



# Logistic Regression

Rishabh Iyer

University of Texas at Dallas

based on the slides of Nick Rouzzi and Vibhav Gogate

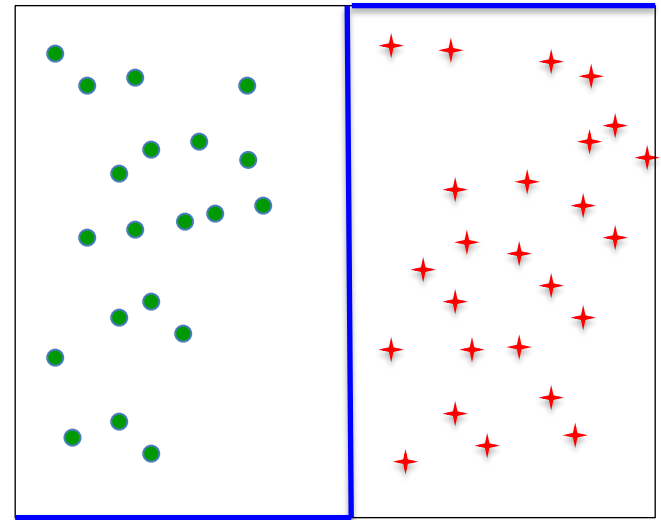
- Supervised learning via naive Bayes
  - Use MLE to estimate a distribution  $p(x, y) = p(y)p(x|y)$
  - Classify by looking at the conditional distribution,  $p(y|x)$
- Today: logistic regression

# Logistic Regression



- Learn  $p(Y|X)$  directly from the data
  - Assume a particular functional form, e.g., a linear classifier  $p(Y = 1|x) = 1$  on one side and 0 on the other
  - Not differentiable...
    - Makes it difficult to learn
    - Can't handle noisy labels

$$p(Y = 1|x) = 0$$



$$p(Y = 1|x) = 1$$

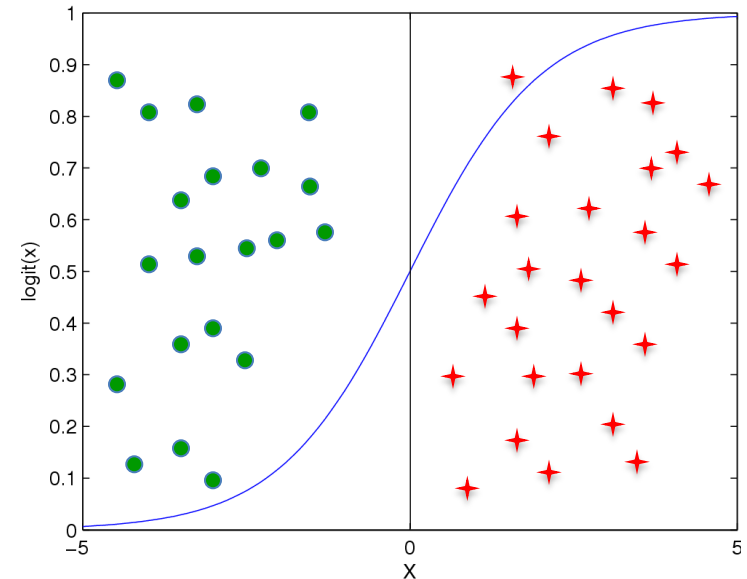
# Logistic Regression



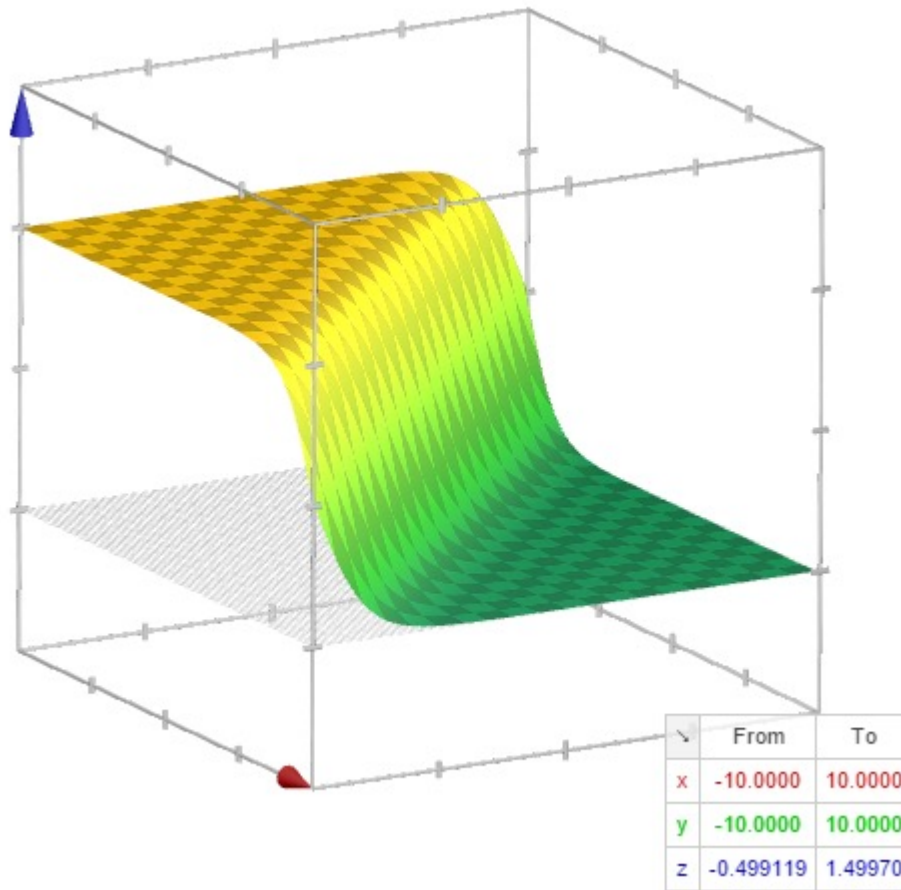
- Learn  $p(y|x)$  directly from the data
- Assume a particular functional form

$$p(Y = -1|x) = \frac{1}{1 + \exp(w^T x + b)}$$

$$p(Y = 1|x) = \frac{\exp(w^T x + b)}{1 + \exp(w^T x + b)}$$



# Logistic Function in $m$ Dimensions



$$p(Y = -1|x) = \frac{1}{1 + \exp(w^T x + b)}$$

**Can be applied to  
discrete and  
continuous features**

- Given some  $w$  and  $b$ , we can classify a new point  $x$  by assigning the label 1 if  $p(Y = 1|x) > p(Y = -1|x)$  and  $-1$  otherwise
  - This leads to a linear classification rule:
    - Classify as a 1 if  $w^T x + b > 0$
    - Classify as a  $-1$  if  $w^T x + b < 0$

- To learn the weights, we maximize the **conditional likelihood**

$$(w^*, b^*) = \arg \max_{w, b} \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b)$$

- This is not the same strategy that we used in the case of naive Bayes
  - For naive Bayes, we maximized the log-likelihood

# Generative vs. Discriminative Classifiers

## Generative classifier: (e.g., Naïve Bayes)

- Assume some **functional form** for  $p(x|y), p(y)$
- Estimate parameters of  $p(x|y), p(y)$  directly from training data
- Use Bayes rule to calculate  $p(y|x)$
- This is a **generative model**
  - **Indirect** computation of  $p(Y|X)$  through Bayes rule
  - As a result, **can also generate a sample of the data**,  
$$p(x) = \sum_y p(y)p(x|y)$$

## Discriminative classifiers: (e.g., Logistic Regression)

- Assume some **functional form for  $p(y|x)$**
- Estimate parameters of  $p(y|x)$  directly from training data
- This is a **discriminative model**
  - Directly learn  $p(y|x)$
  - But **cannot obtain a sample of the data** as  $p(x)$  is not available
  - Useful for discriminating labels



# Learning the Weights



$$\begin{aligned}\ell(w, b) &= \ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \ln p(y^{(i)} | x^{(i)}, w, b)\end{aligned}$$

# Learning the Weights



$$\begin{aligned}\ell(w, b) &= \ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \ln p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln p(Y = 1 | x^{(i)}, w, b) + \left(1 - \frac{y^{(i)} + 1}{2}\right) \ln p(Y = -1 | x^{(i)}, w, b)\end{aligned}$$

# Learning the Weights



$$\begin{aligned}\ell(w, b) &= \ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \ln p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln p(Y = 1 | x^{(i)}, w, b) + \left(1 - \frac{y^{(i)} + 1}{2}\right) \ln p(Y = -1 | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln \frac{p(Y = 1 | x^{(i)}, w, b)}{p(Y = -1 | x^{(i)}, w, b)} + \ln p(Y = -1 | x^{(i)}, w, b)\end{aligned}$$

# Learning the Weights



$$\begin{aligned}\ell(w, b) &= \ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \ln p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln p(Y = 1 | x^{(i)}, w, b) + \left(1 - \frac{y^{(i)} + 1}{2}\right) \ln p(Y = -1 | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln \frac{p(Y = 1 | x^{(i)}, w, b)}{p(Y = -1 | x^{(i)}, w, b)} + \ln p(Y = -1 | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} (w^T x^{(i)} + b) - \ln(1 + \exp(w^T x^{(i)} + b))\end{aligned}$$

# Learning the Weights



$$\begin{aligned}\ell(w, b) &= \ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \ln p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln p(Y = 1 | x^{(i)}, w, b) + \left(1 - \frac{y^{(i)} + 1}{2}\right) \ln p(Y = -1 | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln \frac{p(Y = 1 | x^{(i)}, w, b)}{p(Y = -1 | x^{(i)}, w, b)} + \ln p(Y = -1 | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} (w^T x^{(i)} + b) - \ln(1 + \exp(w^T x^{(i)} + b))\end{aligned}$$

This is concave in  $w$  and  $b$ : take derivatives and solve!

# Learning the Weights



$$\begin{aligned}\ell(w, b) &= \ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \ln p(y^{(i)} | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln p(Y = 1 | x^{(i)}, w, b) + \left(1 - \frac{y^{(i)} + 1}{2}\right) \ln p(Y = -1 | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} \ln \frac{p(Y = 1 | x^{(i)}, w, b)}{p(Y = -1 | x^{(i)}, w, b)} + \ln p(Y = -1 | x^{(i)}, w, b) \\ &= \sum_{i=1}^N \frac{y^{(i)} + 1}{2} (w^T x^{(i)} + b) - \ln(1 + \exp(w^T x^{(i)} + b))\end{aligned}$$

No closed form solution ☹

$$\ell(w, b) = \sum_{i=1}^N \frac{y^{(i)} + 1}{2} (w^T x^{(i)} + b) - \ln(1 + \exp(w^T x^{(i)} + b))$$

The above is the Likelihood  
which we maximize!

- Can apply gradient **ascent** to maximize the conditional likelihood

$$\frac{\partial \ell}{\partial b} = \sum_{i=1}^N \left[ \frac{y^{(i)} + 1}{2} - p(Y = 1 | x^{(i)}, w, b) \right]$$

$$\frac{\partial \ell}{\partial w_j} = \sum_{i=1}^N x_j^{(i)} \left[ \frac{y^{(i)} + 1}{2} - p(Y = 1 | x^{(i)}, w, b) \right]$$



- Can define priors on the weights to prevent overfitting
  - Normal distribution, zero mean, identity covariance

$$p(w) = \prod_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{w_j^2}{2\sigma^2}\right)$$

- “Pushes” parameters towards zero
- Regularization
  - Helps avoid very large weights and overfitting

- The log-MAP objective with this Gaussian prior is then

$$\ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) p(w) p(b) = \left[ \sum_i^N \ln p(y^{(i)} | x^{(i)}, w, b) \right] - \frac{\lambda}{2} \|w\|_2^2$$

- Quadratic penalty: drives weights towards zero
- Adds a negative linear term to the gradients
- Different priors can produce different kinds of regularization

- The log-MAP objective with this Gaussian prior is then

$$\ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) p(w) p(b) = \left[ \sum_i^N \ln p(y^{(i)} | x^{(i)}, w, b) \right] - \frac{\lambda}{2} \|w\|_2^2$$

- Quadratic penalty: drives weights towards zero
- Adds a negative linear term to the gradients
- Different priors can produce different kinds of regularization



$\ell_2$  regularizer

# L2 vs L1 Regularization



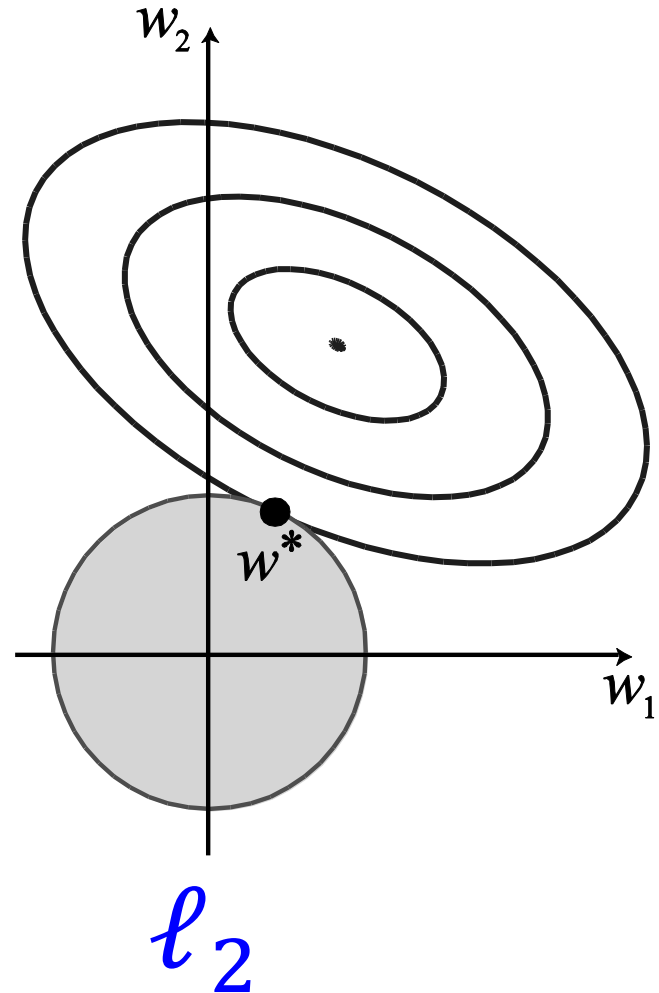
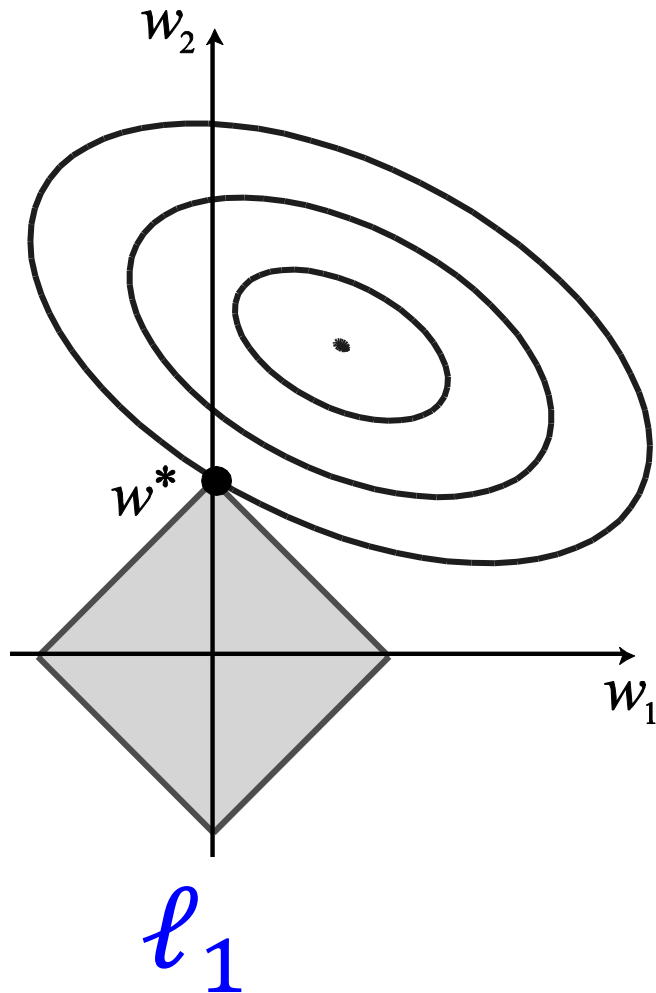
The Likelihood with L2 Regularization:

$$\ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) p(w) p(b) = \left[ \sum_i^N \ln p(y^{(i)} | x^{(i)}, w, b) \right] - \frac{\lambda}{2} \|w\|_2^2$$

Alternate formulation is L1 Regularization:

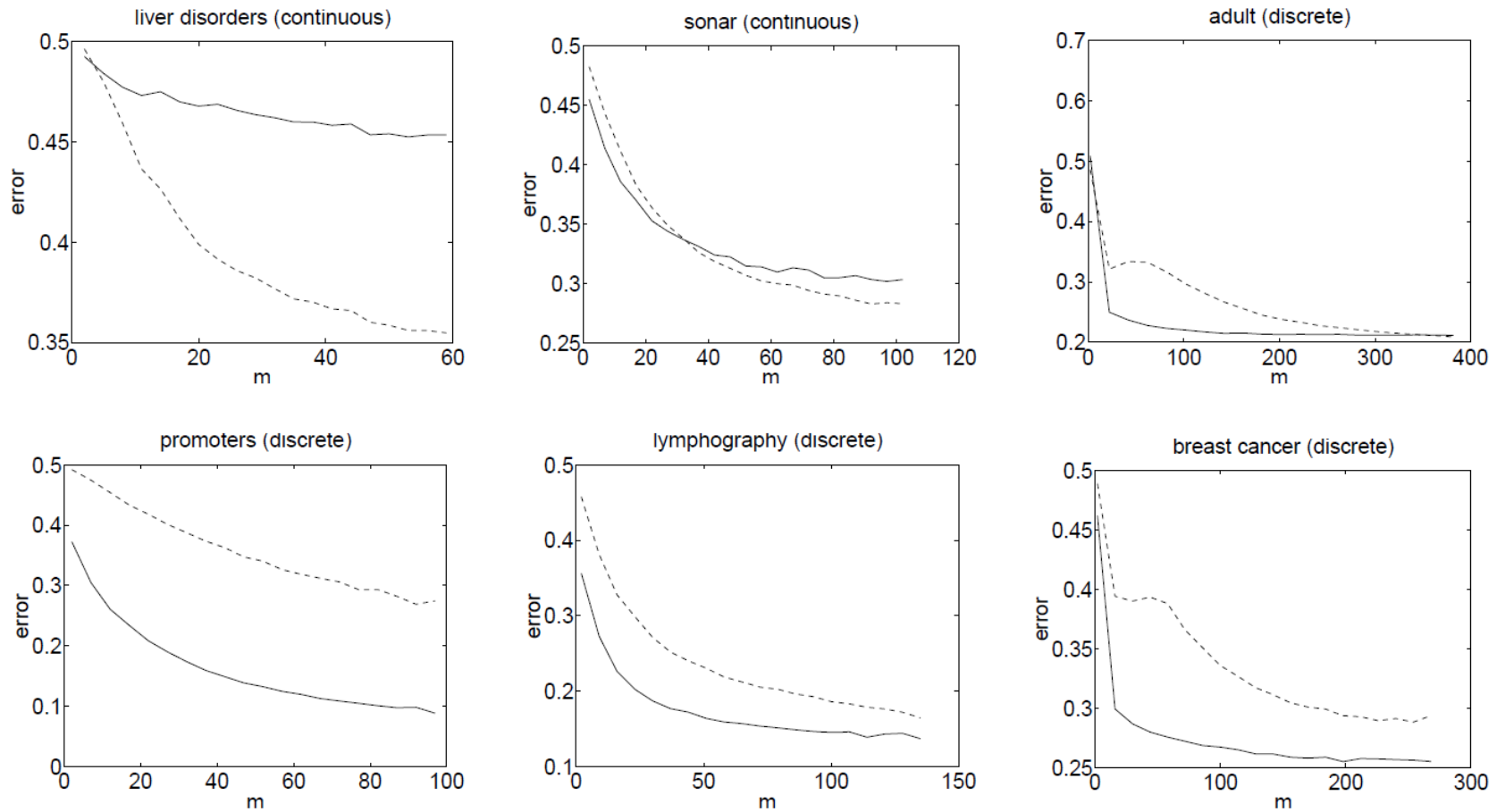
$$\ln \prod_{i=1}^N p(y^{(i)} | x^{(i)}, w, b) p(w) p(b) = \left[ \sum_i^N \ln p(y^{(i)} | x^{(i)}, w, b) \right] - \frac{\lambda}{2} \|w\|_1$$

# Regularization



- Non-asymptotic analysis (for Gaussian NB)
  - Convergence rate of parameter estimates as size of training data tends to infinity ( $n = \#$  of attributes in  $X$ )
    - Naïve Bayes needs  $O(\log n)$  samples
      - NB converges quickly to its (perhaps less helpful) asymptotic estimates
  - Logistic Regression needs  $O(n)$  samples
    - LR converges more slowly but makes no independence assumptions (typically less biased)

# NB vs. LR (on UCI datasets)



— Naïve bayes  
..... Logistic Regression

Sample size  $m$

- Suppose that  $y \in \{1, \dots, R\}$ , i.e., that there are  $R$  different class labels
- Can define a collection of weights and biases as follows
  - Choose a vector of biases and a matrix of weights such that for  $y \neq R$

$$p(Y = k|x) = \frac{\exp(b_k + \sum_i w_{ki}x_i)}{1 + \sum_{j < R} \exp(b_j + \sum_i w_{ji}x_i)}$$

and

$$p(Y = R|x) = \frac{1}{1 + \sum_{j < R} \exp(b_j + \sum_i w_{ji}x_i)}$$