

# Probabilistic Models for Supervised Learning

Piyush Rai

Introduction to Machine Learning (CS771A)

August 16, 2018

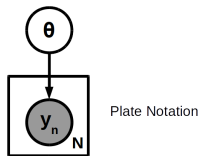


# Announcements

- 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



- A probabilistic model is specified by two key components
  - An observation model  $p(\mathbf{y}|\theta)$ , 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**  $p(\mathbf{y}, \theta)$  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 **"hyperparameters"** (fixed/unknown)



# Recap: Parameter Estimation

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

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

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

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

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

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



# Recap: Predictive Distribution

- Using estimated  $\theta$ , 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

$$\begin{aligned} p(y_*|\mathbf{y}) &= \underbrace{\int p(y_*, \theta|\mathbf{y})d\theta}_{\text{sum rule of probability}} = \underbrace{\int p(y_*|\theta, \mathbf{y})p(\theta|\mathbf{y})d\theta}_{\text{chain/product rule of probability}} \\ &= \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta \quad (\text{assuming i.i.d. data}) \end{aligned}$$

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

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

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

- 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



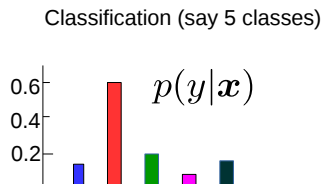
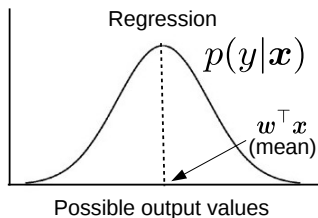
# Probabilistic Models for Supervised Learning

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



# Why Probabilistic Models for Supervised Learning?

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



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

$$p(y|\mathbf{w}, \mathbf{x}) = \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \beta^{-1}) \quad (\text{prob. linear regression})$$

$$p(y|\mathbf{w}, \mathbf{x}) = \text{Bernoulli}[\sigma(\mathbf{w}^\top \mathbf{x})] \quad (\text{prob. linear binary classification})$$

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

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

$$p(y|\mathbf{x}, \theta) = \frac{p(\mathbf{x}, y|\theta)}{p(\mathbf{x}|\theta)} \quad (\text{note: } \theta \text{ collectively denotes all the parameters})$$

$$p(y = k|\mathbf{x}, \theta) = \frac{p(\mathbf{x}, y = k|\theta)}{p(\mathbf{x}|\theta)} = \frac{p(\mathbf{x}|y = k, \theta)p(y = k|\theta)}{\sum_{\ell=1}^K p(\mathbf{x}|y = \ell, \theta)p(y = \ell|\theta)} \quad (\text{for } K \text{ class classification})$$

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





# Today: Discriminative Models for Probabilistic Regression/Classification

## 1: Probabilistic Linear Regression

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

## 2: Logistic Regression for Binary Classification

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

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

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



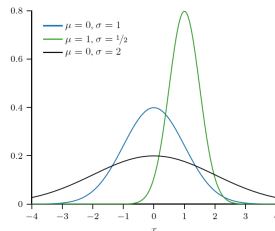
# Gaussian Distribution: Brief Review



# Univariate Gaussian Distribution

- Distribution over real-valued scalar r.v.  $x$
- Defined by a scalar **mean**  $\mu$  and a scalar **variance**  $\sigma^2$
- Distribution defined as

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



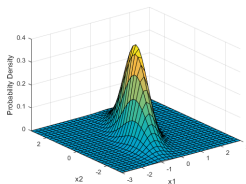
- Mean:  $\mathbb{E}[x] = \mu$
- Variance:  $\text{var}[x] = \sigma^2$
- Precision (inverse variance)  $\beta = 1/\sigma^2$



# 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**  $\boldsymbol{\Sigma}$

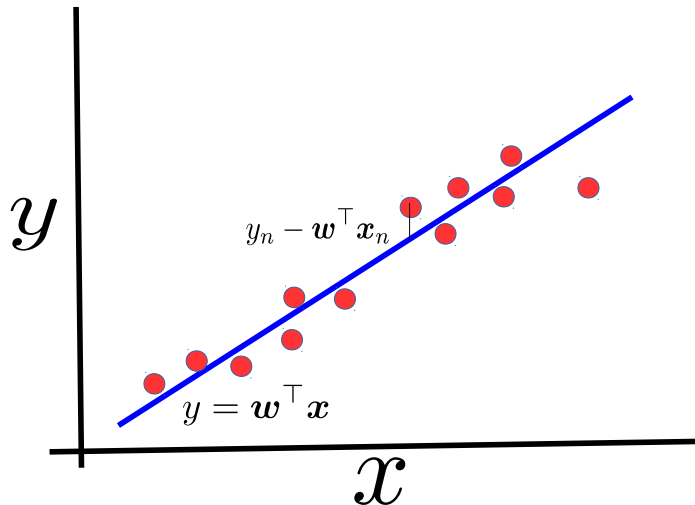
$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$



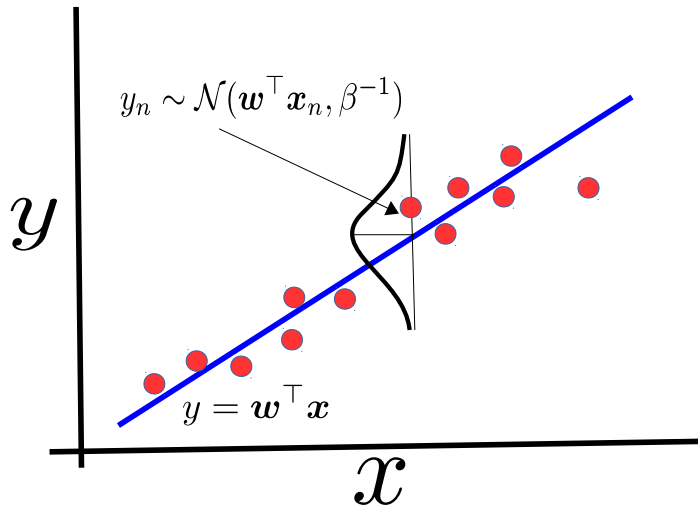
- The covariance matrix  $\boldsymbol{\Sigma}$  must be symmetric and positive definite
  - All eigenvalues are positive
  - $\mathbf{z}^\top \boldsymbol{\Sigma} \mathbf{z} > 0$  for any real vector  $\mathbf{z}$



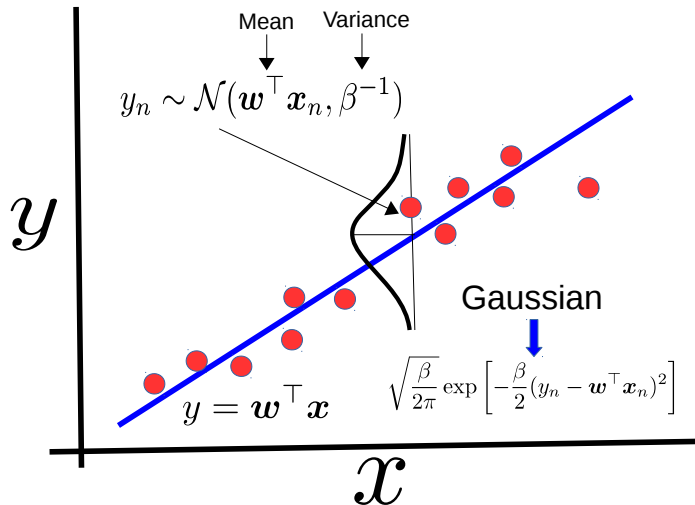
# Linear Regression: A Probabilistic View



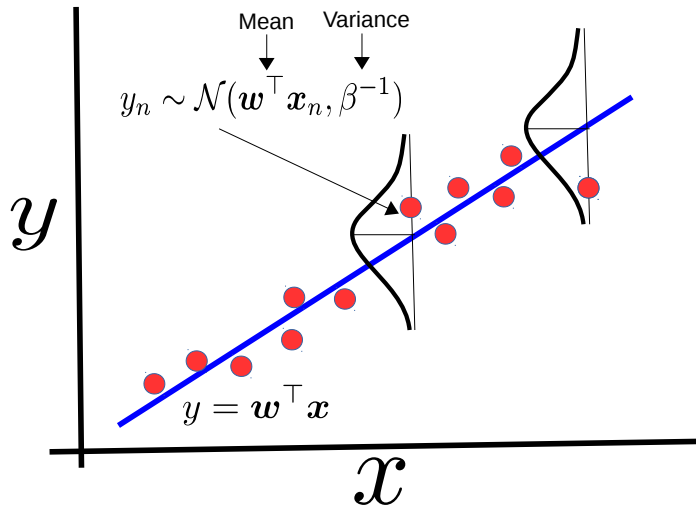
# Linear Regression: A Probabilistic View



# Linear Regression: A Probabilistic View

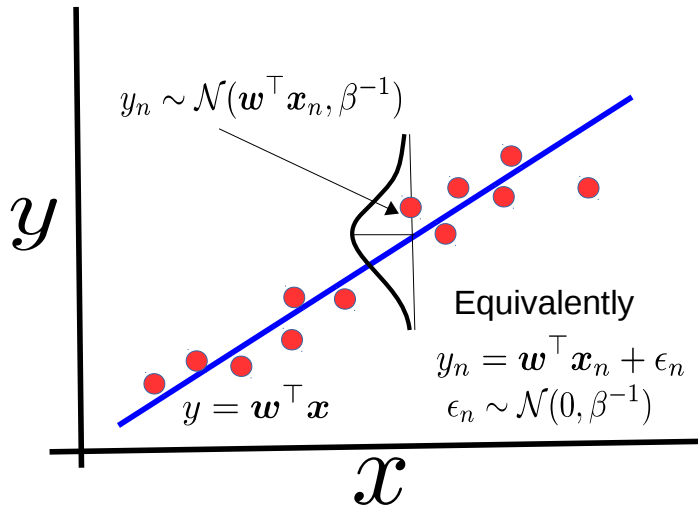


# Linear Regression: A Probabilistic View



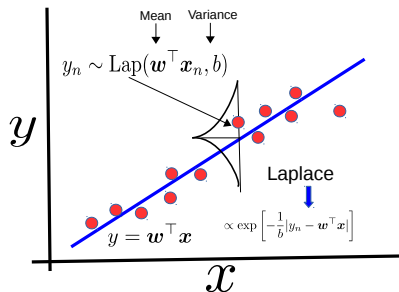
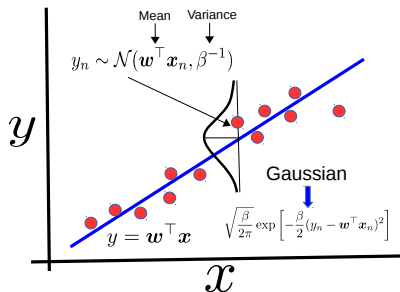


# Linear Regression: A Probabilistic View



# Probabilistic Linear Regression: Some Comments

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



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

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



# MLE for Probabilistic Linear Regression

- Since each likelihood term is a Gaussian, we have

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

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

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

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

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

- 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  $\mathbf{w}$

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

- This prior encourages each weight  $w_d$  to be small (close to zero), similar to  $\ell_2$  regularization
- The MAP objective (log-posterior) will be the **log-likelihood** + **log  $p(\mathbf{w})$**

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

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

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

- 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  $\mathbf{w}$

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

- 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)

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

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



# Predictive Distribution

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

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

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

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

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

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

$$p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_N^\top \mathbf{x}_*, \beta^{-1} + \mathbf{x}_*^\top \Sigma_N \mathbf{x}_*)$$

- 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  $\mathbf{w}$
- Fully Bayesian approach gives the full posterior
- MLE/MAP based prediction uses a single best estimate of  $\mathbf{w}$
- 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  $\ell_2$  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,  $x$  treated as “fixed”)

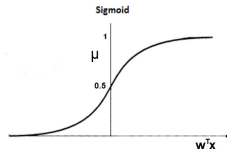




# Logistic Regression

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

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



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

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \mu = \sigma(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})} = \frac{\exp(\mathbf{w}^\top \mathbf{x})}{1 + \exp(\mathbf{w}^\top \mathbf{x})}$$

$$p(y = 0|\mathbf{x}, \mathbf{w}) = 1 - \mu = 1 - \sigma(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}^\top \mathbf{x})}$$

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

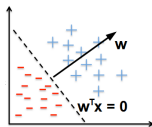


# Logistic Regression: A Closer Look..

- At the decision boundary where both classes are equiprobable:

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

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



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



# MLE for Logistic Regression

- Each label  $y_n = 1$  with probability  $\mu_n = \frac{\exp(\mathbf{w}^\top \mathbf{x}_n)}{1 + \exp(\mathbf{w}^\top \mathbf{x}_n)}$

- Assuming i.i.d. labels, likelihood is product of Bernoullis

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

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

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

- MLE solution:  $\hat{\mathbf{w}}_{MLE} = \arg \min_{\mathbf{w}} \text{NLL}(\mathbf{w})$ . 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  $\text{NLL}(\mathbf{w})$  and note the form of the gradient



# MAP Estimation for Logistic Regression

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

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

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

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



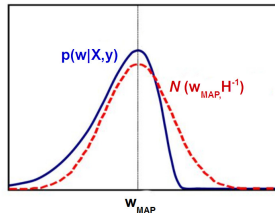
# Fully Bayesian Estimation for Logistic Regression

- Doing fully Bayesian inference would require computing the posterior

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

- **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 =  $\mathbf{w}_{MAP}$**  and **covariance = inverse hessian** (hessian = second derivative of  $\log p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ )

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



# Logistic Regression: Predictive Distributions

- When using MLE, the predictive distribution will be

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

- When using MAP, the predictive distribution will be

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

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

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

- **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,  $y_n \in \{1, 2, \dots, K\}$  and label probabilities are defined as

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

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

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

where  $y_{n\ell} = 1$  if true class of example  $n$  is  $\ell$  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 averaging** when computing the predictive distribution
  - Can get variance/confidence estimate in our predictions
- Can model  $p(y|\mathbf{x})$  directly (discriminative models) or via  $p(\mathbf{x}, y)$  (generative models)
- Looked at discriminative models for regression and classification
- Will look at generative models for learning  $p(y|\mathbf{x})$  next week

