Classification: (Bayesian) Logistic Regression (and our first tryst with non-conjugacy!)

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

Jan 25, 2018

Probabilistic Models for Classification

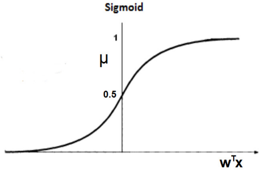
- The goal is to learn $p(y|\boldsymbol{x})$. Here $p(y|\boldsymbol{x})$ will be a discrete distribution (e.g., Bernoulli, multinoulli)
- Usually two approaches to learn $p(y|\boldsymbol{x})$: Discriminative Classification and Generative Classification
- Discriminative Classification: Model and learn $p(y|\boldsymbol{x})$ 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 $\boldsymbol{w}$

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

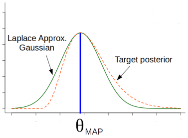
- The denominator is intractable in general (logistic-Bernoulli and Gaussian are not conjugate)
  - Can't get a closed form expression for $p(\boldsymbol{w}|\mathbf{X},\boldsymbol{y})$. Must approximate it!
  - Several ways to do it, e.g., MCMC, variational inference, Laplace approximation (today)

Laplace Approximation
(Our first posterior approximation method)

Laplace Approximation of Posterior Distribution

- Approximate the posterior distribution $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D},\theta)}{p(\mathcal{D})}$ by the following Gaussian

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta_{MAP}, \mathbf{H}^{-1})$$



- Note: θ_{MAP} is the maximum-a-posteriori (MAP) estimate of θ, i.e.,

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

- Usually θ_{MAP} can be easily solved for (e.g., using first/second order iterative methods)
- $\mathbf{H}$ is the Hessian matrix of the negative log-posterior (or negative log-joint-prob) at θ_{MAP}

$$\mathbf{H} = -\nabla^2 \log p(\theta|\mathcal{D})\Big|_{\theta=\theta_{MAP}} = -\nabla^2 \log p(\mathcal{D},\theta)\Big|_{\theta=\theta_{MAP}} = -\nabla^2[\log p(\mathcal{D}|\theta) + \log p(\theta)]\Big|_{\theta=\theta_{MAP}}$$

Derivation of the Laplace Approximation

- Let's write the Bayes rule as

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D},\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D},\theta)}{\int p(\mathcal{D},\theta)d\theta} = \frac{e^{\log p(\mathcal{D},\theta)}}{\int e^{\log p(\mathcal{D},\theta)}d\theta}$$

- Suppose $\log p(\mathcal{D},\theta) = f(\theta)$. Let's approximate $f(\theta)$ using its 2nd order Taylor expansion

$$f(\theta) \approx f(\theta_0) + (\theta-\theta_0)^\top \nabla f(\theta_0) + \frac{1}{2}(\theta-\theta_0)^\top \nabla^2 f(\theta_0)(\theta-\theta_0)$$

where θ_0 is some arbitrarily chosen point in the domain of f

- Let's choose $\theta_0 = \theta_{MAP}$. Note that $\nabla f(\theta_{MAP}) = \nabla \log p(\mathcal{D},\theta_{MAP}) = 0$. Therefore

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

Derivation of the Laplace Approximation

- Plugging in this 2nd order Taylor approximation for $\log p(\mathcal{D},\theta)$, we have

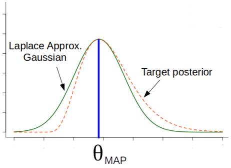
$$p(\theta|\mathcal{D}) = \frac{e^{\log p(\mathcal{D},\theta)}}{\int e^{\log p(\mathcal{D},\theta)} d\theta} \approx \frac{e^{\log p(\mathcal{D},\theta_{MAP}) + \frac{1}{2}(\theta-\theta_{MAP})^\top \nabla^2 \log p(\mathcal{D},\theta_{MAP})(\theta-\theta_{MAP})}}{\int e^{\log p(\mathcal{D},\theta_{MAP})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

- Further simplifying, we have

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

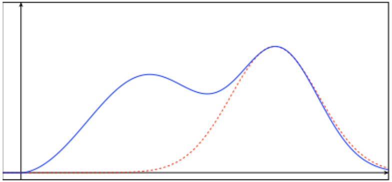
- Therefore the Laplace approximation of the posterior $p(\theta|\mathcal{D})$ is a Gaussian and is given by

$$\boxed{p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta|\theta_{MAP}, \mathbf{H}^{-1})} \qquad \text{where} \quad \mathbf{H} = -\nabla^2 \log p(\mathcal{D},\theta_{MAP})$$



Properties of Laplace Approximation

- Usually straightforward if derivatives (first and second) can be computed easily
- Expensive if the number of parameters is very large (due to Hessian computation and inversion)
- Can do badly if the (true) posterior is multimodal



- Can actually apply it when working with any regularized loss function (not just probabilistic models) to get something like a posterior distribution over the parameters
  - negative log-likelihood (NLL) $=$ loss function, negative log-prior $=$ regularizer

Laplace Approximation for Bayesian Logistic Regression

- Data $\mathcal{D} = (\mathbf{X}, \boldsymbol{y})$ and parameter $\theta = \boldsymbol{w}$. The Laplace approximation of posterior will be

$$p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}_{MAP}, \mathbf{H}^{-1})$$

- The required quantities are defined as

$$\begin{aligned} \boldsymbol{w}_{MAP} &= \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}) = \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X}) = \arg\min_{\boldsymbol{w}}[-\log p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X})] \\ \mathbf{H} &= \nabla^2[-\log p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X})]\big|_{\boldsymbol{w}=\boldsymbol{w}_{MAP}} \end{aligned}$$

- We can compute $\boldsymbol{w}_{MAP}$ using iterative methods (gradient descent):
  - First-order (gradient) methods: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$. Requires gradient $\boldsymbol{g}$ of $-\log p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X})$

$$\boldsymbol{g} = \nabla[-\log p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X})]$$

  - Second-order methods. $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \mathbf{H}_t^{-1} \boldsymbol{g}_t$. Requires both gradient and Hessian (defined above)
- Note: When using second order methods for estimating $\boldsymbol{w}_{MAP}$, we anyway get the Hessian needed for the Laplace approximation of the posterior

An Aside: Gradient and Hessian for Logistic Regression

- The LR objective function $-\log p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X}) = -\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) - \log p(\boldsymbol{w})$ can be written as

$$-\log \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) - \log p(\boldsymbol{w}) = -\sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) - \log p(\boldsymbol{w})$$

- For the logistic regression model, $p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \mu_n^{y_n}(1-\mu_n)^{1-y_n}$ where $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$
- With a Gaussian prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}\mathbf{I}) \propto \exp(-\lambda \boldsymbol{w}^\top \boldsymbol{w})$, the gradient and Hessian will be

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

- $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_N]^\top$ is $N \times 1$ and $\mathbf{S}$ is a $N \times N$ diagonal matrix with $S_{nn} = \mu_n(1-\mu_n)$

Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$\begin{aligned} p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*) \\ p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*)) \end{aligned}$$

- When using MAP, the predictive distribution will be

$$\begin{aligned} p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*) \\ p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) &= \text{Bernoulli}(\sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*)) \end{aligned}$$

- When using Bayesian inference, the posterior predictive distribution, based on posterior averaging

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

- Above is hard in general. :-(If using the Laplace approximation for $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$, it will be

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \approx \int \sigma(\boldsymbol{w}^\top \boldsymbol{x}_*) \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \mathbf{H}^{-1}) d\boldsymbol{w}$$

- Even after Laplace approximation for $p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$, the above integral to compute posterior predictive is intractable. So we will need to also approximate the predictive posterior. :-)

Posterior Predictive via Monte-Carlo Sampling

- The posterior predictive is given by the following integral

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \quad = \quad \int \sigma(\boldsymbol{w}^\top \boldsymbol{x}_*)\mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \mathbf{H}^{-1})d\boldsymbol{w}$$

- Monte-Carlo approximation: Draw several samples of $\boldsymbol{w}$ from $\mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \mathbf{H}^{-1})$ and replace the above integral by an empirical average of $\sigma(\boldsymbol{w}^\top \boldsymbol{x}_*)$ computed using each of those samples

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \quad \approx \quad \frac{1}{S}\sum_{s=1}^{S} \sigma(\boldsymbol{w}_s^\top \boldsymbol{x}_*)$$

  where $\boldsymbol{w}_s \sim \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \mathbf{H}^{-1})$, $s = 1, \ldots, S$

- More on Monte-Carlo methods when we discuss MCMC sampling

Predictive Posterior via Probit Approximation

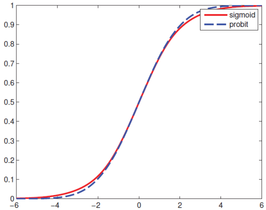
- The posterior predictive we wanted to compute was

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \quad = \quad \int \sigma(\boldsymbol{w}^\top \boldsymbol{x}_*)\mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \mathbf{H}^{-1})d\boldsymbol{w}$$

- In the above, let's replace the sigmoid $\sigma(\boldsymbol{w}^\top \boldsymbol{x}_*)$ by $\Phi(\boldsymbol{w}^\top \boldsymbol{x}_*)$, i.e., CDF of standard normal

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2} dt \qquad \text{(Note: } z \text{ is a scalar and } 0 \leq \Phi(z) \leq 1\text{)}$$

- Note: $\Phi(z)$ is also called the probit function



- This approach relies on numerical approximation (as we will see)

Predictive Posterior via Probit Approximation

- With this approximation, the predictive posterior will be

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int \Phi(\boldsymbol{w}^\top \boldsymbol{x}_*)\mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \mathbf{H}^{-1})d\boldsymbol{w} \qquad \text{(an expectation)}$$

$$= \int_{-\infty}^{\infty} \Phi(a)p(a|\mu_a, \sigma_a^2)da \qquad \text{(an equivalent expectation)}$$

- Since $a = \boldsymbol{w}^\top \boldsymbol{x}_* = \boldsymbol{x}_*^\top \boldsymbol{w}$, and $\boldsymbol{w}$ is normally distributed, $p(a|\mu_a, \sigma_a^2) = \mathcal{N}(a|\mu_a, \sigma_a^2)$, with $\mu_a = \boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*$ and $\sigma_a^2 = \boldsymbol{x}_*^\top \mathbf{H}^{-1} \boldsymbol{x}_*$ (follows from the linear trans. property of random vars)
- Given $\mu_a = \boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*$ and $\sigma_a^2 = \boldsymbol{x}_*^\top \mathbf{H}^{-1} \boldsymbol{x}_*$, the predictive posterior will be

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int_{-\infty}^{\infty} \Phi(a)\mathcal{N}(a|\mu_a, \sigma_a^2)da = \Phi\left(\frac{\mu_a}{\sqrt{1 + \sigma_a^2}}\right)$$

- Note that the variance σ_a^2 also "moderates" the probability of y_n being 1 (MAP would give $\Phi(\mu_a)$)
- Since logistic and probit aren't exactly identical, we usually scale a by a scalar t s.t. $t^2 = \pi/8$

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int_{-\infty}^{\infty} \Phi(ta)\mathcal{N}(a|\mu_a, \sigma_a^2)da = \Phi\left(\frac{\mu_a}{\sqrt{t^{-2} + \sigma_a^2}}\right)$$

Bayesian Logistic Regression: Posterior over Linear Classifiers!

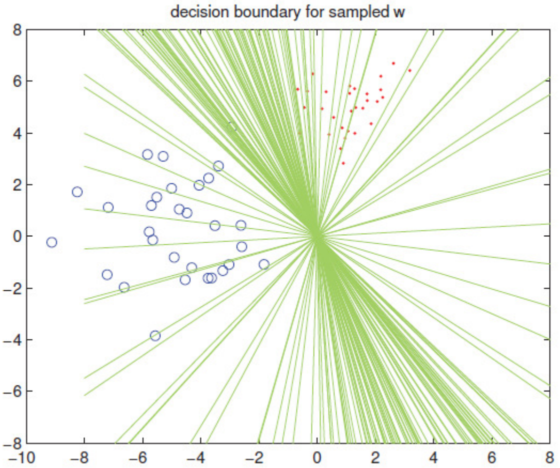
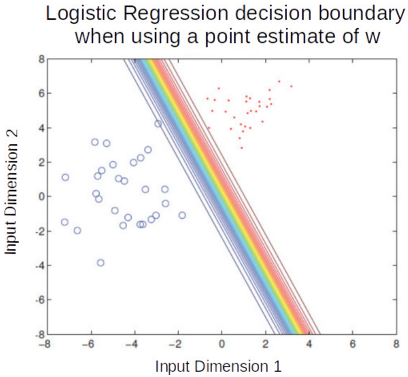
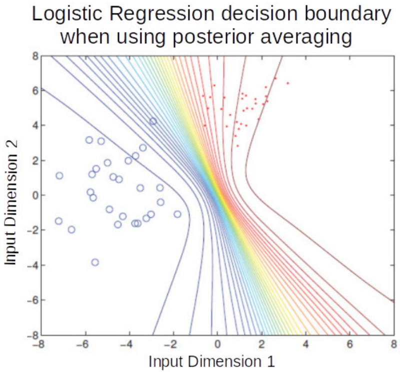


Figure courtesy: MLAPP (Murphy)

Logistic Regression: Plug-in Prediction vs Bayesian Averaging

- (Left) Predictive distribution when using a point estimate uses only a single linear hyperplane $\boldsymbol{w}$
- (Right) Posterior predictive distribution averages over many linear hyperplanes $\boldsymbol{w}$





Some Comments

- We saw basic logistic regression model and some ways to perform Bayesian inference for this model
  - We assumed the hyperparameters (e.g., precision/variance of $p(\boldsymbol{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I})$) to be fixed. However, these can also be learned if desired
  - LR is a linear classification model. Can be extended to nonlinear classification (more on this later)
- Logistic Regression (and its Bayesian version) is widely used in probabilistic classification
- Its multiclass extension is softmax regression (which again can be treated in a Bayesian manner)
- LR and softmax some of the simplest models for discriminative classification but non-conjugate
- The Laplace approximation is one of the simplest approximations to handle non-conjugacy
- A variety of other approximate inference algorithms exist for these models
  - We will revisit LR when discussing such approximate inference methods