Probabilistic Models for Supervised Learning

Piyush Rai

Introduction to Machine Learning (CS771A)

August 16, 2018

Announcements

- Homework 1 will be out tonight. Due on August 31, 11:59pm. Please start early.
- Project ideas will be posted by tomorrow.
- Project group formation deadline extended to August 25.
    - Piazza has a "Search for Teammates" features (the very first pinned post)
- Please sign-up on Piazza (we won't sign you up if you are waiting for that :-))
- TA office hours and office locations posted on Piazza (under resources/staff section)

Recap: Probabilistic Modeling

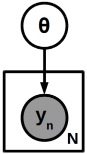


Plate Notation

- A probabilistic model is specified by two key components
  - An observation model $p(\boldsymbol{y}|\theta)$, a.k.a. the likelihood model
  - (Optionally) A prior distribution $p(\theta)$ over the unknown parameters
- Note that these two components specify the joint distribution $p(\boldsymbol{y}, \theta)$ of data and unknowns
- We can incorporate our assumptions about the data via the observation/likelihood model
- We can incorporate our assumptions about the parameters via the prior distribution
- Note: Likelihood and/or prior may depend on additional "hyperparamers" (fixed/unknown)

Recap: Parameter Estimation

- Can do point estimation (via MLE/MAP) for θ or infer its full posterior (via Bayesian inference)
- MLE maximizes the (log of) likelihood w.r.t. the parameters θ. For i.i.d. data,

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \sum_{n=1}^{N} \log p(y_n \mid \theta) = \arg\min_{\theta} NLL(\theta)$$

- MLE is akin to empirical/training loss minimization (no regularization)
- MAP estimation maximizes the (log of) posterior w.r.t. the parameters θ. For i.i.d. data,

$$\hat{\theta}_{MAP} = \arg\max_{\theta} [\sum_{n=1}^{N} \log p(y_n \mid \theta) + \log p(\theta)] = \arg\min_{\theta} [NLL(\theta) - \log p(\theta)]$$

- MAP is akin to regularized loss minimization (prior acts as a regularizer)
- Bayesian inference computes the full posterior distribution of θ

$$p(\theta|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\theta)p(\theta)}{p(\boldsymbol{y})} = \frac{p(\boldsymbol{y}|\theta)p(\theta)}{\int p(\boldsymbol{y}|\theta)p(\theta)d\theta} \quad \text{(intractable in general)}$$

Recap: Predictive Distribution

- Using estimated θ, we usually want the predictive distribution $p(y_*|\mathbf{y})$ for some future data y_*
- The proper, exact way of getting the predictive distribution (assuming i.i.d. data) is

$$p(y_*|\mathbf{y}) = \underbrace{\int p(y_*, \theta|\mathbf{y})d\theta}_{\text{sum rule of probability}} = \underbrace{\int p(y_*|\theta, \mathbf{y})p(\theta|\mathbf{y})d\theta}_{\text{chain/product rule of probability}}$$

$$= \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta \qquad \text{(assuming i.i.d. data)}$$

- If using a point estimate $\hat{\theta}$ (e.g., MLE/MAP), $p(\theta|\mathbf{y}) \approx \delta_{\hat{\theta}}(\theta)$, where $\delta()$ denotes Dirac function

$$p(y_*|\mathbf{y}) = \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta \approx p(y_*|\hat{\theta}_{MLE}) \qquad \text{(MLE based prediction)}$$

$$p(y_*|\mathbf{y}) = \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta \approx p(y_*|\hat{\theta}_{MAP}) \qquad \text{(MAP based prediction)}$$

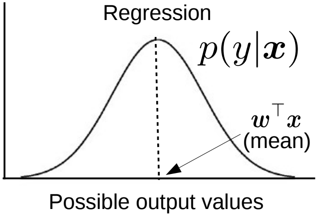
- If using the fully Bayesian inference, $p(y_*|\mathbf{y}) = \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta \Leftarrow$ uses the proper way!
  - The integral here may not always be tractable and may need to be approximated

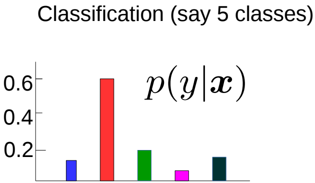
Probabilistic Models for Supervised Learning

Want models that give us $p(y|x)$

Why Probabilistic Models for Supervised Learning?

- Often, we want the distribution $p(y|\boldsymbol{x})$ over possible outputs y, given an input $\boldsymbol{x}$





- The distribution $p(y|\boldsymbol{x})$ is more informative, since it can tell us
  - What is the "expected" or "most likely" value of the predicted output y?
  - What is the "uncertainty" in the predicted output y?
  - .. and gives "soft" predictions (e.g., rather than yes/no prediction, gives prob. of "yes")
- Moreover, we can use priors over model parameters, perform fully Bayesian inference, etc.

Probabilistic Models for Supervised Learning

- Usually <u>two ways</u> to model the conditional distribution $p(y|\boldsymbol{x})$ of outputs given inputs
- **Approach 1:** Don't model $\boldsymbol{x}$, and model $p(y|\boldsymbol{x})$ <u>directly</u> using a prob. distribution, e.g.,

$$\begin{aligned} p(y|\boldsymbol{w},\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1}) && \text{(prob. linear regression)} \\ p(y|\boldsymbol{w},\boldsymbol{x}) &= \text{Bernoulli}[\sigma(\boldsymbol{w}^\top \boldsymbol{x})] && \text{(prob. linear binary classification)} \end{aligned}$$

(note: $\boldsymbol{w}^\top \boldsymbol{x}$ above only for linear prob. model; can even replace it by a possibly nonlinear $f(\boldsymbol{x})$)

- **Approach 2:** Model both $\boldsymbol{x}$ and y via the joint distr. $p(\boldsymbol{x}, y)$, and then get the conditional as

$$\begin{aligned} p(y|\boldsymbol{x},\theta) &= \frac{p(\boldsymbol{x},y|\theta)}{p(\boldsymbol{x}|\theta)} && \text{(note: } \theta \text{ collectively denotes all the parameters)} \\ p(y=k|\boldsymbol{x},\theta) &= \frac{p(\boldsymbol{x},y=k|\theta)}{p(\boldsymbol{x}|\theta)} = \frac{p(\boldsymbol{x}|y=k,\theta)p(y=k|\theta)}{\sum_{\ell=1}^{K} p(\boldsymbol{x}|y=\ell,\theta)p(y=\ell|\theta)} && \text{(for } K \text{ class classification)} \end{aligned}$$

- Approach 1 called Discriminative Modeling; Approach 2 called fully Generative Modeling
  - Discriminative models only model y, not $\boldsymbol{x}$, Generative Models model both y and $\boldsymbol{x}$

Today: Discriminative Models for Probabilistic Regression/Classification

1: Probabilistic Linear Regression

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

2: Logistic Regression for Binary Classification

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

(Remember that these do NOT model $\boldsymbol{x}$, but only model y)

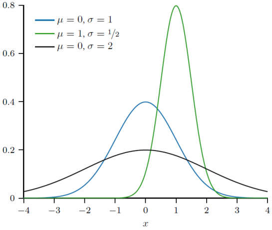
(Also, both are linear models (note the $\boldsymbol{w}^\top \boldsymbol{x}$))

Gaussian Distribution: Brief Review

Univariate Gaussian Distribution

- Distribution over real-valued scalar r.v. x
- Defined by a scalar **mean** μ and a scalar **variance** σ^2
- Distribution defined as

$$\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

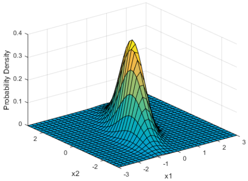


- Mean: $\mathbb{E}[x] = \mu$
- Variance: $\text{var}[x] = \sigma^2$
- Precision (inverse variance) $\beta = 1/\sigma^2$

Multivariate Gaussian Distribution

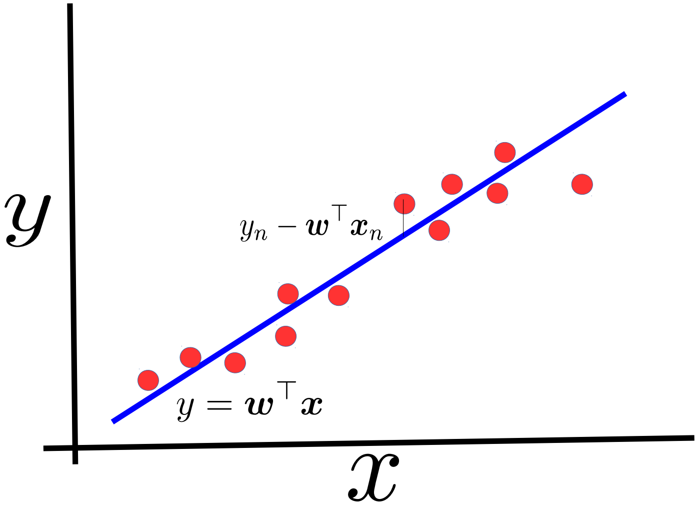
- Distribution over a multivariate r.v. vector $\mathbf{x} \in \mathbb{R}^D$ of real numbers
- Defined by a **mean vector** $\boldsymbol{\mu} \in \mathbb{R}^D$ and a $D \times D$ **covariance matrix** $\boldsymbol{\Sigma}$

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

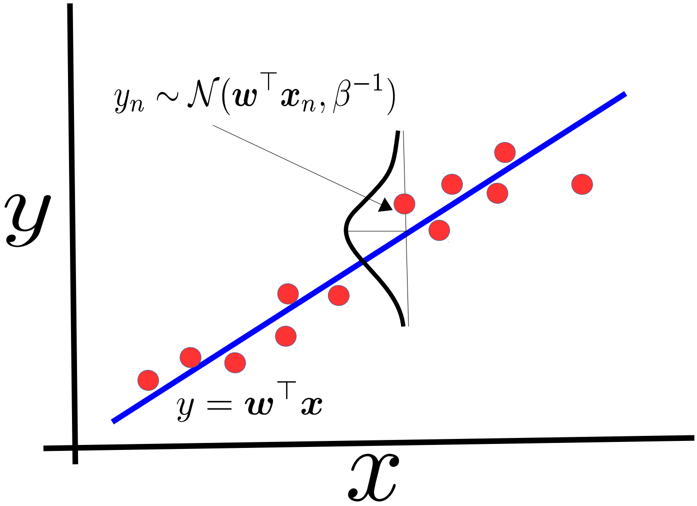


- The covariance matrix $\boldsymbol{\Sigma}$ must be symmetric and positive definite
  - All eigenvalues are positive
  - $\boldsymbol{z}^\top \boldsymbol{\Sigma} \boldsymbol{z} > 0$ for any real vector $\boldsymbol{z}$

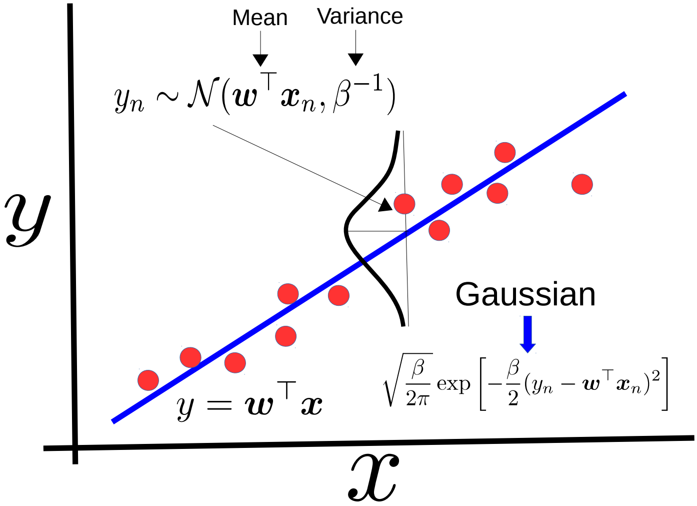
Linear Regression: A Probabilistic View



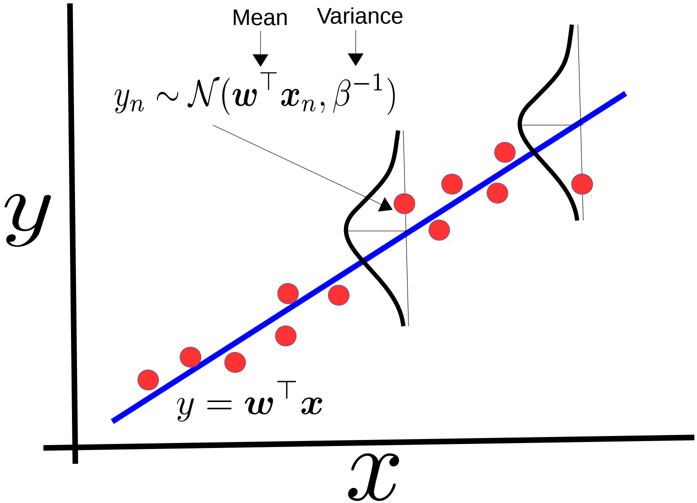
Linear Regression: A Probabilistic View



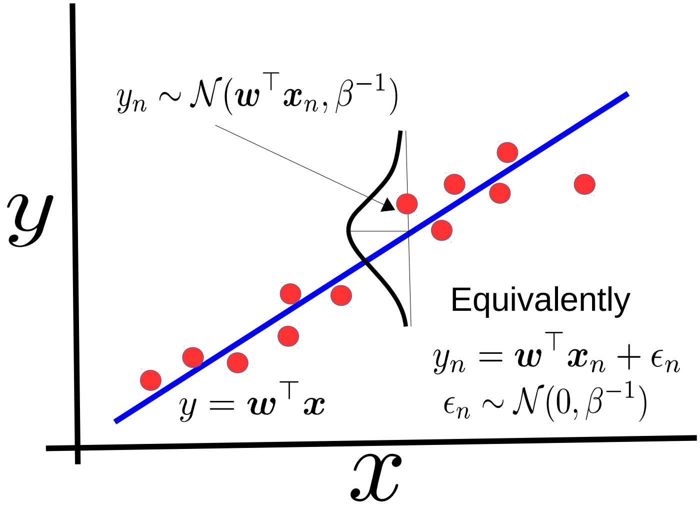
Linear Regression: A Probabilistic View



Linear Regression: A Probabilistic View

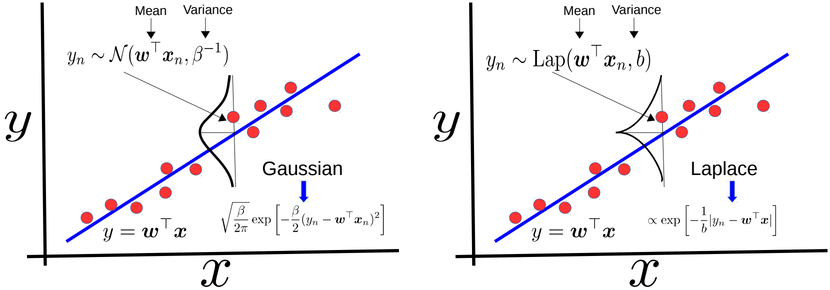


Linear Regression: A Probabilistic View



Probabilistic Linear Regression: Some Comments

- Modeling $p(y|\boldsymbol{w}, \boldsymbol{x})$ as a Gaussian $p(y|\boldsymbol{w}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$ is just one possibility
- Can model $p(y|\boldsymbol{w}, \boldsymbol{x})$ using other distributions too, e.g., Laplace (better handles outliers)



- Even with Gaussian, can assume each output to have a different variance (heteroscedastic noise)

$$p(y|\boldsymbol{w}, \boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n, \beta_n^{-1})$$

MLE for Probabilistic Linear Regression

- Since each likelihood term is a Gaussian, we have

$$p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1}) = \sqrt{\frac{\beta}{2\pi}} \exp\left[-\frac{\beta}{2}(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2\right]$$

- Thus the likelihood (assuming i.i.d. responses) will be

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \left(\frac{\beta}{2\pi}\right)^{N/2} \exp\left[-\frac{\beta}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2\right]$$

- Note: $\boldsymbol{x}_n$ (features) assumed given/fixed. Only modeling the response y_n
- **Log-likelihood** (ignoring constants w.r.t. $\boldsymbol{w}$)

$$\boxed{\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) \propto -\frac{\beta}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2}$$

- Note that **negative** log likelihood (NLL) in this case is similar to squared loss function
- Therefor MLE with this model will give the same solution as (unregularized) least squares

MAP Estimation for Probabilistic Linear Regression

- Let's assume a zero-mean **multivariate Gaussian prior** on weight vector $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I}_D) \propto \exp\left[-\frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}\right] = \exp\left[-\frac{\lambda}{2}\sum_{d=1}^{D} w_d^2\right]$$

- This prior encourages each weight w_d to be small (close to zero), similar to ℓ_2 regularization
- The MAP objective (log-posterior) will be the log-likelihood + $\log p(\boldsymbol{w})$

$$-\frac{\beta}{2}\sum_{n=1}^{N}(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2 - \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

- Maximizing this is equivalent to minimizing the following w.r.t. $\boldsymbol{w}$

$$\boxed{\hat{\mathbf{w}}_{MAP} = \arg\min_{\mathbf{w}} \sum_{n=1}^{N}(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n)^2 + \frac{\lambda}{\beta}\boldsymbol{w}^\top \boldsymbol{w}}$$

- Note that $\frac{\lambda}{\beta}$ is like a regularization hyperparam (as in ridge regression)

Fully Bayesian Inference for Probabilistic Linear Regression

- Can also compute the full posterior distribution over $\boldsymbol{w}$

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

- Since the likelihood (Gaussian) and prior (Gaussian) are conjugate, posterior is easy to compute
- After some algebra, it can be shown that (will provide a note)

$$\begin{aligned} p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}) &= \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N) \\ \boldsymbol{\Sigma}_N &= (\beta \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} \\ \boldsymbol{\mu}_N &= (\mathbf{X}^\top \mathbf{X} + \frac{\lambda}{\beta} \mathbf{I}_D)^{-1} \mathbf{X}^\top \boldsymbol{y} \end{aligned}$$

- Note: We are assuming the hyperparameters β and λ to be known
- Note: For brevity, we have omitted the hyperparams from the conditioning in various distributions such as $p(\boldsymbol{w})$, $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})$, $p(\boldsymbol{y}|\mathbf{X})$, $p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X})$

Predictive Distribution

- Now we want the predictive distribution $p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y})$ of the output y_* for a new input $\boldsymbol{x}_*$
- With MLE/MAP estimate of $\boldsymbol{w}$, the prediction can be made by simply plugging in the estimate

$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \approx p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) = \mathcal{N}(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*, \beta^{-1}) \quad \text{- MLE prediction}$$
$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \approx p(y_*|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) = \mathcal{N}(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*, \beta^{-1}) \quad \text{- MAP prediction}$$

- When doing fully Bayesian inference, we can compute the posterior predictive distribution

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

- Due to Gaussian conjugacy, this too will be a Gaussian (note the form, ignore the proof :-))

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

- In this case, we also get an input-specific predictive variance (unlike MLE/MAP prediction)
  - Very useful in applications where we want confidence estimates of the predictions made by the model

MLE, MAP/Fully Bayesian Linear Regression: Summary

- MLE/MAP give point estimate of $\boldsymbol{w}$
- Fully Bayesian approach gives the full posterior
- MLE/MAP based prediction uses a single best estimate of $\boldsymbol{w}$
- Fully Bayesian prediction does posterior averaging
- **Some things to keep in mind:**
    - MLE estimation of a parameter leads to unregularized solutions
    - MAP estimation of a parameter leads to regularized solutions
    - A Gaussian likelihood model corresponds to using squared loss
    - A Gaussian prior on parameters acts as an ℓ_2 regularizer
    - Other likelihoods/priors can be chosen (result in other loss functions and regularizers)

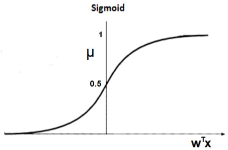
Discriminative Models for Probabilistic Classification

(Again, only y will be modeled, $\boldsymbol{x}$ treated as "fixed")

Logistic Regression

- Perhaps the simplest discriminative probabilistic model for linear binary classification
- Defines $\mu = p(y=1|\boldsymbol{x})$ using the sigmoid function

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



- Here $\boldsymbol{w}^\top \boldsymbol{x}$ is the score for input $\boldsymbol{x}$. The sigmoid turns it into a probability. Thus we have

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

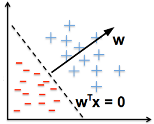
- **Note:** If we assume $y \in \{-1,+1\}$ instead of $y \in \{0,1\}$ then $p(y|\boldsymbol{x},\boldsymbol{w}) = \frac{1}{1+\exp(-y\boldsymbol{w}^\top \boldsymbol{x})}$

Logistic Regression: A Closer Look..

- At the decision boundary where both classes are equiprobable:

$$
\begin{aligned}
p(y=1|\boldsymbol{x},\boldsymbol{w}) &= p(y=0|\boldsymbol{x},\boldsymbol{w}) \\
\frac{\exp(\boldsymbol{w}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} &= \frac{1}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x})} \\
\exp(\boldsymbol{w}^\top \boldsymbol{x}) &= 1 \\
\boldsymbol{w}^\top \boldsymbol{x} &= 0
\end{aligned}
$$

- Thus the decision boundary of LR is a linear hyperplane
- Therefore $y = 1$ if $\boldsymbol{w}^\top \boldsymbol{x} \geq 0$, otherwise $y = 0$



- High positive (negative) score $\boldsymbol{w}^\top \boldsymbol{x}$: High (low) probability of label 1

MLE for Logistic Regression

- Each label $y_n = 1$ with probability $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$
- Assuming i.i.d. labels, likelihood is product of Bernoullis

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

- Negative log-likelihood: $\text{NLL}(\boldsymbol{w}) = -\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \log \mu_n + (1-y_n)\log(1-\mu_n))$
- Note: The NLL in this case is the same as cross-entropy loss function (a classification loss fn)
- Plugging in $\mu_n = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}{1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)}$ and chugging, we get (verify yourself)

$$\boxed{\text{NLL}(\boldsymbol{w}) = -\sum_{n=1}^{N}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \log(1+\exp(\boldsymbol{w}^\top \boldsymbol{x}_n)))}$$

- MLE solution: $\hat{\boldsymbol{w}}_{MLE} = \arg\min_{\boldsymbol{w}} \text{NLL}(\boldsymbol{w})$. No closed form solution (you can verify)
- Requires iterative methods (e.g., gradient descent). We will look at these later.
  - Exercise: Try computing the gradient of $\text{NLL}(\boldsymbol{w})$ and note the form of the gradient

MAP Estimation for Logisic Regression

- To do MAP estimation for $\boldsymbol{w}$, can use a prior $p(\boldsymbol{w})$ on $\boldsymbol{w}$
- Just like the probabilistic linear regression case, let's put a Gausian prior on $\boldsymbol{w}$

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

- MAP objective (log of posterior) = MLE objective + $\log p(\boldsymbol{w})$
- The MAP estimate of $\boldsymbol{w}$ will be

$$\hat{\boldsymbol{w}}_{MAP} = \arg\min_{\boldsymbol{w}} \left[\text{NLL}(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w} \right]$$

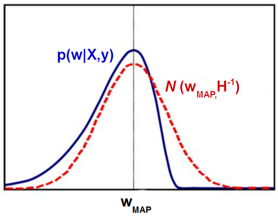
Fully Bayesian Estimation for Logistic Regression

- Doing fully Bayesian inference would require computing the posterior

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

- Intractable. Reason: likelihood (logistic-Bernoulli) and prior (Gaussian) here are **not conjugate**
- Need to do *approximate inference* in this case
- A crude approximation: **Laplace approximation**: Approximate a posterior by a **Gaussian** with mean $=\boldsymbol{w}_{MAP}$ and covariance $=$ inverse hessian (hessian $=$ second derivative of $\log p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$)

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



Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \approx p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MLE}) = \sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*)$$
$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \approx \text{Bernoulli}(\sigma(\boldsymbol{w}_{MLE}^\top \boldsymbol{x}_*))$$

- When using MAP, the predictive distribution will be

$$p(y_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \approx p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{MAP}) = \sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*)$$
$$p(y_*|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) \approx \text{Bernoulli}(\sigma(\boldsymbol{w}_{MAP}^\top \boldsymbol{x}_*))$$

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

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

- Note: Unlike the linear regression case, for logistic regression (and for non-conjugate models in general), posterior averaging can be intractable (and may require approximations)

Multiclass Logistic (a.k.a. Softmax) Regression

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

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

- $\mathbf{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_K]$ is $D \times K$ weight matrix. $\boldsymbol{w}_1 = \mathbf{0}_{D\times 1}$ (assumed for identifiability)
- Popularly known as the softmax function
- Each likelihood $p(y_n|\boldsymbol{x}_n, \mathbf{W})$ is a multinoulli distribution. Therefore

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

  where $y_{n\ell} = 1$ if true class of example n is ℓ and $y_{n\ell'} = 0$ for all other $\ell' \neq \ell$
- Can do MLE/MAP/fully Bayesian estimation for $\mathbf{W}$ similar to the logistic regression model
- Will look at optimization methods for this and other loss functions later.

Summary

- Looked at probabilistic models for supervised learning (regression and classification)
- Can do MLE/MAP, of fully Bayesian inference in these models
- MLE/MAP is like loss function (or regularized loss function) minimization
- Fully Bayesian inference is usually harder/expensive but often considered better
  - We get the full posterior over the parameters
  - We can do posterior averagring when computing the predictive distribution
  - Can get variance/confidence estimate in our predictions
- Can model $p(y|\boldsymbol{x})$ direcly (discriminative models) or via $p(\boldsymbol{x}, y)$ (generative models)
- Looked at discriminative models for regression and classification
- Will look at generative models for learning $p(y|\boldsymbol{x})$ next week