two parameters might be a poor measure of distance between corresponding distributions.

This motivates the use of natural-gradient algorithms, which replaces the penalty on parameters with a penalty on distributions:

$$\lambda_{t+1} \leftarrow \arg\min_{\lambda} \langle \nabla_\lambda \mathcal{L}(\lambda_t), \lambda \rangle + \frac{1}{\rho_t} \mathcal{D}_{KL}[q_\theta || q_t(\theta)]$$

- I thought the notation was confusing here. $\mathbb{D}_{KL}$ (specifically the q's) is also a function of λ.
- I think this is what they meant by second order expansion of the KLD-term:

$$\mathbb{D}_{KL}[q_\lambda(\theta) || q_{\lambda+\delta_t}(\theta)] = \mathbb{E}_q\Big[\frac{\log q_\lambda(\theta)}{\log q_{\lambda+\delta_t}(\theta)}\Big],$$
$$\log q_{\lambda+\delta_t}(\theta) = \log q_\lambda(\theta) + [\nabla_\lambda \log q_\lambda(\theta)]'\delta_t + \delta_t'[\nabla^2 \log q_\lambda(\theta)]\delta_t,$$

  where $\delta_t = \lambda_t - \lambda$.

- The natural-gradient descent update is given by

$$\lambda_{t+1} \leftarrow \lambda_t - \rho_t F(\lambda_t)^{-1} \nabla_\lambda \mathcal{L}(\lambda_t)$$

3 Derivation of BLR

BLR can be derived as:

1. **Natural-gradient descent** using a second order expansion of KL divergence (section 2)
2. **Mirror descent** using Legendre-duality
   - *Hoping someone else can explain this b/c I'm not very familiar with mirror descent.*

4 Optimization algorithms from BLR

4.1 Examples

- Gradient descent is obtained from BLR by choosing $q(\theta) = N(\theta|m, I)$ with unknown mean m, and using delta method to approximate natural-gradient.
- Newton's method extends this by setting $q(\theta) = N(\theta|m, S^{-1})$ with unknown precision S.
- We could do multimodal optimization by setting $q(\theta) = \sum_{k=1}^K \pi_k \mathcal{N}(\theta|m_k, S_k^{-1})$.
  - The fisher info. matrix for finite mixture models could be singular so the joint $q(\theta, z = k)$, where z is the mixture-component indicator, should be used instead.
  - The authors derived a BLR update for fixed π_k.

5 Deep-learning algorithms from BLR

- Dropout is a popular regularization method in deep learning where hidden units are randomly dropped from neural networks during training. The authors propose an interesting way to include dropout into the Bayesian learning rule by using a spike-and-slab mixture for the weights.
  - I wonder what the BLR updates would look like if you use a horseshoe prior instead.

  Dropouts:

$$f_{i,l+1}(x) = h\Big(\sum_{j=1}^{n_l} \theta_{ijl} z_{jl} f_{jl}(x)\Big),$$

  where $f_{jl}(x)$ denotes the j'th unit in the l'th layer for input x; z_{jl} are independent Bernoulli w/ $P(z_{jl} = 1) = \pi_1$; and θ_{ijl} are the weights.

  Spike-and-slab mixture for θ_{jl}

$$q(\theta_{jl}) = \pi \mathcal{N}(\theta_{jl} | m_{jl}, S_{jl}^{-1}) + (1 - \pi_1)\mathcal{N}(0, s_0^{-1} I_{n_l}),$$

  where $s_0 > 0$ is a fixed small value.
  - Spike at 0 to induce sparsity

  Horseshoe (**Not in the paper**)

$$q(\theta_{jl} | \lambda_i) = \mathcal{N}(\theta_{jl} | 0, \lambda_i^2 \tau^2),\ q(\lambda_i) = C^+(0, 1),$$

  where $C^+(0, 1)$ is the half-Cauchy distribution.