MAP estimate on GLMs

DATA : y_n output , $x_n \in \mathbb{R}^D$ input vector, $n = 1, 2, \ldots, N$

LINEAR MODEL: $y_n = x_n^T \underbrace{z}_{\text{weights}} + \underbrace{\epsilon_n}_{\text{Noise}} \Rightarrow p(y_n / x_n^T z) = \mathcal{N}(x_n^T z, \sigma^2)$

(Let $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$)

GENERALIZED : Replace p by any Exp-Family distribution.
(GLM)

MAXIMUM LIKELIHOOD : Estimate z !
(MLE)

$$z_{MLE} = \arg\max_z \sum_{n=1}^{N} \log p(y_n / x_n^T z) \quad \underbrace{- \frac{\lambda}{2} \|z\|_2^2}$$

MAP:

$$z_{MAP} = \arg\max_z \quad // \quad + \log \mathcal{N}(z / 0, 1/\lambda \, I)$$

$$\boxed{z_{MAP} = \arg\max_z \underbrace{\sum_{n=1}^{N} \log p(y_n / z^T x_n)}_{\text{Data term}} + \underbrace{\log p(z)}_{\text{Regularizer}}}$$

Stochastic Gradient Descent (SGD)

$$\underline{z}_{MAP} = \arg\max_{\underline{z}} \overbrace{\sum_{n=1}^{N} \log p(y_n / \underline{z}^T \underline{x}_n) + \log p(\underline{z})} := L(\underline{z})$$

Randomly sample an n and compute an "unbiased" estimate

$$\hat{L}(z) = N \underbrace{\log p(y_n / \underline{z}^T \underline{x}_n)} + \log p(z), \quad \mathbb{E}[\hat{L}(z)] = L(z)$$

$$\widehat{\frac{\partial L}{\partial z}} = N \frac{\partial}{\partial z} \text{''} + \frac{\partial}{\partial z} \text{''}$$

scalar step-size >0

SGD UPDATE: $$z_{t+1} = z_t + \alpha_t \left. \widehat{\frac{\partial L}{\partial z}} \right|_{z=z_t}$$

Q: converge?

MAP to Bayesian Inference

MAP: $$\underline{z}_{MAP} = \arg\max_{\underline{z}} \underbrace{\sum_{n=1}^{N} \log p(y_n / \underline{z}^T \underline{x}_n) + \log p(\underline{z})}_{\log \left[\prod_{n=1}^{N} p(y_n | z^T x_n) \times p(z) \right]}$$

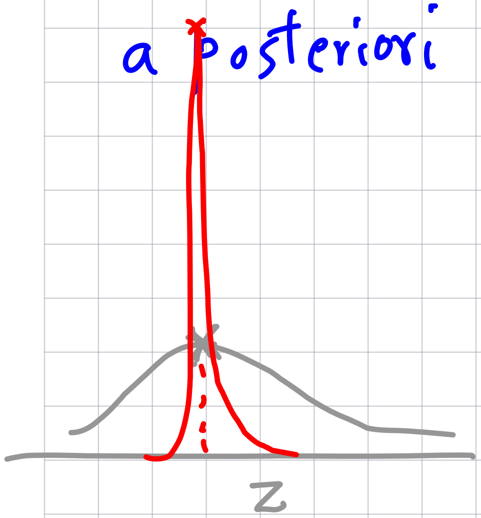
↓

Maximum a posteriori

$$= \arg\max_{\underline{z}} \cancel{\log}\, p(y, z)$$

$$= \text{Mode of } p(y, z) = \text{Mode of } p(z / y)$$

$$= \text{Max. of a Posterior}$$



Bayes rule: $$\underbrace{p(\underline{z} | \underline{y})}_{\text{Posterior}} = \frac{p(\underline{y}, \underline{z})}{\underbrace{\int p(\underline{y}, \underline{z})\, d\underline{z}}_{}} = \frac{p(y, z)}{p(y)}$$

← Normalizing constant

INTRACTABLE INTEGRAL

Computational issues

GLM : $$p(\underline{z}\,|\,\underline{y}) \propto \overbrace{\prod_{n=1}^{N} p(y_n / \underline{z}^T \underline{x}_n)\; p(\underline{z})}^{p(y,z)}$$

POSTERIOR — Likelihoods — Prior

Gaussian — Gaussian (say)

CONJUGATE PRIORS : $$\int \underbrace{\prod_{n=1}^{N} \mathcal{N}(y_n / z^T x_n, \sigma^2) \times \mathcal{N}(z/0, I)}\, dz$$

Complete the square!

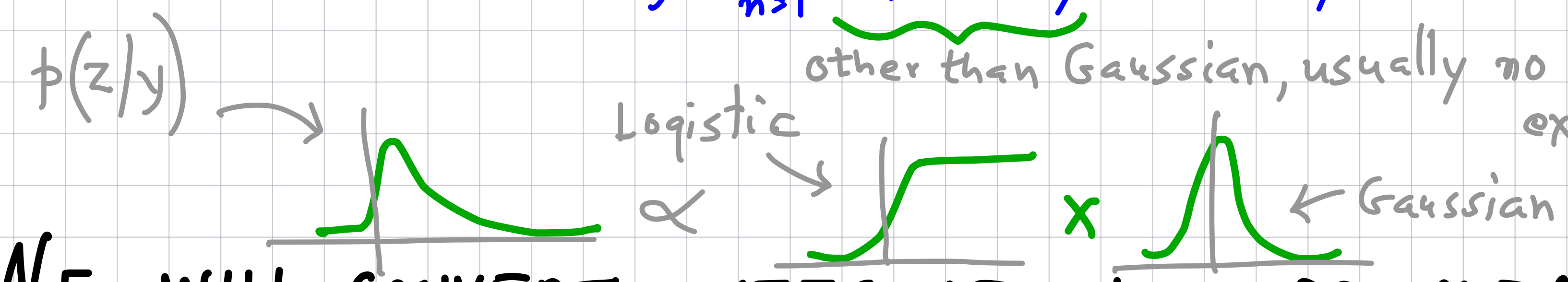
$$= c \times \int \mathcal{N}(z/\;,\;)\, dz \;=\; \log \det |\Sigma|$$

A covariance matrix!

Q: Comp. cost for Gaussian?

NON-CONJUGATE : $$\int \prod_{n=1}^{N} \underbrace{p(y_n / \ldots)} \times \mathcal{N}(z/0, I)\, dz$$

other than Gaussian, usually no closed-form expression

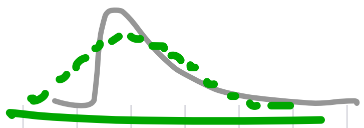


WE WILL CONVERT INTEGRATION → OPTIMIZATION

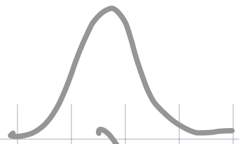
Question

Why is Bayesian inference computationally challenging?

Variational Inference



MAIN IDEA: $p(\underline{z} \mid \underline{y}) \simeq q(z;\lambda)$ e.g. $= \mathcal{N}\big(z \,/\, \underbrace{\underset{\text{mean}}{m},\ \underset{\text{covariance}}{V}}_{=\lambda}\big)$



LOWER BOUND:

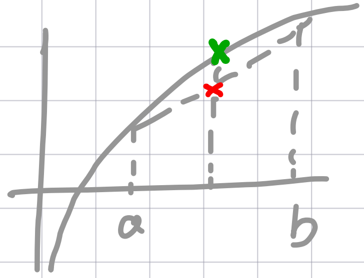
$$\log p(y) = \log \int p(y,z)\,dz = \log \int q(z;\lambda)\,\frac{p(y,z)}{q(z;\lambda)}\,dz$$

$$\geq \int q(\;)\,\log \frac{\prod_n p(y_n \mid \cdot)\,p(z)}{q(z;\lambda)} \quad \text{(Jensen's inequality)}$$

$$= \sum_{n=1}^{N} \int q(\;)\,\log p(y_n \mid \cdot)\,dz - \int q(\;)\,\log \frac{q(z;\lambda)}{p(z)}\,dz$$

$\log[ta + (1-t)b]$ (green)

$\geq t\log a + (1-t)\log b$ (red)



VI: $$\max_{\lambda} \sum_{n=1}^{N} \mathbb{E}_{q(z;\lambda)}\left[\log p(y_n \mid x_n^T z)\right] - D_{KL}\left[q(z;\lambda) \,\|\, p(z)\right]$$

Data-term Regularizer

Q: Equality?

Question

How does variational inference solve the computational problems with Bayesian Inference?

VI by using SGD

(SEE "VI in five lines in python")

MAP: $\max_{z} \sum_{n=1}^{N} \log p(y_n / z^T x_n) + \log p(z)$

VI: $\max_{\lambda} \sum_{n=1}^{N} \mathbb{E}_{q(z;\lambda)} \left[\log p(y_n \mid x_n^T z) \right] - D_{KL}\left[q(z;\lambda) \,\|\, P(z) \right] = \mathcal{L}(\lambda)$

How to compute an unbiased stochastic gradient?

Doubly-stochastic estimate: Sample an "n" at random

Sample a $z_t^* \sim q(z;\lambda_t)$

$L(\lambda) \approx \hat{L}(\lambda) = N \log p(y_n / x_n^T z_t^*) - D_{KL}\left[q(z_t^* / \lambda) \,\|\, P(z_t^*) \right]$

SGD: $\lambda_{t+1} = \lambda_t + \alpha_t \left. \widehat{\frac{\partial \mathcal{L}}{\partial \lambda}} \right|_{\lambda = \lambda_t}$

gradient w.r.t. λ

Reparameterization trick: say $\lambda = \{m, \sigma^2\}$ then, $z^* = m + \sigma \epsilon$ where $\epsilon \sim \mathcal{N}(0,1)$

Q: Issues?

Question

Describe one method to perform variational inference.

Discuss computational challenges associated with it.

Mean-Field Inference

ISSUE : size of λ could be > size of z !

EXAMPLE :

SOLUTION :

Mean-field : $q(z;\underline{\lambda}) = \prod_{i=1}^{D} q_i(z_i;\underline{\lambda}_i)$

Q: Issues?

Variational Auto-Encoder

Kingma & Welling (ICLR 2014)

Consider unsupervised learning with data $y_n \in \mathbb{R}^D$, $n = 1, \dots, N$

Variational Auto-Encoder

Variational Lower bound : $\mathbb{E}_{q_\phi}\left[\log p_\theta(y|z)\right] - D_{KL}\left[q_\phi \| p(z)\right]$

" SGD + variance reduction " works quite well !

SHOW RESULTS

Question

How does variational auto-encoder solve the computational problem with the standard mean-field inference?

Assignment

Question 1: Why is Bayesian inference computationalley challenging?

Question 2: How does variational inference solve the computational problems with Bayesian Inference?

Question 3: Describe one method to perform variational inference. Discuss computational challenges associated with it?

~~Question 4: How does variational auto-encoder solve the computational problem with the standard mean-field inference?~~