Laplace Approximation (Contd) and Generalized Linear Models

CS772A: Probabilistic Machine Learning

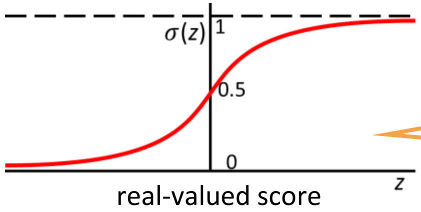
Piyush Rai

Logistic Regression

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

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

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



$$\sigma(z) = \frac{1}{1+\exp(-z)} = \frac{\exp(z)}{1+\exp(z)}$$

Large positive score $\boldsymbol{w}^\top \boldsymbol{x}$ means large prob of label being 1, and large negative score means low prob

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

Likelihood

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

- Can use a Gaussian prior on $\boldsymbol{w}$: $p(\boldsymbol{w}|\lambda) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0},\lambda^{-1}\boldsymbol{I})$

Can also use a sparsity-inducing prior, such as spike-and-slab or a scale-mixture of Gaussians

- Point estimation (MLE/MAP) for LR gives global optima (NLL is convex in $\boldsymbol{w}$)
- We will mainly focus on fully Bayesian inference (computing the posterior)

Logistic Regression: The Posterior

$$\begin{aligned} \mathbf{g} &= -\sum_{n=1}^{N}(y_n-\mu_n)\mathbf{x}_n + \lambda \mathbf{I}\mathbf{w} = \mathbf{X}^\top(\boldsymbol{\mu}-\mathbf{y}) + \lambda \mathbf{w} \quad \text{(a } D\times 1 \text{ vector)} \\ \mathbf{H} &= \sum_{n=1}^{N}\mu_n(1-\mu_n)\mathbf{x}_n\mathbf{x}_n^\top + \lambda \mathbf{I} = \mathbf{X}^\top \mathbf{S}\mathbf{X} + \lambda \mathbf{I} \quad \text{(a } D\times D \text{ matrix)} \end{aligned}$$

$$\mu_n = \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)$$

- The posterior will be

Gaussian

Bernoulli

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

Hyperparam λ not shown

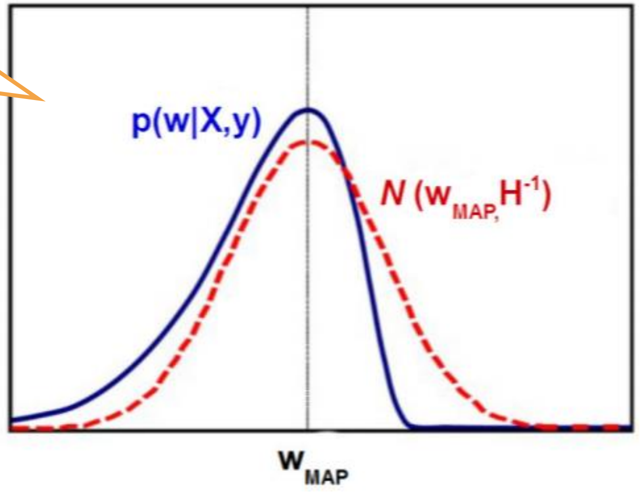
Unfortunately, Gaussian and Bernoulli are not conjugate with each other, so analytic expression for the posterior can't be obtained unlike prob. linear regression

- Need to approximate the posterior in this case
- For now, we will use a simple approximation called Laplace approximation

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

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



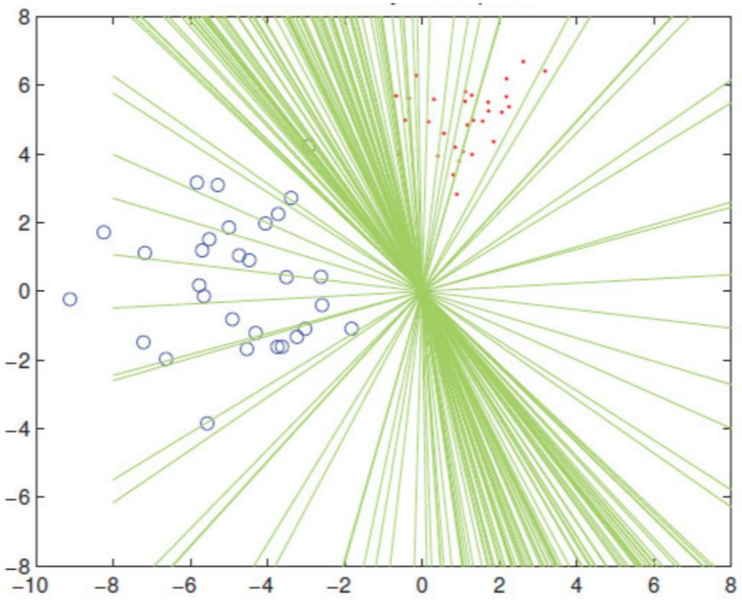
$$\begin{aligned} \boldsymbol{w}_{MAP} &= \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{w}|\boldsymbol{y},\mathbf{X}) \\ &= \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{y},\boldsymbol{w}|\mathbf{X}) \\ &= \arg\min_{\boldsymbol{w}} [-\log p(\boldsymbol{y},\boldsymbol{w}|\mathbf{X})] \end{aligned}$$

First or second-order optimization methods can be used

$$\mathbf{H} = \nabla^2[-\log p(\boldsymbol{y},\boldsymbol{w}|\mathbf{X})]\big|_{\boldsymbol{w}=\boldsymbol{w}_{MAP}}$$

LR Posterior: An Illustration

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



Not all separators from from the posterior are equally good; their "goodness" will depends on their posterior probabilities $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})$

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

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

- Each such $\boldsymbol{w}$ corresponds to one of the separators in the above figure

LR: Posterior Predictive Distribution

- The posterior predictive distribution can be computed as

$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y_* = 1|\boldsymbol{w}, \boldsymbol{x}_*)p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})d\boldsymbol{w}$$

Integral not tractable and must be approximated

sigmoid

Gaussian (if using Laplace approx.)

- Monte-Carlo approximation of this integral is one possible way
  - Draw $\boldsymbol{M}$ samples $\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_M$, from the approx. of posterior
  - Approximate the PPD as follows

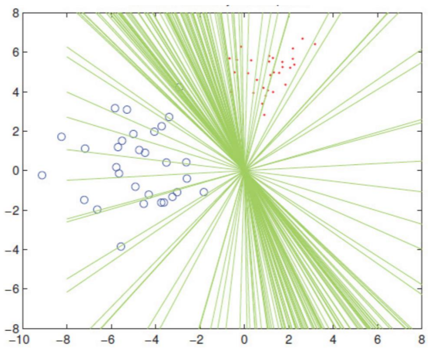
$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) \approx \frac{1}{M}\sum_{m=1}^{M} p(y_* = 1|\boldsymbol{w}_{\mathrm{m}}, \boldsymbol{x}_*) = \frac{1}{M}\sum_{m=1}^{M} \sigma(\boldsymbol{w}_m^\top \boldsymbol{x}_n)$$

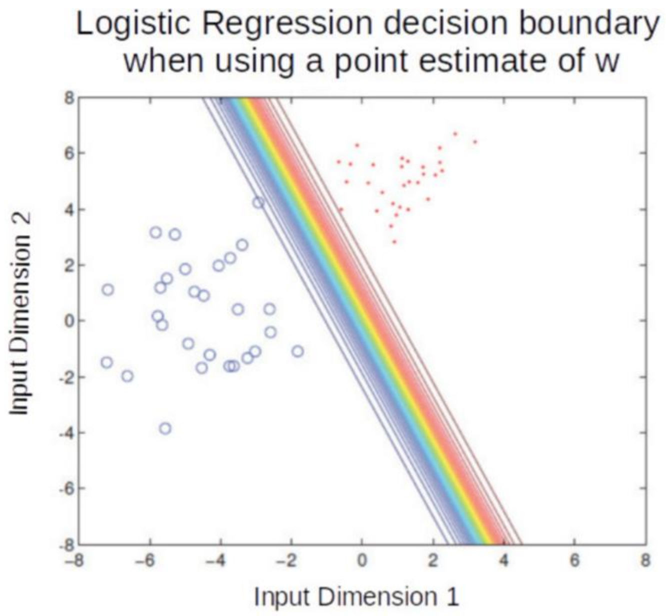
- In contrast, when using MLE/MAP solution $\hat{\boldsymbol{w}}_{opt}$, the plug-in pred. distribution

$$\begin{aligned} p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) &= \int p(y_* = 1|\boldsymbol{w}, \boldsymbol{x}_*)p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})d\boldsymbol{w} \\ &\approx p\left(y_* = 1\middle|\hat{\boldsymbol{w}}_{opt}, \boldsymbol{x}_*\right) = \sigma(\hat{\boldsymbol{w}}_{opt}{}^\top \boldsymbol{x}_n) \end{aligned}$$

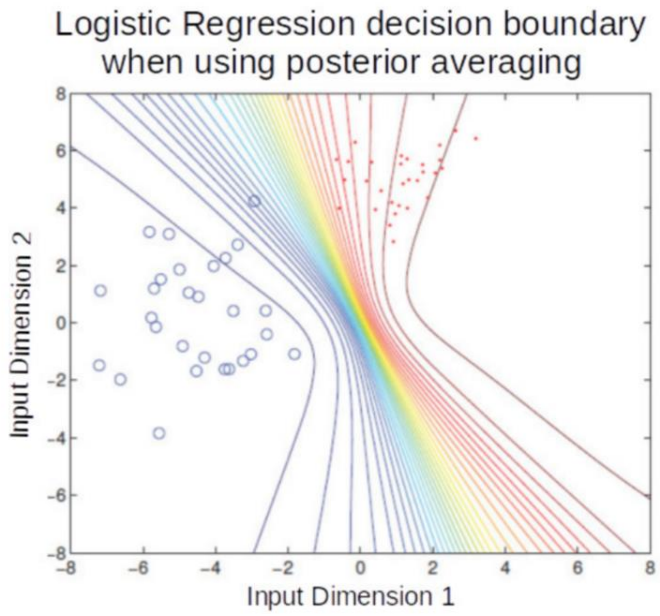
LR: Plug-in Prediction vs Bayesian Averaging

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





Color transitions (red to blue) in both plots denote how the probability of an input changes from belonging to red class to belonging to blue class. All inputs on a line (or curve on RHS plot) have the same probability of belonging to the red/blue class



Posterior averaging is like using an ensemble of models. In this example, each model is a linear classifier but the ensemble-like effect resulted in nonlinear boundaries

$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) \approx \sigma(\hat{\boldsymbol{w}}_{opt}{}^\top \boldsymbol{x}_n)$$

$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) \approx \frac{1}{M}\sum_{m=1}^{M} \sigma(\boldsymbol{w}_m^\top \boldsymbol{x}_n)$$

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

Softmax function

$$p(y_n = k|\boldsymbol{x}_n, \boldsymbol{W}) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K}\exp(\boldsymbol{w}_\ell^\top \boldsymbol{x}_n)} = \mu_{nk}$$

Also note that $\sum_{\ell=1}^{K}\mu_{n\ell} = 1$ for any input $\boldsymbol{x}_n$

- K weight vecs $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ (one per class), each D-dim, and $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K]$
- Each likelihood $p(y_n|\boldsymbol{x}_n, \boldsymbol{W})$ is a multinoulli distribution. Therefore total likelihood

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

Notation: $y_{n\ell} = 1$ if true class of $\boldsymbol{x}_n$ is ℓ and $y_{n\ell'} = 0 \; \forall \ell' \neq \ell$

- Can do MLE/MAP/fully Bayesian estimation for $\boldsymbol{W}$ similar to LR model

Laplace Approximation of Posterior Distribution

- Consider a posterior distribution that is intractable to compute

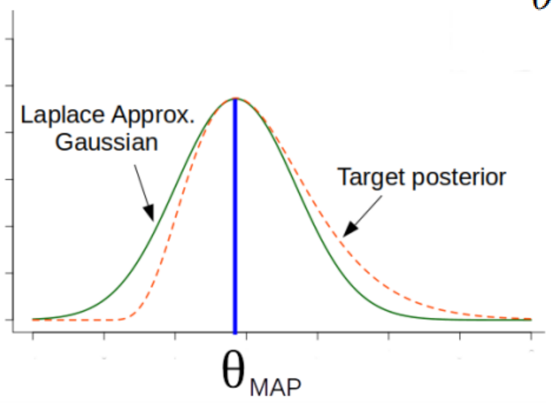
Unknowns of the model

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

$-\mathbf{H}$

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

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

Aha! This is a Gaussian!

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

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

Constant w.r.t. $\boldsymbol{\theta}$

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

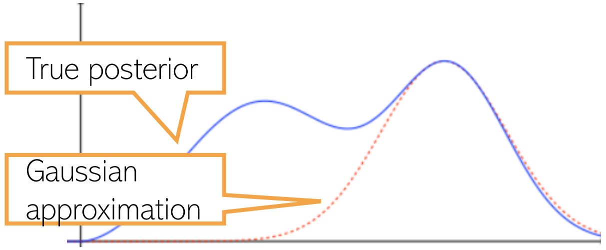
- Thus Laplace approx. is based on a second-order Taylor approx. of the posterior

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



For multimodal posteriors, can use a mixture of Laplace approximation*

Useful for deep learning models

If K local modes, then define the approx. posterior as a mixture of K Gaussians

$$p(\theta|D) \approx \sum_{k=1}^{K} \pi^{(k)} \mathcal{N}(\theta|\theta_{MAP}^{(k)}, H^{(k)^{-1}})$$

(see paper cited below for details)

- Applicable only when $\boldsymbol{\theta}$ 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

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

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

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

Fisher Information Matrix (FIM)

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

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

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

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

- 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)

Generalized Linear Models

- (Probabilistic) Linear Regression: when response y is real-valued

$$p(y|\mathbf{x},\mathbf{w}) = \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \beta^{-1})$$

- Logistic Regression: when response y is binary (0/1)

$$p(y|\mathbf{x},\mathbf{w}) = \text{Bernoulli}(\sigma(\mathbf{w}^\top \mathbf{x})) = [\sigma(\mathbf{w}^\top \mathbf{x})]^y [1-\sigma(\mathbf{w}^\top \mathbf{x})]^{1-y}$$

$$\sigma(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{x})} = \frac{\exp(\mathbf{w}^\top \mathbf{x})}{1+\exp(\mathbf{w}^\top \mathbf{x})}$$

- In both, the model depends on the inputs $\boldsymbol{x}$ via a linear model $\boldsymbol{w}^\top \boldsymbol{x}$

Note: Probabilistic Linear Regression and Logistic Regression are also GLMs

- 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

The reason why GLMs can model a wide variety of responses

- GLMs model the response using an exponential family distribution

Response $\boldsymbol{y}$ assumed univariate but vector GLMs also exist

Scalar natural param (depends on input $\boldsymbol{x}$)

Scalar suff-stats $\phi(y) = y$

$$\boxed{p(y|\eta) = h(y)\exp(\eta y - A(\eta))}$$

- The inputs $\boldsymbol{x}$ only appear via a linear model $\xi = \boldsymbol{w}^\top \boldsymbol{x}$ and the overall pipeline is

For prob. linear regression with Gaussian lik, f is identify since mean $\mu = \mathbb{E}[y|\boldsymbol{x}] = \boldsymbol{w}^\top \boldsymbol{x}$

For logistic regression, f is sigmoid since mean $\mu = \mathbb{E}[y|\boldsymbol{x}] = \sigma(\boldsymbol{w}^\top \boldsymbol{x})$

Response Function

Link Function

$$\mathrm{w},\ x \rightarrow \xi \xrightarrow{f} \mu \xrightarrow{\psi} \eta$$

Conditional mean of response $\mu = \mathbb{E}[y|\boldsymbol{x}] = f(\xi) = f(\boldsymbol{w}^\top \boldsymbol{x})$

Natural parameter $\eta = \psi(\mu)$

For GLM with Canonical Response Function (next slide), $\psi = f^{-1}$ and thus nat. param. $\eta = \xi = \boldsymbol{w}^\top \boldsymbol{x}$

f known as "inverse link function" in this case

- Note: Some GLM are represented via exponential <u>dispersion</u> family given by

If σ^2 is fixed, it is the standard exponential family

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

Called the "dispersion parameter"

Examples: Gaussian GLM, Gamma GLM

Recall cumulant results of exp-fam

$\mathbb{E}[y] = A'(\eta)$

$\mathrm{var}[y] = A''(\eta)\sigma^2$

Generalized Linear Models: Examples

- Consider the overdispersed GLMs

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

Note that here we expressed the Gaussian in the overdispersed GLM form unlike how we did it earlier when discussing exp-family

- Consider a linear regression model with Gaussian likelihood

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

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

- Can likewise express other models for exp-family distributions $p(y|\boldsymbol{x})$
  - Regardless of the form, all will have $\eta = \boldsymbol{w}^\top\boldsymbol{x}$

GLM with Canonical Response Function

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

$$p(y|\eta) \;=\; h(y)\exp(\eta y - A(\eta)) = \; h(y)\exp(y\mathbf{w}^\top \mathbf{x} - A(\eta))$$

The simple form of canonical GLM (nat. param just a linear function $\boldsymbol{w}^\top\boldsymbol{x}$) makes parameter estimation via MLE/MAP easy since gradient and Hessian have simple expressions (though the Hessian may be expensive to compute/invert)

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

$$L(\eta) = \log p(Y|\eta) = \log \prod_{n=1}^{N} h(y_n)\exp(y_n \mathbf{w}^\top \mathbf{x}_n - A(\eta_n)) \quad = \sum_{n=1}^{N} \log h(y_n) + \mathbf{w}^\top \sum_{n=1}^{N} y_n \mathbf{x}_n - \sum_{n=1}^{N} A(\eta_n)$$

- Convexity of $A(\eta)$ guarantees a global optima. Gradient of log-lik w.r.t. $\boldsymbol{w}$

Exp of suff-stats $\mathbb{E}[y_n]$

Corrective updates for $\boldsymbol{w}$

$$\mathbf{g} = \sum_{n=1}^{N}\left(y_n \mathbf{x}_n - A'(\eta_n)\frac{d\eta_n}{d\mathbf{w}}\right) = \sum_{n=1}^{N}(y_n\mathbf{x}_n - \mu_n \mathbf{x}_n) = \sum_{n=1}^{N}(y_n - \mu_n)\mathbf{x}_n$$

The Hessian can also be shown to be

$$\mathbf{H} = -\nabla\mathbf{g} = \sum_{n=1}^{N} f'(\eta_n)\boldsymbol{x}_n\boldsymbol{x}_n^\top$$

- Note $\mu_n = f(\xi_n) = f(\boldsymbol{w}^\top \boldsymbol{x}_n)$ and $f = \psi^{-1}$ ("inverse link") depends on the model
  - Real-valued y (linear regression): f is identity, i.e., $\mu_n = \boldsymbol{w}^\top\boldsymbol{x}_n$
  - Binary y (logistic regression): f is sigmoid function, i.e., $\mu_n = \frac{\exp(\boldsymbol{w}^\top\boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x}_n)}$
  - Count-valued y (Poisson regression): f is exp, i.e., $\mu_n = \exp(\boldsymbol{w}^\top\boldsymbol{x}_n)$
  - Non-negative y (gamma regression): f is inverse negative i.e., $\mu_n = -1/(\boldsymbol{w}^\top\boldsymbol{x}_n)$

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 $\boldsymbol{w}$ 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 $\boldsymbol{w}_{MAP}$ and Hessian)
  - MCMC or variational inference (will see later)

Various Types of GLMs

$$
\begin{tikzcd}
w \arrow[dr] & & & \\
& \xi \arrow[r, "f"] & \mu \arrow[r, "\Psi"] & \eta \\
x \arrow[ur] & & &
\end{tikzcd}
$$

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

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