Laplace Approximation (Contd) and Generalized Linear Models

CS772A: Probabilistic Machine Learning
Piyush Rai
Logistic Regression

- A discriminative model for binary classification \((y \in \{0,1\})\)
- A linear model with parameters \(w \in \mathbb{R}^D\) computes a score \(w^T x\) for input \(x\)
- A sigmoid function maps this real-valued score into probability of label being 1

\[
p(y = 1|x, w) = \mu = \sigma(w^T x)
\]

- Thus conditional distribution of label \(y \in \{0,1\}\) given \(x\) is the following Bernoulli

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

- Can use a Gaussian prior on \(w\): \(p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1}I)\)
- Point estimation (MLE/MAP) for LR gives global optima (NLL is convex in \(w\))
- We will mainly focus on fully Bayesian inference (computing the posterior)

\[
p(y = 1|x, w) = \mu = \Phi(w^T x)\text{ where }\Phi\text{ is the CDF of }\mathcal{N}(0,1).\text{ This model is known as "Probit Regression".}
\]

Likelihood

There are other ways too that can convert the score into a probability, such as a CDF: \(p(y = 1|x, w) = \mu = \Phi(w^T x)\) where \(\Phi\) is the CDF of \(\mathcal{N}(0,1)\). This model is known as “Probit Regression”.

Large positive score \(w^T x\) means large prob of label being 1, and large negative score means low prob

Can also use a sparsity-inducing prior, such as spike-and-slab or a scale-mixture of Gaussians
Logistic Regression: The Posterior

- The posterior will be

\[
p(w|X, y) = \frac{p(w)p(y|X, w)}{p(y|X)} = \frac{\prod_{n=1}^{N} p(y_n|w, x_n)}{\int p(w) \prod_{n=1}^{N} p(y_n|w, x_n) \, dw}
\]

- Need to approximate the posterior in this case

- For now, we will use a simple approximation called Laplace approximation

Laplace approx: Approximates the intractable posterior by a Gaussian whose mean is the MAP solution of the LR model. .. and the covariance matrix of this Gaussian is set to the inverse of the Hessian matrix (second derivative) of the model’s negative log-joint of params and data, evaluated at the MAP solution.

Unfortunately, Gaussian and Bernoulli are not conjugate with each other, so analytic expression for the posterior can’t be obtained unlike prob. linear regression.

Hyperparam \(\lambda\) not shown.

Other approx. inference methods, such as MCMC and VI later.

First or second-order optimization methods can be used.

Gaussian

Bernoulli

w_{MAP} = \arg\max_w \log p(w|y, X)
= \arg\max_w \log p(y, w|X)
= \arg\min_w [-\log p(y|w, X)]

w_{MAP} = \arg\max_w \log p(w|y, X)
= \arg\max_w \log p(y, w|X)
= \arg\min_w [-\log p(y|w, X)]
LR Posterior: An Illustration

- Assuming the Gaussian approximation, some samples from the posterior of LR

- Each sample drawn from $p(w|X, y)$ will give a weight vector

- Each such $w$ corresponds to one of the separators in the above figure

Not all separators from the posterior are equally good; their “goodness” will depend on their posterior probabilities $p(w|X, y)$.

When making predictions, we can still use all of them but weighted by their importance based on their posterior probabilities.

That’s exactly what we do when computing the predictive distribution.
The posterior predictive distribution can be computed as

\[ p(y_* = 1|x_*, X, y) = \int p(y_* = 1|w, x_*)p(w|X, y)dw \]

Monte-Carlo approximation of this integral is one possible way

- Draw \( M \) samples \( w_1, w_2, ..., w_M \), from the approx. of posterior
- Approximate the PPD as follows

\[ p(y_* = 1|x_*, X, y) \approx \frac{1}{M} \sum_{m=1}^{M} p(y_* = 1|w_m, x_*) = \frac{1}{M} \sum_{m=1}^{M} \sigma(w_m^T x_n) \]

In contrast, when using MLE/MAP solution \( \hat{w}_{opt} \), the plug-in pred. distribution

\[ p(y_* = 1|x_*, X, y) = \int p(y_* = 1|w, x_*)p(w|X, y)dw \]

\[ \approx p(y_* = 1|\hat{w}_{opt}, x_*) = \sigma(\hat{w}_{opt}^T x_n) \]
LR: Plug-in Prediction vs Bayesian Averaging

- Plug-in prediction uses a single \( w \) (point est) to make prediction
- PPD does an averaging using all possible \( w \)'s from the posterior

\[
p(y_* = 1|x_*, X, y) \approx \sigma(\hat{w}_{opt}^\top x_n)
\]

\[
p(y_* = 1|x_*, X, y) \approx \frac{1}{M} \sum_{m=1}^{M} \sigma(w_m^\top x_n)
\]
Multiclass Logistic (a.k.a. Softmax) Regression

- Also called multinoulli/multinomial regression: Basically, LR for $K > 2$ classes
- In this case, $y_n \in \{1, 2, \ldots, K\}$ and label probabilities are defined as

\[
p(y_n = k | x_n, W) = \frac{\exp(w_k^T x_n)}{\sum_{k=1}^{K} \exp(w_k^T x_n)} = \mu_{nk}
\]

- $K$ weight vecs $w_1, w_2, \ldots, w_K$ (one per class), each $D$-dim, and $W = [w_1, w_2, \ldots, w_K]$
- Each likelihood $p(y_n | x_n, W)$ is a multinoulli distribution. Therefore total likelihood

\[
p(y | X, W) = \prod_{n=1}^{N} \prod_{\ell=1}^{K} \mu_{n\ell}^{y_{n\ell}}
\]

- Can do MLE/MAP/fully Bayesian estimation for $W$ similar to LR model
Laplace Approximation of Posterior Distribution

- Consider a posterior distribution that is intractable to compute

\[ p(\theta | D) = \frac{p(D | \theta) p(\theta)}{p(D)} = \frac{p(D, \theta)}{p(D)} \]

- Laplace approximation approximates the above using a Gaussian distribution

\[ \theta_{MAP} = \arg \max_\theta p(\theta | D) = \arg \max_\theta p(D, \theta) = \arg \max_\theta p(D | \theta) p(\theta) = \arg \max_\theta \left[ \log p(D | \theta) + \log p(\theta) \right] \]

\[ H = -\nabla^2 \log p(D | \theta) \big|_{\theta = \theta_{MAP}} = -\nabla^2 \log p(D, \theta) \big|_{\theta = \theta_{MAP}} = -\nabla^2 \left[ \log p(D | \theta) + \log p(\theta) \right] \big|_{\theta = \theta_{MAP}} \]

- Why is the above Gaussian a reasonable approximation to the posterior?
Derivation of the Laplace Approximation

- Let's write the Bayes rule as
  
  \[
  p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}, \theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}, \theta)}{\int p(\mathcal{D}, \theta) d\theta} = \frac{e^{\log p(\mathcal{D}, \theta)}}{\int e^{\log p(\mathcal{D}, \theta)} d\theta}
  \]

- Approximating \( \log p(\mathcal{D}, \theta) \) by a quadratic function of \( \theta \) will make it a Gaussian.

- Consider the second-order Taylor approx of a function \( f(\theta) \) around some \( \theta_0 \)

  \[
  f(\theta) \approx f(\theta_0) + (\theta - \theta_0)^T \nabla f(\theta_0) + \frac{1}{2}(\theta - \theta_0)^T \nabla^2 f(\theta_0)(\theta - \theta_0)
  \]

- Assuming \( f(\theta) = \log p(\mathcal{D}, \theta) \) and \( \theta_0 = \theta_{MAP}, \nabla f(\theta_{MAP}) = \nabla \log p(\mathcal{D}, \theta_{MAP}) = 0 \)

  \[
  \log p(\mathcal{D}, \theta) \approx \log p(\mathcal{D}, \theta_{MAP}) + \frac{1}{2}(\theta - \theta_{MAP})^T \nabla^2 \log p(\mathcal{D}, \theta_{MAP})(\theta - \theta_{MAP})
  \]

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

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

Aha! This is a Gaussian!

Comparing with a Gaussian PDF
Mean = \( \theta_{MAP} \)
Cov. Matrix = \( H^{-1} \)
Properties of Laplace Approximation

- Usually straightforward if derivatives (first and second) can be computed easily
- Expensive if parameter $\theta$ is very high dimensional
  - 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 $\theta$ is real-valued (won’t if, say, it is positive, binary etc)
- Note: Even if we have a non-probabilistic 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

\[ p(\theta|D) \approx \sum_{k=1}^{K} \pi(k) N(\theta|\mu_{MAP}^{(k)}, H^{(k)^{-1}}) \]

*Mixtures of Laplace Approximations for Improved Post-Hoc Uncertainty in Deep Learning (Eschenhagen et al, 2021)
Laplace Approx. for High-Dimensional Problems

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

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

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

Assuming a discriminative model with parameters $\theta$

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

Fisher Information Matrix (FIM)

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

KFAC paper: “A Scalable Laplace Approximation for Neural Networks” (Ritter et al, ICLR 2018)
Generalized Linear Models

- (Probabilistic) Linear Regression: when response $y$ is real-valued
  \[ p(y|x, w) = \mathcal{N}(w^T x, \beta^{-1}) \]
- Logistic Regression: when response $y$ is binary (0/1)
  \[ p(y|x, w) = \text{Bernoulli}(\sigma(w^T x)) = [\sigma(w^T x)]^y[1 - \sigma(w^T x)]^{1-y} \]
  \[ \sigma(w^T x) = \frac{1}{1 + \exp(-w^T x)} = \frac{\exp(w^T x)}{1 + \exp(w^T x)} \]
- In both, the model depends on the inputs $x$ via a linear model $w^T x$
- Generalized Linear Models (GLM) allow modeling other types of responses, e.g.,
  - Counts (e.g., predicting the hourly hits on a website)
  - Positive reals (e.g., predicting depth of different pixels in a scene, or stock prices)
  - Fractions between 0 and 1 (e.g., predicting proportion of crude oil convertible to gasoline)
- Note: Can convert responses to real values and apply standard regression, but it is better to model them directly (e.g., for better interpretability of the model)
Generalized Linear Models: Formally

- GLMs model the response using an exponential family distribution

\[
p(y|\eta) = h(y) \exp(\eta y - A(\eta))
\]

- The inputs \( x \) only appear via a linear model \( \xi = w^T x \) and the overall pipeline is

  \[
  w \xrightarrow{\mu} \xi = w^T x \xrightarrow{\eta} \psi(\mu)
  \]

- Note: Some GLMs are represented via exponential **dispersion** family given by

  \[
p(y|\eta, \sigma^2) = h(y, \sigma^2) \exp \left[ \frac{\eta y - A(\eta)}{\sigma^2} \right]
  \]

  \[
  \mathbb{E}[y] = A'(\eta) \\
  \text{var}[y] = A''(\eta) \sigma^2
  \]
Generalized Linear Models: Examples

- Consider the overdispersed GLMs

\[
p(y|\eta, \sigma^2) = h(y, \sigma^2) \exp \left[ \frac{\eta y - A(\eta)}{\sigma^2} \right] = \exp \left[ \frac{\eta y - A(\eta)}{\sigma^2} + \log h(y, \sigma^2) \right]
\]

- Consider a linear regression model with Gaussian likelihood

\[
p(y|x, w, \sigma^2) \propto \exp \left[ - \frac{(y - w^T x)^2}{2\sigma^2} \right] = \exp \left[ - \frac{y^2 + (w^T x)^2 - 2yw^T x}{2\sigma^2} \right] = \exp \left[ \frac{yw^T x - (w^T x)^2/2}{\sigma^2} \right] - \frac{y^2}{2\sigma^2}
\]

- Comparing the expressions, \( \eta = w^T x, A(\eta) = \frac{\eta^2}{2}, \log h(y, \sigma^2) = -\frac{y^2}{2\sigma^2} \)

- Can likewise express other models for exp-family distributions \( p(y|x) \)
  - Regardless of the form, all will have \( \eta = w^T x \)
GLM with Canonical Response Function

- For GLM with Canon Resp Func (a.k.a., canonical GLM)

\[ p(y|\eta) = h(y) \exp(\eta y - A(\eta)) = h(y) \exp(y w^T x - A(\eta)) \]

- Consider doing MLE (assuming \( N \) i.i.d. responses). The log likelihood

\[ L(\eta) = \log p(Y|\eta) = \log \prod_{n=1}^{N} h(y_n) \exp(y_n w^T x_n - A(\eta_n)) = \sum_{n=1}^{N} \log h(y_n) + w^T \sum_{n=1}^{N} y_n x_n - \sum_{n=1}^{N} A(\eta_n) \]

- Convexity of \( A(\eta) \) guarantees a global optima.

\[ g = \sum_{n=1}^{N} \left( y_n x_n - A'(\eta_n) \frac{d\eta_n}{dw} \right) = \sum_{n=1}^{N} (y_n x_n - \mu_n x_n) = \sum_{n=1}^{N} (y_n - \mu_n)x_n \]

- Note \( \mu_n = f(\xi_n) = f(w^T x_n) \) and \( f = \psi^{-1} \) (“inverse link”) depends on the model

  - Real-valued \( y \) (linear regression): \( f \) is identity, i.e., \( \mu_n = w^T x_n \)
  - Binary \( y \) (logistic regression): \( f \) is sigmoid function, i.e., \( \mu_n = \frac{\exp(w^T x_n)}{1+\exp(w^T x_n)} \)
  - Count-valued \( y \) (Poisson regression): \( f \) is exp, i.e., \( \mu_n = \exp(w^T x_n) \)
  - Non-negative \( y \) (gamma regression): \( f \) is inverse negative i.e., \( \mu_n = -1/(w^T x_n) \)

The simple form of canonical GLM (nat. param just a linear function \( w^T x \)) makes parameter estimation via MLE/MAP easy since gradient and Hessian have simple expressions (though the Hessian may be expensive to compute/invert).
Fully Bayesian Inference for GLMs

- Most GLMs, except linear regression with Gaussian lik. and Gaussian prior, do not have conjugate pairs of likelihood and priors (recall logistic regression)

- Posterior over the weight vector $\mathbf{w}$ is intractable

- Approximate inference methods needed, e.g.,
  - Laplace approximation (have already seen): Easily applicable since derivatives (first and second) can be easily computed (note that we need $\mathbf{w}_{\text{MAP}}$ and Hessian)
  - MCMC or variational inference (will see later)
Various Types of GLMs

<table>
<thead>
<tr>
<th>Type of response</th>
<th>Type of GLM</th>
<th>Link Function $\Psi$ (Inv Link Func if canon. GLM) (Operates on $\xi = w^T x$)</th>
<th>Response Function $f$</th>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Gaussian</td>
<td>Identity</td>
<td>Identity</td>
<td>Linear</td>
</tr>
<tr>
<td>Binary</td>
<td>Logistic</td>
<td>Log-odds: $\log \frac{\mu}{1-\mu}$</td>
<td>Sigmoid</td>
<td>Sigmoid</td>
</tr>
<tr>
<td>Binary</td>
<td>Probit</td>
<td>Inv CDF: $\Phi^{-1}(\mu)$</td>
<td>$\Phi$ (CDF of N(0,1))</td>
<td>Probit</td>
</tr>
<tr>
<td>Categorical</td>
<td>Multinoulli</td>
<td>Log-odds: $\log \frac{\mu_k}{1-\mu_k}$</td>
<td>Softmax</td>
<td>Softmax</td>
</tr>
<tr>
<td>Count</td>
<td>Poisson</td>
<td>$\log \mu$</td>
<td>exp</td>
<td></td>
</tr>
<tr>
<td>Non-negative real</td>
<td>gamma</td>
<td>Negative of inverse</td>
<td>Negative of inverse</td>
<td></td>
</tr>
<tr>
<td>Binary</td>
<td>Gumbel</td>
<td>Gumbel Inv CDF: $\log(-\log())$</td>
<td>Gumbel CDF: $\exp(-\exp(-))$</td>
<td></td>
</tr>
</tbody>
</table>

.. and several others (exponential, inverse Gaussian, Binomial, Tweedie, etc)