Probabilistic Models for Classification: Logistic Regression

CS698X: Topics in Probabilistic Modeling and Inference

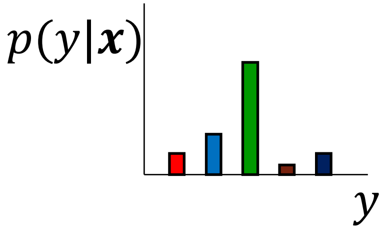
Piyush Rai

Probabilistic Models for Classification

- Goal: Learn the conditional distribution (PMF) of discrete label $\boldsymbol{y}$ given input $\boldsymbol{x}$

Will depend on some parameters (not shown here for brevity)

$$p(y|\boldsymbol{x})$$



A discrete distribution, e.g., Bernoulli or multinoulli whose parameters will depend on the inputs $\boldsymbol{x}$

- Two ways to learn this conditional distribution
- Discriminative Approach: Don't model the inputs $\boldsymbol{x}$ and directly define $p(y|\boldsymbol{x})$
- Generative Approach: Also model the inputs $\boldsymbol{x}$ and define $p(y|\boldsymbol{x})$ as

Note: Can also use it for regression if we can define the joint $p(\boldsymbol{x}, y)$ and can obtain $p(y|\boldsymbol{x})$ from that joint (usually easy if the joint is Gaussian)

Prior probability of class y

a.k.a. "class-prior"

Distribution of inputs from class y

a.k.a. "class-conditional" distribution

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})} = \frac{p(y)p(\boldsymbol{x}|y)}{p(\boldsymbol{x})} = \frac{p(y)p(\boldsymbol{x}|y)}{\sum_y p(y)p(\boldsymbol{x}|y)}$$

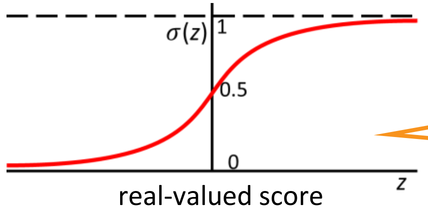
- Both discriminative and generative approaches can be learned via point estimation or by using fully Bayesian inference

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

- The posterior will be

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

Gaussian

Bernoulli

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

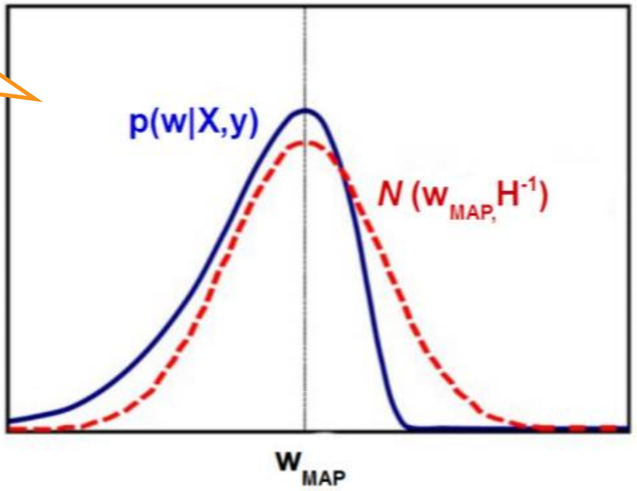
- Need to approximate the posterior in this case

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

- 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



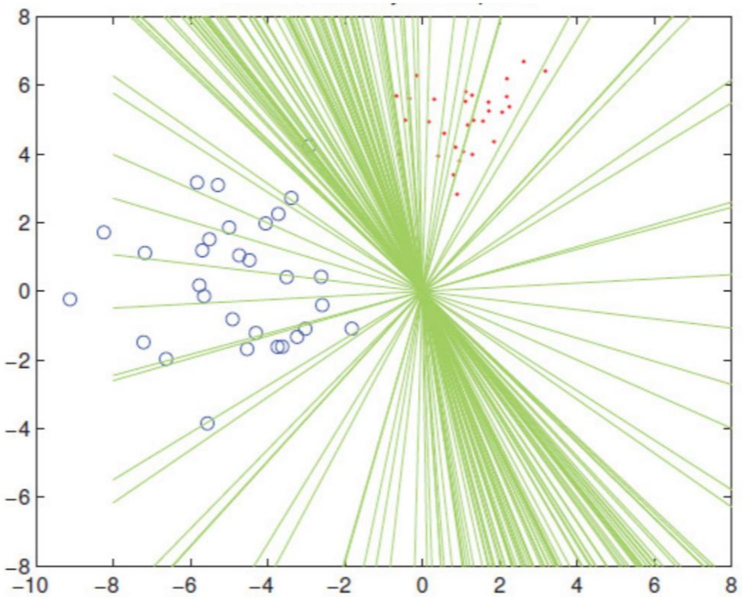
$$\begin{aligned} \boldsymbol{w}_{MAP} &= \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{w}|\boldsymbol{y},\mathbf{X}) \\ &= \arg\max_{\boldsymbol{w}} \log p(\boldsymbol{y},\boldsymbol{w}|\mathbf{X}) \\ &= \arg\min_{\boldsymbol{w}} [-\log p(\boldsymbol{y},\boldsymbol{w}|\mathbf{X})] \end{aligned}$$

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 M samples $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_M$, from the approx. of posterior
  - Approximate the PPD as follows

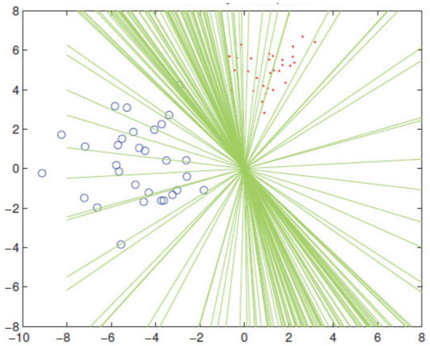
$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) \approx \frac{1}{M}\sum_{m=1}^{M} p(y_* = 1|\boldsymbol{w}_{\mathrm{m}}, \boldsymbol{x}_*) = \frac{1}{M}\sum_{m=1}^{M} \sigma(\boldsymbol{w}_m^\top \boldsymbol{x}_n)$$

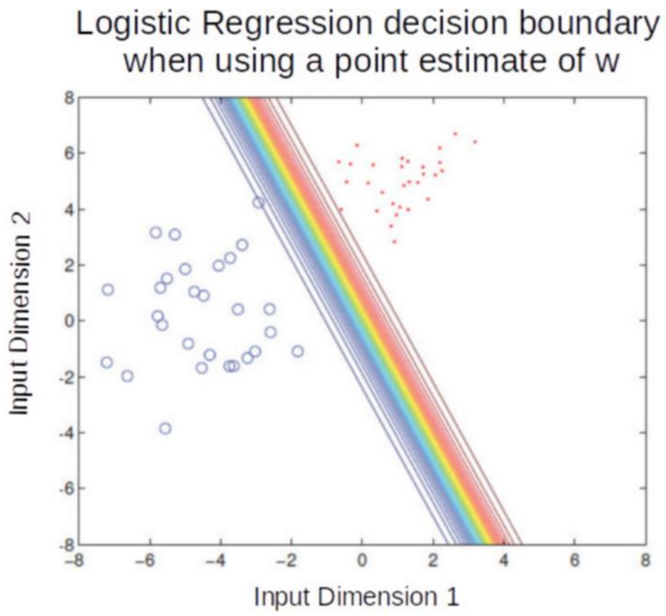
- In contrast, when using MLE/MAP solution $\hat{\boldsymbol{w}}_{opt}$, the plug-in pred. distribution

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

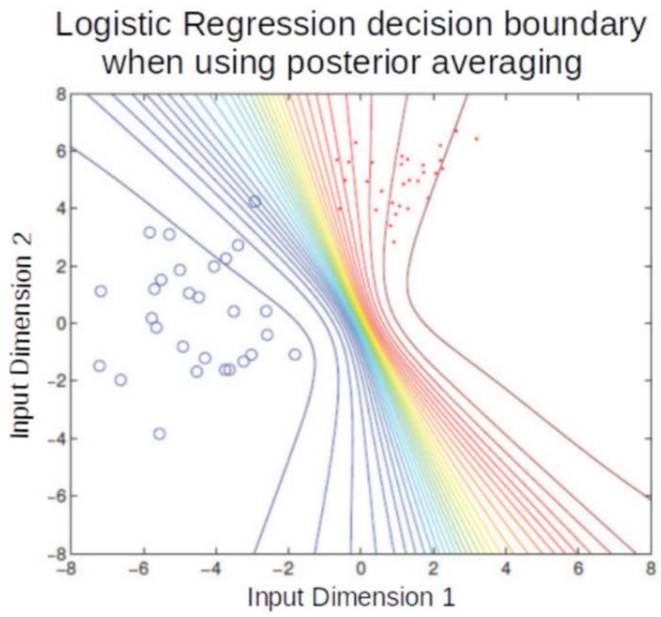
LR: Plug-in Prediction vs Bayesian Averaging

- 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




Logistic Regression decision boundary when using a point estimate of w

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


Logistic Regression decision boundary when using posterior averaging

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

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

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

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

Coming Up

- Laplace Approximation of the posterior
- Generative Classification