Bayesian Linear Regression (Hyperparameter Estimation, Sparse Priors), Bayesian Logistic Regression

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

Jan 21, 2019

Recap: Bayesian Linear Regression

- Assume Gaussian likelihood: $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1}) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N)$
- Assume zero-mean spherical Gaussian prior: $p(\boldsymbol{w}|\lambda) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, \lambda^{-1}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$
- Assuming hyperparameters as fixed, the posterior is Gaussian

$$
\begin{aligned}
p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) &= \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N) \\
\boldsymbol{\Sigma}_N &= (\beta \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top + \lambda \mathbf{I}_D)^{-1} = (\beta \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} \quad \text{(posterior's covariance matrix)} \\
\boldsymbol{\mu}_N &= \boldsymbol{\Sigma}_N \left[\beta \sum_{n=1}^{N} y_n \boldsymbol{x}_n\right] = \boldsymbol{\Sigma}_N \left[\beta \mathbf{X}^\top \boldsymbol{y}\right] = (\mathbf{X}^\top \mathbf{X} + \frac{\lambda}{\beta}\mathbf{I}_D)^{-1} \mathbf{X}^\top \boldsymbol{y} \quad \text{(posterior's mean)}
\end{aligned}
$$

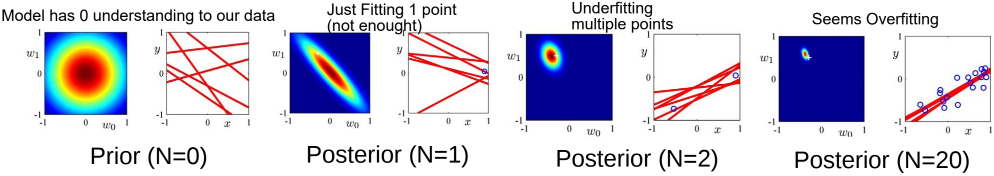
- The posterior predictive distribution is also Gaussian

$$
p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}, \beta, \lambda) = \int p(y_*|\boldsymbol{w}, \boldsymbol{x}_*, \beta) p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}, \beta, \lambda) d\boldsymbol{w} = \mathcal{N}(\boldsymbol{\mu}_N^\top \boldsymbol{x}_*, \beta^{-1} + \boldsymbol{x}_*^\top \boldsymbol{\Sigma}_N \boldsymbol{x}_*)
$$

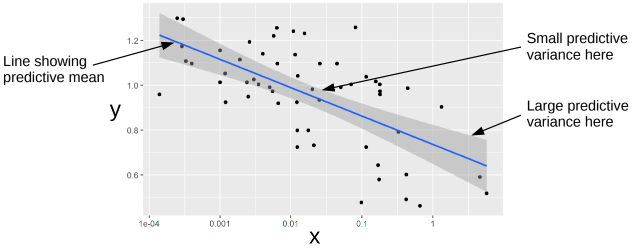
- Gives both predictive mean and predictive variance (imp: pred-var is different for each input)

A Visualization of Uncertainty in Bayesian Linear Regression

- Posterior $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$ and lines (w_0 intercept, w_1 slope) corresponding to some random $\boldsymbol{w}$'s

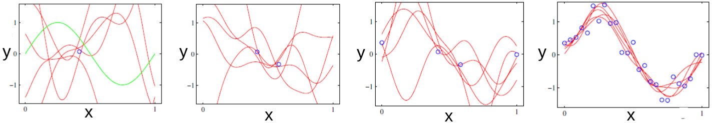


- A visualization of the posterior predictive of a Bayesian linear regression model

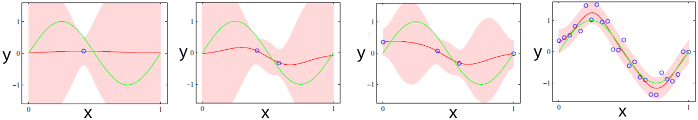


A Visualization of Uncertainty (Contd)

- We can similarly visualize a Bayesian nonlinear regression model
- Figures below: Green curve is the true function and blue circles are observations (x_n, y_n)
- Posterior of the nonlinear regression model: Some curves drawn from the posterior



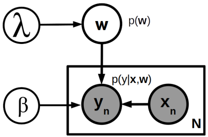
- Posterior predictive: Red curve is predictive mean, shaded region denotes predictive uncertainty



Estimating Hyperparameters for Bayesian Linear Regression

Learning Hyperparameters in Probabilistic Models

- Can treat hyperparams as just a bunch of additional unknowns
- Can be learned using a suitable inference algorithm (point estimation or fully Bayesian)
- Example: For the linear regression model, the full set of parameters would be $(\boldsymbol{w}, \lambda, \beta)$



- Can assume priors on all these parameters and infer their "joint" posterior distribution

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}, \lambda, \beta)}{p(\boldsymbol{y}|\mathbf{X})} = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta, \lambda) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda)}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\lambda) p(\beta) p(\lambda)\ d\boldsymbol{w}\ d\lambda d\beta}$$

- Infering the above is usually intractable (rare to have conjugacy). Requires approximations. Also,
  - What priors (or "hyperpriors") to choose for β and λ?
  - What about the hyperparameters of those priors?

Learning Hyperparameters via Point Estimation

- One popular way to estimate hyperparameters is by maximizing the marginal likelihood
- For our linear regression model, this quantity (a function of the hyperparams) will be

$$p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

- The "optimal" hyperparameters in this case can be then found by

$$\hat{\beta}, \hat{\lambda} = \arg\max_{\beta,\lambda} \log p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda)$$

- This is called MLE-II or (log) evidence maximization
  - Akin to doing MLE to estimate the hyperparameters where the "main" parameter (in this case $\boldsymbol{w}$) has been integrated out from the model's likelihood function
- Note: If the likelihood and prior are conjugate then marginal likelihood is available in closed form

What is MLE-II Doing?

- For linear regression case, would ideally like the posterior over all unknowns, i.e., $p(\boldsymbol{w}, \lambda, \beta | \mathbf{X}, \boldsymbol{y})$

$$p(\boldsymbol{w}, \beta, \lambda | \mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda) p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \qquad \text{(from product rule)}$$

- Note that $p(\boldsymbol{w} | \mathbf{X}, \boldsymbol{y}, \beta, \lambda)$ is easy if λ, β are known
- However $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y} | \mathbf{X}, \beta, \alpha) p(\beta) p(\lambda)}{p(\boldsymbol{y} | \mathbf{X})}$ is hard (lack of conjugacy, intractable denominator)
- Let's approximate it by a point function δ at the mode of $p(\beta, \lambda | \mathbf{X}, \boldsymbol{y})$

$$p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda}) \quad \text{where} \quad \hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\beta, \lambda | \mathbf{X}, \boldsymbol{y}) = \arg\max_{\beta, \lambda} p(\boldsymbol{y} | \mathbf{X}, \beta, \lambda) p(\lambda) p(\beta)$$

- Moreover, if $p(\beta)$, $p(\lambda)$ are uniform/uninformative priors then

$$\hat{\beta}, \hat{\lambda} = \arg\max_{\beta, \lambda} p(\boldsymbol{y} | \mathbf{X}, \beta, \lambda)$$

- Thus MLE-II is approximating the posterior of hyperparams by their point estimate assuming uniform priors (therefore we don't need to worry about a prior over the hyperparams)

MLE-II for Linear Regression

- For the linear regression case, the marginal likelihood is defined as

$$p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$

- Since $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I}_N)$ and $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}\mathbf{I}_D)$, the marginal likelihood

$$\begin{aligned} p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda) &= \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \beta^{-1}\mathbf{I} + \lambda^{-1}\mathbf{X}\mathbf{X}^\top) \\ &= \frac{1}{(2\pi)^{N/2}}|\beta^{-1}\mathbf{I} + \lambda^{-1}\mathbf{X}\mathbf{X}^\top|^{-1/2}\exp(-\frac{1}{2}\boldsymbol{y}^\top(\beta^{-1}\mathbf{I} + \lambda^{-1}\mathbf{X}\mathbf{X}^\top)^{-1}\boldsymbol{y}) \end{aligned}$$

- MLE-II maximizes $\log p(\boldsymbol{y}|\mathbf{X}, \beta, \lambda)$ w.r.t. β and λ to estimate these hyperparams
  - This objective doesn't have a closed form solution
  - Solved using iterative/alternating optimization
  - PRML Chapter 3 contains the iterative update equations
- Note: Can also do "MAP-II" using a suitable prior on these hyperparams (e.g., gamma)
- Note: Can also use different λ_d for each w_d

Using MLE-II Estimates for Making Prediction

- With the MLE-II approximation $p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx \delta(\hat{\beta}, \hat{\lambda})$, the posterior over unknowns

$$p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y}) = p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) \approx p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})$$

- The posterior predictive distribution can also be approximated as

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}, \beta, \lambda|\mathbf{X}, \boldsymbol{y})\; d\boldsymbol{w}\; d\beta\; d\lambda \\ &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \beta, \lambda)p(\beta, \lambda|\mathbf{X}, \boldsymbol{y}) d\beta\; d\lambda\; d\boldsymbol{w} \\ &\approx \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}, \hat{\beta}, \hat{\lambda})\; d\boldsymbol{w} \end{aligned}$$

- This is also the same as the usual posterior predictive distribution we have seen earlier, except we are treating the hyperparams $\hat{\beta}, \hat{\lambda}$ fixed at their MLE-II based estimates

Modeling Sparse Weights

Modeling Sparse Weights

- Many probabilistic models consist of weights that are given zero-mean Gaussian priors, e.g.,

$$\mu(\boldsymbol{x}) = \sum_{d=1}^{D} w_d x_d \qquad \text{(mean of a prob. lin reg model)}$$

$$\mu(\boldsymbol{x}) = \sum_{n=1}^{N} w_n k(\boldsymbol{x}_n, \boldsymbol{x}) \qquad \text{(mean of a prob. kernel based nonlin reg model)}$$

- A zero-mean prior is of the form $p(w_d) = \mathcal{N}(0, \lambda^{-1})$ or $p(w_d) = \mathcal{N}(0, \lambda_d^{-1})$
- Precision λ or λ_d specifies our belief about how close to zero w_d is (like regularization hyperparam)
- However, such a prior usually gives small weights but not very strong sparsity
- Putting a gamma prior on precision can give sparsity (will soon see why)
- Sparsity of weights will be a very useful thing to have in many models, e.g.,
  - For linear model, this helps learn relevance of each feature x_d
  - For kernel based model, this helps learn the relevance of each input $\boldsymbol{x}_n$ (Relevance Vector Machine)

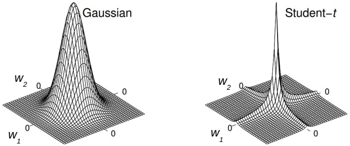
Sparsity via a Hierarchical Prior

- Consider linear regression with prior $p(w_d|\lambda_d) = \mathcal{N}(0, \lambda_d^{-1})$ on each weight
- Let's treat precision λ_d as unknown and use a gamma (shape $= a$, rate $= b$) prior on it

$$p(\lambda_d) = \text{Gamma}(a, b) = \frac{b^a}{\Gamma(a)} \lambda_d^{a-1} \exp(-b\lambda_d)$$

- Marginalizing the precision leads to a Student-t prior on each w_d

$$p(w_d) = \int p(w_d|\lambda_d)p(\lambda_d)d\lambda_d = \frac{b^a\Gamma(a+1/2)}{\sqrt{2\pi}\Gamma(a)}(b + w_d^2/2)^{-(a+1/2)}$$



- Note: Can make the prior an uninformative prior by setting a and b to be very small (e.g., 10^{-4})
- Note: Some other priors on λ_d (e.g., exponential distribution) also result in sparse priors on w_d

Bayesian Linear Regression with Sparse Prior on Weights

- Posterior inference for $\boldsymbol{w}$ not straightforward since $p(\boldsymbol{w}) = \prod_{d=1}^{D} p(w_d)$ is no longer Gaussian
- Approximate inference is usually needed for inferring the full posterior
- Many approaches exist (which we will see later)
- Such approaches are mostly in form of alternating estimation of $\boldsymbol{w}$ and λ
  - Estimate λ_d given w_d, estimate w_d given λ_d
- Popular approaches: EM, Gibbs sampling, variational inference, etc
- Working with such sparse priors is known as Sparse Bayesian Learning
  - Used in many models where we want to have sparsity in the weights (very few non-zero weights)
- Note: We will later look at other ways of getting sparsity (e.g., spike-and-slab priors defined by binary switch variables for each weight)

Bayesian Logistic Regression

(..a simple, single-parameter, yet non-conjugate model)

Probabilistic Models for Classification

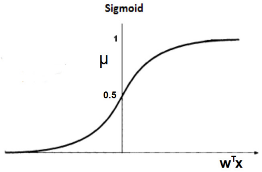
- The goal is to learn $p(y|\boldsymbol{x})$. Here $p(y|\boldsymbol{x})$ will be a discrete distribution (e.g., Bernoulli, multinoulli)
- Usually two approaches to learn $p(y|\boldsymbol{x})$: Discriminative Classification and Generative Classification
- Discriminative Classification: Model and learn $p(y|\boldsymbol{x})$ directly
  - This approach does not model the distribution of the inputs $\boldsymbol{x}$
- Generative Classification: Model and learn $p(y|\boldsymbol{x})$ "indirectly" as $p(y|\boldsymbol{x}) = \frac{p(y)p(\boldsymbol{x}|y)}{p(\boldsymbol{x})}$
  - Called generative because, via $p(\boldsymbol{x}|y)$, we model how the inputs $\boldsymbol{x}$ of each class are generated
  - The approach requires first learning class-marginal $p(y)$ and class-conditional distributions $p(\boldsymbol{x}|y)$
  - Usually harder to learn than discriminative but also has some advantages (more on this later)
- Both approaches can be given a non-Bayesian or Bayesian treatment
  - The Bayesian treatment won't rely on point estimates but infer the posterior over unknowns

Discriminative Classification via Logistic Regression

- **Logistic Regression** (LR) is an example of discriminative binary classification, i.e., $y \in \{0,1\}$
- Logistic Regression models $\boldsymbol{x}$ to y relationship using the sigmoid function

$$p(y=1|\boldsymbol{x},\boldsymbol{w}) = \mu = \sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})}$$

where $\boldsymbol{w} \in \mathbb{R}^D$ is the weight vector. Also note that $p(y=0|\boldsymbol{x},\boldsymbol{w}) = 1-\mu$



- A large positive (negative) "score" $\boldsymbol{w}^\top \boldsymbol{x}$ means large probability of label being 1 (0)
- Is sigmoid the only way to convert the score into a probability?
  - No, while LR does that, there exist models that define μ in other ways. E.g. Probit Regression

$$\mu = p(y=1|\boldsymbol{x},\boldsymbol{w}) = \Phi(\boldsymbol{w}^\top \boldsymbol{x}) \qquad \text{(where } \Phi \text{ denotes the CDF of } \mathcal{N}(0,1))$$

Logistic Regression

- The LR classification rule is

$$p(y=1|\boldsymbol{x},\boldsymbol{w}) = \mu = \sigma(\boldsymbol{w}^\top\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w}^\top\boldsymbol{x})} = \frac{\exp(\boldsymbol{w}^\top\boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})}$$

$$p(y=0|\boldsymbol{x},\boldsymbol{w}) = 1-\mu = 1-\sigma(\boldsymbol{w}^\top\boldsymbol{x}) = \frac{1}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})}$$

- This implies a Bernoulli likelihood model for the labels

$$p(y|\boldsymbol{x},\boldsymbol{w}) = \text{Bernoulli}(\sigma(\boldsymbol{w}^\top\boldsymbol{x})) = \left[\frac{\exp(\boldsymbol{w}^\top\boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})}\right]^y \left[\frac{1}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})}\right]^{(1-y)}$$

- Given N observations $(\mathbf{X},\boldsymbol{y}) = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, we can do point estimation for $\boldsymbol{w}$ by maximizing the log-likelihood (or minimizing the negative log-likelihood). This is basically MLE.

$$\boldsymbol{w}_{MLE} = \arg\max_{\boldsymbol{w}} \sum_{n=1}^N \log p(y_n|\boldsymbol{x}_n,\boldsymbol{w}) = \arg\min_{\boldsymbol{w}} -\sum_{n=1}^N \log p(y_n|\boldsymbol{x}_n,\boldsymbol{w}) = \arg\min_{\boldsymbol{w}} NLL(\boldsymbol{w})$$

- Convex loss function. Global minima. Both first order and second order methods widely used.
  - Can also add a regularizer on $\boldsymbol{w}$ to prevent overfitting. This corresponds to doing MAP estimation with a prior on $\boldsymbol{w}$, i.e., $\boldsymbol{w}_{MAP} = \arg\max_{\boldsymbol{w}}[\sum_{n=1}^N \log p(y_n|\boldsymbol{x}_n,\boldsymbol{w}) + \log p(\boldsymbol{w})]$

Bayesian Logistic Regression

- MLE/MAP only gives a point estimate. We would like to infer the full posterior over $\boldsymbol{w}$
- Recall that the likelihood model is Bernoulli

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = \text{Bernoulli}(\sigma(\boldsymbol{w}^\top \boldsymbol{x})) = \left[\frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})}\right]^y \left[\frac{1}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})}\right]^{(1-y)}$$

- Just like the Bayesian linear regression case, let's use a Gausian prior on $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I}_D) \propto \exp(-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w})$$

- Given N observations $(\mathbf{X}, \boldsymbol{y}) = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where $\mathbf{X}$ is $N \times D$ and $\boldsymbol{y}$ is $N \times 1$, the posterior over $\boldsymbol{w}$

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}} = \frac{\prod_{n=1}^N p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})}{\int \prod_{n=1}^N p(y_n|\boldsymbol{x}_n, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}$$

- The denominator is intractable in general (logistic-Bernoulli and Gaussian are not conjugate)
  - Can't get a closed form expression for $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$. Must approximate it!
  - Several ways to do it, e.g., MCMC, variational inference, Laplace approximation (next class)

Next Class

- Laplace approximation
- Computing posterior and posterior predictive for logistic regression
- Properties/benefits of Bayesian logistic regression
- Bayesian approach to generative classification