

Logistic Regression

Lyle Ungar

Learning objectives

Logistic model & loss

Decision boundaries as hyperplanes

Multi-class regression

What do you do with a binary y ?

◆ Can you use linear regression?

- $y = \mathbf{w}^T \mathbf{x}$

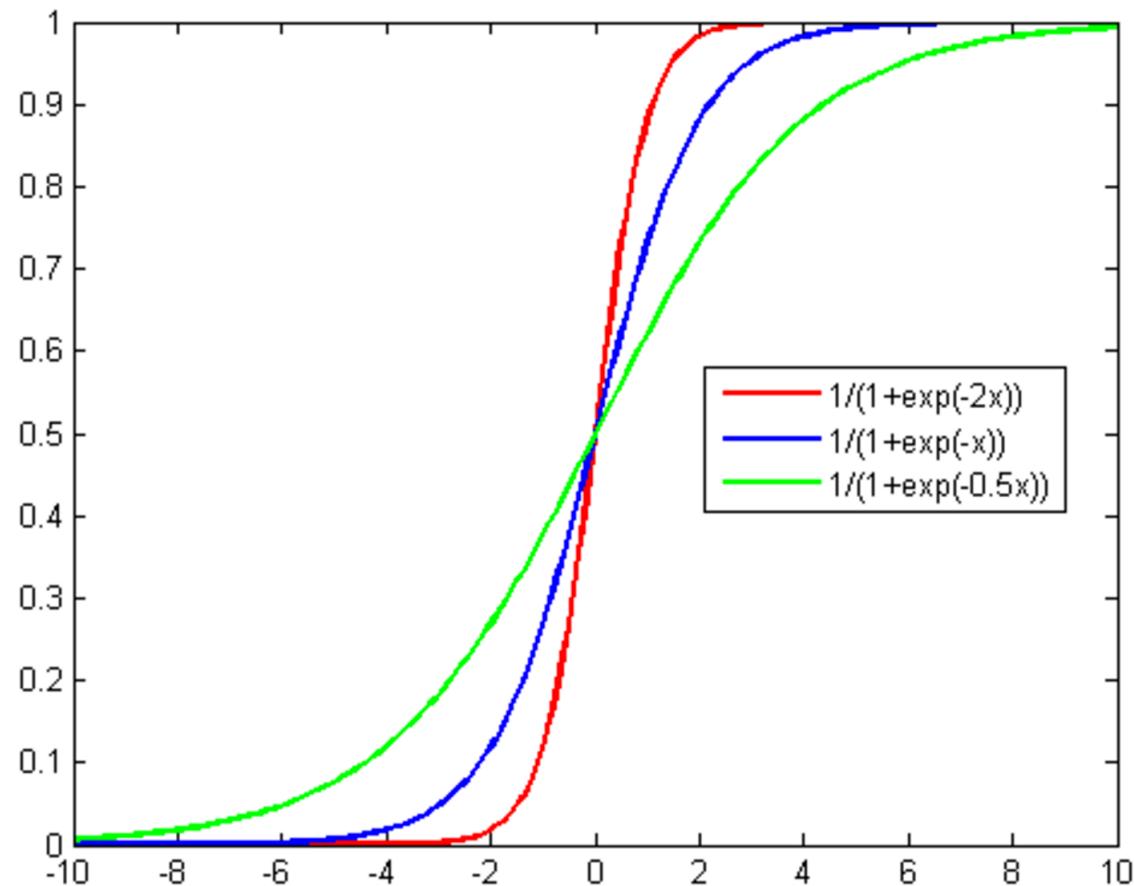
◆ How about a different link function?

- $y = f(\mathbf{w}^T \mathbf{x})$

◆ Or a different probability distribution

- $P(y=1|\mathbf{x}) = f(\mathbf{w}^T \mathbf{x})$

Logistic function



$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Logistic Regression

$$P(Y = 1 | \mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp\{-\sum_j w_j x_j\}} = \frac{1}{1 + \exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{1}{1 + \exp\{-y \mathbf{w}^\top \mathbf{x}\}}$$

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

$$\log\left(\frac{P(Y=1|\mathbf{x},\mathbf{w})}{P(Y=-1|\mathbf{x},\mathbf{w})}\right) = \mathbf{w}^\top \mathbf{x}$$

Log odds

Log likelihood of data

$$\begin{aligned}\log(P(D_Y|D_X, \mathbf{w})) &= \log \left(\prod_i \frac{1}{1 + \exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\}} \right) \\ &= - \sum_i \log(1 + \exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\})\end{aligned}$$

$y = 1$ or -1

Decision Boundary

$$P(Y = 1 | \mathbf{x}, \mathbf{w}) = P(Y = -1 | \mathbf{x}, \mathbf{w})$$

$$\frac{1}{1 + \exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{\exp\{-\mathbf{w}^\top \mathbf{x}\}}{1 + \exp\{-\mathbf{w}^\top \mathbf{x}\}}$$

$$\mathbf{w}^\top \mathbf{x} = 0$$

Representing Hyperplanes

- How do we represent a line?

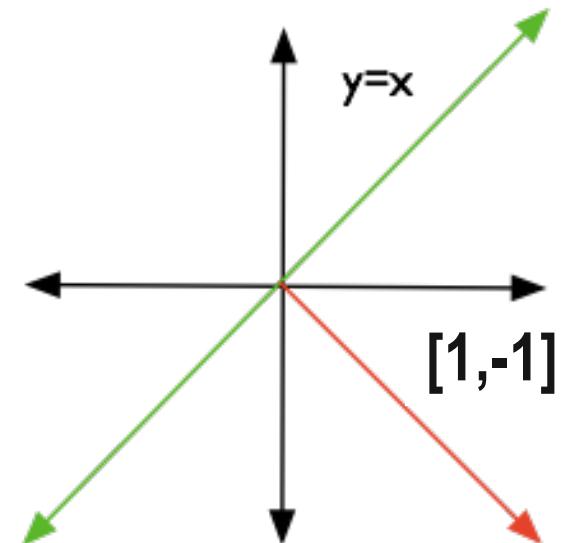
$$y = x$$

$$0 = x - y$$

$$0 = [1, -1] \begin{bmatrix} x \\ y \end{bmatrix}$$

- In general, a hyperplane is defined by

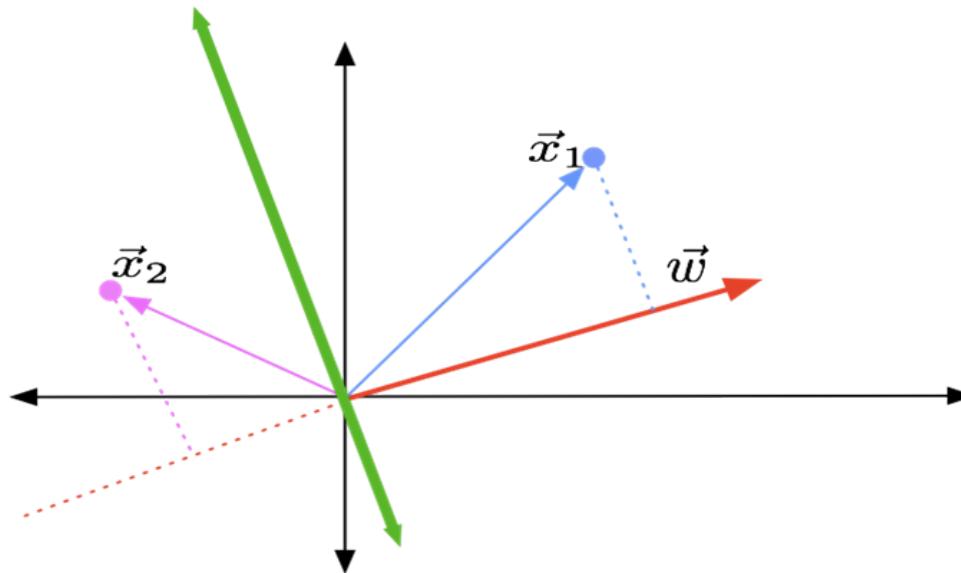
$$0 = w^T x$$



The red vector (w) defines the green hyper plane that is orthogonal to it.

Why bother with this weird representation?

Projections



$(\vec{w} \cdot \vec{x})\vec{w}$ is the projection of \vec{x} onto \vec{w}

Now classification is easy!

$$h(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x})$$

Computing MLE

- ◆ Use gradient ascent

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t \nabla_{\mathbf{w}} \ell(\mathbf{w})$$

Loss function =
log-likelihood

$$\nabla_{\mathbf{w}} \ell(\mathbf{w}) = \frac{\delta \log(P(D_Y|D_X, \mathbf{w}))}{\delta \mathbf{w}} = \sum_i \frac{y_i \mathbf{x}_i \exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\}}{1 + \exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\}} = \sum_i y_i \mathbf{x}_i (1 - P(y_i | \mathbf{x}_i, \mathbf{w}))$$

Computing MAP

- ◆ Prior

$$w_j \sim \mathcal{N}(0, \gamma^2) \text{ so } P(\mathbf{w}) = \prod_j \frac{1}{\gamma\sqrt{2\pi}} \exp\left\{-\frac{w_j^2}{2\gamma^2}\right\}$$

- ◆ So solve

$$\arg \max_{\mathbf{w}} \log P(\mathbf{w} \mid D, \gamma) = \arg \max_{\mathbf{w}} (\ell(\mathbf{w}) + \log P(\mathbf{w} \mid \gamma))$$

$$\arg \max_{\mathbf{w}} \left(\ell(\mathbf{w}) - \frac{1}{2\gamma^2} \mathbf{w}^\top \mathbf{w} \right)$$

- ◆ Again use gradient descent



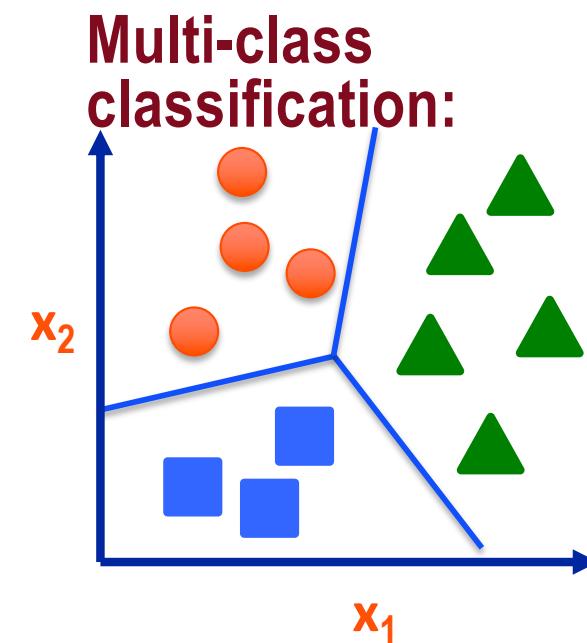
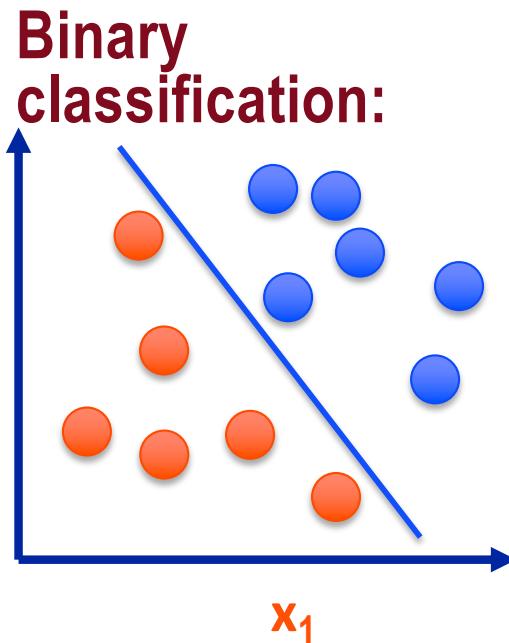
Questions?

Top



Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

Multi-Class Classification



Disease diagnosis:

healthy / cold / flu / pneumonia

Object classification:

desk / chair / monitor / bookcase

Multi-Class Logistic Regression

◆ For 2 classes:

$$h_{\theta}(x) = \frac{1}{1 + \exp(-\theta^T x)} = \frac{\exp(\theta^T x)}{1 + \exp(\theta^T x)}$$



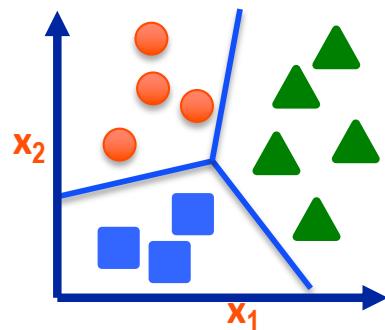
weight assigned to $y = -1$ weight assigned to $y = 1$

◆ For K classes:

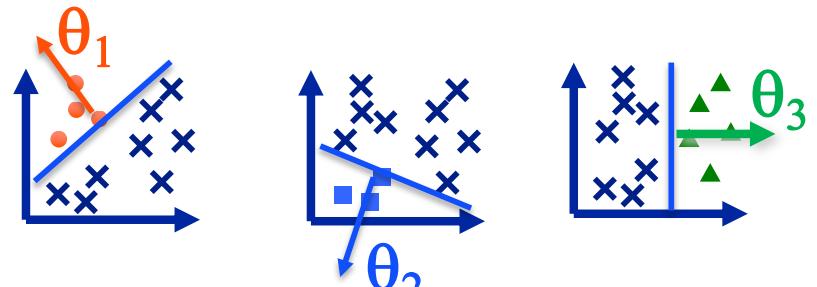
$$p(y = k \mid \mathbf{x}; \theta_1, \dots, \theta_k) = \frac{\exp(\theta_k^\top \mathbf{x})}{\sum_{k=1}^K \exp(\theta_k^\top \mathbf{x})}$$

- Called the **softmax** function
 - maps a vector to a probability distribution

Multi-Class Logistic Regression



Split into One vs. Rest:



- ◆ Train a logistic regression classifier for each class k to predict the probability that $y = k$ with

$$h_k(\mathbf{x}) = \frac{\exp(\theta_k^\top \mathbf{x})}{\sum_{k=1}^K \exp(\theta_k^\top \mathbf{x})}$$

Implementing Multi-Class Logistic Regression

- ◆ **P(y=k|x) estimated by:** $h_k(\mathbf{x}) = \frac{\exp(\theta_k^\top \mathbf{x})}{\sum_{k=1}^K \exp(\theta_k^\top \mathbf{x})}$
- ◆ **Gradient descent simultaneously updates all parameters for all models**
 - Same derivative as before, just with the above $h_k(\mathbf{x})$
- ◆ **Predict class label as the most probable label**

You should know

- ◆ *Logistic model & loss*
 - *Linear in log-odds*
- ◆ *Decision boundaries*
 - *hyperplane*
- ◆ *Softmax*
 - *Maps vector to probability distribution*



Questions?

Top



Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app