Dropout as a Bayesian Approximation

Jaewoo Park

Department of Applied Statistics, Yonsei University

Variational Inference (VI)

- Approximates the posterior through $q_\xi(\boldsymbol{W}, \boldsymbol{b})$ (variational distribution)
- Find $q_\xi(\boldsymbol{W}, \boldsymbol{b})$ that minimizes the Kullback–Leibler (KL) divergence between $q_\xi(\boldsymbol{W}, \boldsymbol{b})$ and $p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})$
  - We need to set a class of distributions (e.g., Gaussian)
  - A practical option for BNN

Variational Inference (contd.)

- KL divergence is defined as

$$\mathrm{KL}(q_\xi(\boldsymbol{W}, \boldsymbol{b}) || p(\boldsymbol{W}, \boldsymbol{b} | \boldsymbol{X}, \boldsymbol{Y})) = \int \int q_\xi(\boldsymbol{W}, \boldsymbol{b}) \frac{q_\xi(\boldsymbol{W}, \boldsymbol{b})}{p(\boldsymbol{W}, \boldsymbol{b} | \boldsymbol{X}, \boldsymbol{Y})} d\boldsymbol{W} d\boldsymbol{b}$$

  which is intractable due to $p(\boldsymbol{W}, \boldsymbol{b} | \boldsymbol{X}, \boldsymbol{Y})$ terms
- Minimizing the KL divergence is equivalent to maximizing evidence lower-bound (ELBO)

$$\mathrm{ELBO} = \int \int q_\xi(\boldsymbol{W}, \boldsymbol{b}) \log p(\boldsymbol{Y} | \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) d\boldsymbol{W} d\boldsymbol{b} - \mathrm{KL}(q_\xi(\boldsymbol{W}, \boldsymbol{b}) || p(\boldsymbol{W}) p(\boldsymbol{b}))$$

Variational Inference (contd.)

- We don't need the posterior $p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})$ when we compute ELBO

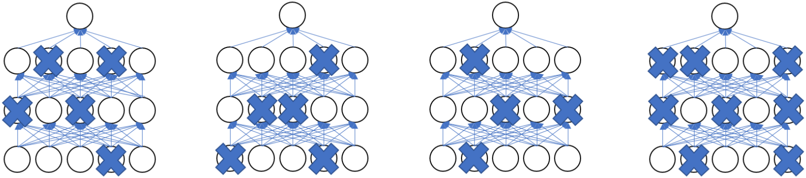
$$\text{ELBO} = \int\int q_{\xi}(\boldsymbol{W}, \boldsymbol{b}) \log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) d\boldsymbol{W} d\boldsymbol{b} -$$

$$\text{KL}(q_{\xi}(\boldsymbol{W}, \boldsymbol{b}) || p(\boldsymbol{W}) p(\boldsymbol{b}))$$

- All we need is
  - Prior: $p(\boldsymbol{W})p(\boldsymbol{b})$
  - Variational distribution: $q_{\xi}(\boldsymbol{W}, \boldsymbol{b})$
  - Likelihood: $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b})$
- Still challenging for obtaining analytical solutions

Monte Carlo (MC) Dropout

- MC dropout (Gal and Ghahramani, 2016) is an practical option
  1. Fit a neural network in a usual way
  2. Based on the fitted model, apply different dropouts to make prediction
- We are fitting the model just a single time (not multiple times)
- MC dropout can approximate ELBO (see later)



- Consider we have $\widehat{\boldsymbol{W}}, \widehat{\boldsymbol{b}}$ from SGD
- For given $\widehat{\boldsymbol{W}}, \widehat{\boldsymbol{b}}$, dropout randomly removes the connections from nodes
- For each dropout, we will have different output realizations (predictive distribution)

Dropout as a Bayesian Approximation

- Go back to VI, we have

$$\text{ELBO} = \int\int q_{\xi}(\boldsymbol{W}, \boldsymbol{b}) \log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) d\boldsymbol{W} d\boldsymbol{b} -$$

$$\text{KL}(q_{\xi}(\boldsymbol{W}, \boldsymbol{b}) || p(\boldsymbol{W}) p(\boldsymbol{b}))$$

- All we need is
  - Prior: $p(\boldsymbol{W})p(\boldsymbol{b}) \rightarrow$ standard normal prior
  - Variational distribution: $q_{\xi}(\boldsymbol{W}, \boldsymbol{b}) = q_{\xi}(\boldsymbol{W}) q_{\xi}(\boldsymbol{b}) \rightarrow$ mixture normal
  - Likelihood: $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) \rightarrow$ (nested) Gaussian process

Dropout as a Bayesian Approximation: $q(\boldsymbol{W}, \boldsymbol{b})$

- We use a mixture normal distribution as

$$q_{\xi}(\boldsymbol{W}) = \prod_{\forall i} q_{\xi}(w_i), \quad q_{\xi}(\boldsymbol{b}) = \prod_{\forall i} q_{\xi}(b_i)$$
$$q_{\xi}(w_i) = pN(\mu_i^w, \sigma^2) + (1-p)N(0, \sigma^2)$$
$$q_{\xi}(b_i) = pN(\mu_i^b, \sigma^2) + (1-p)N(0, \sigma^2)$$

- Bayesian variable selection (dropout)
  - $p \to 0$: likely to drop $(w_i, b_i = 0)$ parameters
  - $p \to 1$: likely to include $(w_i, b_i \neq 0)$ parameters

Dropout as a Bayesian Approximation: $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b})$

- How can we define a likelihood function?
- Let's use a nested Gaussian process

$$\boldsymbol{F}_l|\boldsymbol{F}_{l-1} \sim N(0, \Sigma_l), \quad l = 2, \dots, L$$
$$\boldsymbol{Y}|\boldsymbol{F}_L \sim p(\boldsymbol{Y}|\boldsymbol{F}_L)$$

Here $p(\cdot|\boldsymbol{F}_L)$ is a exponential family distribution

Dropout as a Bayesian Approximation: ELBO

- Remind:

$$\text{ELBO} = \int\int q_{\xi}(\boldsymbol{W})q_{\xi}(\boldsymbol{b}) \log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) d\boldsymbol{W} d\boldsymbol{b} -$$

$$\mathrm{KL}(q_{\xi}(\boldsymbol{W})q_{\xi}(\boldsymbol{b})||p(\boldsymbol{W})p(\boldsymbol{b}))$$

- We can approximate this as

$$\mathcal{L}_{\text{GP-MC}} := \frac{1}{M}\sum_{m=1}^{M} \log p(\boldsymbol{Y}|\boldsymbol{X}, \widehat{\boldsymbol{W}}_m, \widehat{\boldsymbol{b}}_m) - \mathrm{KL}(q_{\xi}(\boldsymbol{W})q_{\xi}(\boldsymbol{b})||p(\boldsymbol{W})p(\boldsymbol{b}))$$

where $\{\widehat{\boldsymbol{W}}_m, \widehat{\boldsymbol{b}}_m\}_{m=1}^{M}$ are MC samples from $q_{\xi}(\boldsymbol{W})q_{\xi}(\boldsymbol{b})$

Dropout as a Bayesian Approximation: Prediction

- The predictive distribution is

$$q(\boldsymbol{Y}^*|\boldsymbol{X}^*) = \int\int p(\boldsymbol{Y}^*|\boldsymbol{X}^*, \boldsymbol{W}, \boldsymbol{b}) q_{\xi}(\boldsymbol{W}) q_{\xi}(\boldsymbol{b}) d\boldsymbol{W} d\boldsymbol{b}$$

- For given unobserved $\boldsymbol{X}^*$, $\boldsymbol{Y}^*$

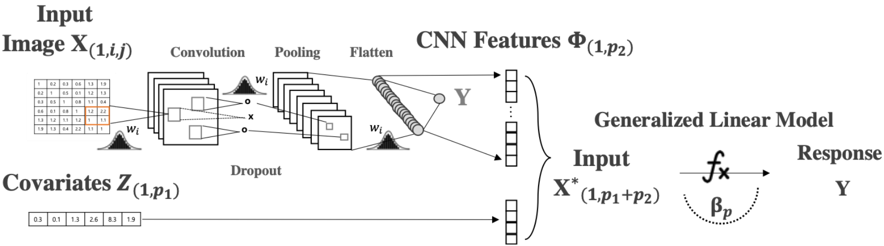
$$\mathbb{E}_q(\boldsymbol{Y}^*) = \int \boldsymbol{Y}^* q(\boldsymbol{Y}^*|\boldsymbol{X}^*) d\boldsymbol{Y}^* \approx \frac{1}{M} \sum_{m=1}^{M} \widehat{\boldsymbol{Y}}_m^*$$

where $\widehat{\boldsymbol{Y}}_m^*$ is sampled from $q(\boldsymbol{Y}^*|\boldsymbol{X}^*)$

A Bayesian Convolutional Neural Network-based Generalized Linear Model (Bayes CGLM)

- Goal: Analyze relationships between a response $\boldsymbol{Y}$ and inputs $\boldsymbol{X}, \boldsymbol{Z}$
- Statistical Methods:
  - Provide interpretation through regression coefficients
  - Quantify uncertainties in estimates/predictions
  - Hard to analyze high-dimensional variables (e.g., image)
- Deep Learning Methods:
  - Popular for high-dimensional variables with high accuracy
  - Hard to interpret the impact of covariates
  - Uncertainty quantification is not trivial
- Jeon et al., (2023) combine statistical and deep learning models

Bayes CGLM



- Step 1: From $\boldsymbol{X}$, Extract $\{\Phi^{(m)}\}_{m=1}^{M}$ via Bayes CNN
- Step 2: Fit Bayes GLMs by regressing $\boldsymbol{Y}$ on $[\boldsymbol{Z}, \Phi^{(m)}]$
- Step 3: Construct an ensemble-posterior distribution

Part 1: Extract Feature Information

- Idea: Monte Carlo dropout (Gal and Ghahramani 2015, 2016)
    - A deep Gaussian process (Damianou and Lawrence, 2013) can represent complex DNN
    - Dropout can approximate the deep Gaussian process
    - We extract MC samples $\{\Phi^{(m)}\}_{m=1}^{M}$ (features) from the last layer of Bayes CNN
- Benefits:
    - Reduce the dimension of high-dimensional $\boldsymbol{X}$
    - Simultaneously analyze both $\boldsymbol{X}$ and $\boldsymbol{Z}$ with appropriate uncertainty quantification

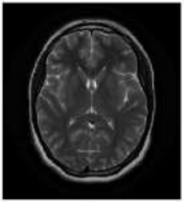
Part 2: Fit Bayes GLMs

- Obtain mth feature-posterior:
  - Model:
$$E[\boldsymbol{Y}|\boldsymbol{Z}, \Phi^{(m)}] = \boldsymbol{Z}\gamma_m + \Phi^{(m)}\boldsymbol{\delta}_m = \boldsymbol{A}^{(m)}\boldsymbol{\beta}_m$$
  - $\boldsymbol{Z}$: covariate matrix
  - $\Phi^{(m)}$: extracted feature from mth dropout sample
  - Fit Bayes LM by regressing $\boldsymbol{Y}$ on $\boldsymbol{A}^{(m)} = [\boldsymbol{Z}, \Phi^{(m)}]$

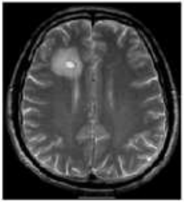
Part 3: An Ensemble Posterior

- We obtain M number of feature-posterior
- We combine them to construct the aggregated posterior
  - You can simply combine posterior samples from each m
- Such aggregation can fully account for uncertainties in the feature extraction (Step 1)

Brain Tumor Image Data



No



Yes

The dataset is collected from Brain Tumor Image Segmentation Challenge (BRATS)

- $\boldsymbol{Y}$: Binary response (brain tumor or not)
- $\boldsymbol{X}$: 240×240 pixel gray images of brains
- $\boldsymbol{Z}$: Vector covariates (first and second order features)
- $N = 2,508$ for training and $N_{\mathrm{cv}} = 2,007$ for validation

Brain Tumor Image Data (contd.)

	BayesCGLM(M=500)	Bayes CNN	GLM
γ_1 (first order feature)	-5.332	0.248	-2.591
	(-7.049,-3.704)	-	(-2.769,-2.412)
γ_2 (second order feature)	4.894	0.160	2.950
	(3.303, 6.564)	-	(2.755, 3.144)
Accuracy	0.924	0.867	0.784
Recall	0.929	0.787	0.783
Precision	0.901	0.907	0.715
Time	293.533	103.924	0.004

Table: For all methods, the posterior mean of $\boldsymbol{\gamma}$, 95% HPD interval, accuracy, recall, precision, and computing time (min) are reported in the table.

Discussion

Bayes CNN-basd GLM (BayesCGLM)

- Extract the important feature of the high-dimensional variables
- Simultaneously utilize covariates with different data structures (e.g., vector and image)
- Provide accurate inference with uncertainty quantification in both estimation/prediction
- Provide interpretation of the impact of covariates
- Computationally efficient due to parallelization