Laplace Approximation

CS698X: Topics in Probabilistic Modeling and Inference

Piyush Rai

Laplace Approximation of Posterior Distribution

- Consider a posterior distribution that is intractable to compute

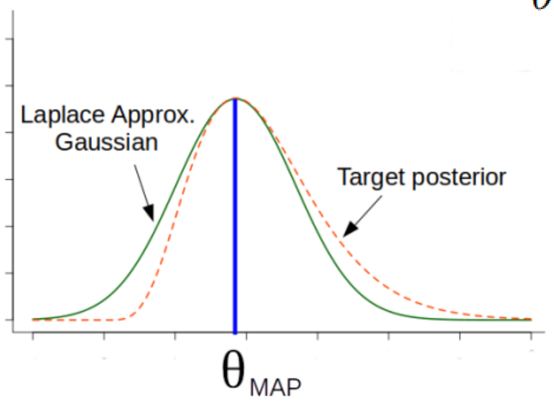
Unknowns of the model

Data

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D},\theta)}{p(\mathcal{D})}$$

- Laplace approximation approximates the above using a Gaussian distribution

$$\boxed{p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta_{MAP}, \mathbf{H}^{-1})}$$



$$\begin{aligned}\theta_{MAP} &= \arg\max_\theta p(\theta|\mathcal{D}) = \arg\max_\theta p(\mathcal{D},\theta) \\ &= \arg\max_\theta p(\mathcal{D}|\theta)p(\theta) \\ &= \arg\max_\theta [\log p(\mathcal{D}|\theta) + \log p(\theta)]\end{aligned}$$

$$\begin{aligned}\mathbf{H} &= -\nabla^2 \log p(\theta|\mathcal{D})\big|_{\theta=\theta_{MAP}} = -\nabla^2 \log p(\mathcal{D},\theta)\big|_{\theta=\theta_{MAP}} \\ &= -\nabla^2 [\log p(\mathcal{D}|\theta) + \log p(\theta)]\big|_{\theta=\theta_{MAP}}\end{aligned}$$

- Why is the above Gaussian a reasonable approximation to the posterior?

Derivation of the Laplace Approximation

- Let's write the Bayes rule as

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D},\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D},\theta)}{\int p(\mathcal{D},\theta)d\theta} = \frac{e^{\log p(\mathcal{D},\theta)}}{\int e^{\log p(\mathcal{D},\theta)}d\theta}$$

$$\approx \frac{1}{2}(\theta-\theta_{MAP})^\top \underbrace{\nabla^2 \log p(\mathcal{D},\theta_{MAP})}_{-\mathbf{H}} (\theta-\theta_{MAP}) + \text{const}$$

Recall that Hessian is the second derivative of the negative of log-joint

Aha! This is a Gaussian!

Comparing with a Gaussian PDF
Mean = $\boldsymbol{\theta}_{MAP}$
Cov. Matrix = $\mathbf{H}^{-1}$

- Approximating $\log p(\mathcal{D},\theta)$ by a quadratic function of $\boldsymbol{\theta}$ will make it a Gaussian
- Consider the second-order Taylor approx of a function $f(\boldsymbol{\theta})$ around some $\boldsymbol{\theta}_0$

$$f(\theta) \approx f(\theta_0) + (\theta-\theta_0)^\top \nabla f(\theta_0) + \frac{1}{2}(\theta-\theta_0)^\top \nabla^2 f(\theta_0)(\theta-\theta_0)$$

- Assuming $f(\boldsymbol{\theta}) = \log p(\mathcal{D},\boldsymbol{\theta})$ and $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_{MAP}$, $\nabla f(\theta_{MAP}) = \nabla \log p(\mathcal{D},\theta_{MAP}) = 0$

$$\log p(\mathcal{D},\theta) \approx \underbrace{\log p(\mathcal{D},\theta_{MAP})}_{\text{Constant w.r.t. } \theta} + \frac{1}{2}(\theta-\theta_{MAP})^\top \nabla^2 \log p(\mathcal{D},\theta_{MAP})(\theta-\theta_{MAP})$$

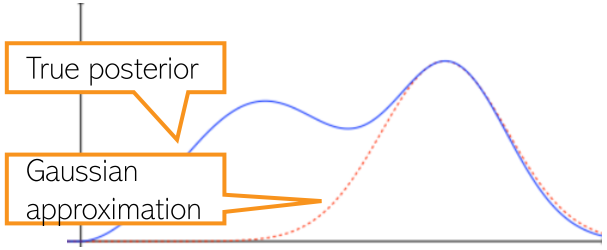
- Thus Laplace approx. is based on a second-order Taylor approx. of the posterior

Properties of Laplace Approximation

- Usually straightforward if derivatives (first and second) can be computed easily
- Expensive if parameter $\boldsymbol{\theta}$ is very high dimensional

E.g., a deep neural network, or even in simpler models (e.g., logistic reg with a very large number of features

  - Reason: We need to invert the Hessian whose size is $D \times D$ (D is the # of params)
- Can do badly if the (true) posterior is multimodal



- Applicable only when $\boldsymbol{\theta}$ is real-valued (won't if, say, it is positive, binary etc)
- Note: Even if we have a <u>non-probabilistic</u> model (loss function + regularization), we can obtain an approx "posterior" for that model using the Laplace approximation
  - Optima of the regularized loss function will be Gaussian's mean
  - Second derivative of the regularized loss function will be the Hessian

Laplace Approx. for High-Dimensional Problems

- When $\boldsymbol{\theta}$ is very high dim, one option is to approximate the Hessian itself
- One such approx. of the Hessian is a diagonal approximation

Fisher Information Matrix (FIM)

$$\mathbf{H} \approx \text{diag}(\mathbf{F})$$

FIM is easily computable in auto-diff frameworks used in deep learning

$$\begin{aligned}\mathbf{F} &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})}\left[\nabla \log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) \nabla \log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})^\top\right] \\ &\approx \mathbb{E}_{p_D(\mathbf{x},\mathbf{y})}\left[\nabla \log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) \nabla \log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})^\top\right] \\ &= \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \nabla \log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) \nabla \log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})^\top\end{aligned}$$

Assuming a discriminative model with parameters $\boldsymbol{\theta}$

Example: A Bayesian neural net for regression/classification ($\boldsymbol{\theta}$ denotes the weights of the network)

- The diagonal approx. of Hessian may be too crude ☹
  - Ignores covariances among params and treats them as being independent of each other
- A block-diagonal approx. proposed recently (in the context of deep neural nets)
  - Treats params across layers to be independent but correlated within the same layer
  - The approach known as Kronecker-Product Factored (KFAC) Laplace approximation

KFAC paper: "A Scalable Laplace Approximation for Neural Networks" (Ritter et al, ICLR 2018)

Coming Up

- Generalized Linear Models (GLM)
  - Models of the form $p(y|x)$ where $p(y|x)$ is some exponential family distribution
  - Note: Prob. linear regression and logistic regression were also examples of GLMs
- Generative Classification