CS60050: Machine Learning

Autumn 2024

Sudeshna Sarkar

Linear Models for Classifciation

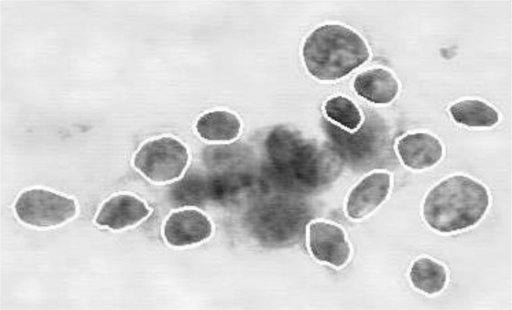
Logistic Regression

1 August 2024

Slides from Pat Virtue, CMU 10-315 Introduction to ML

Example: Breast cancer classification

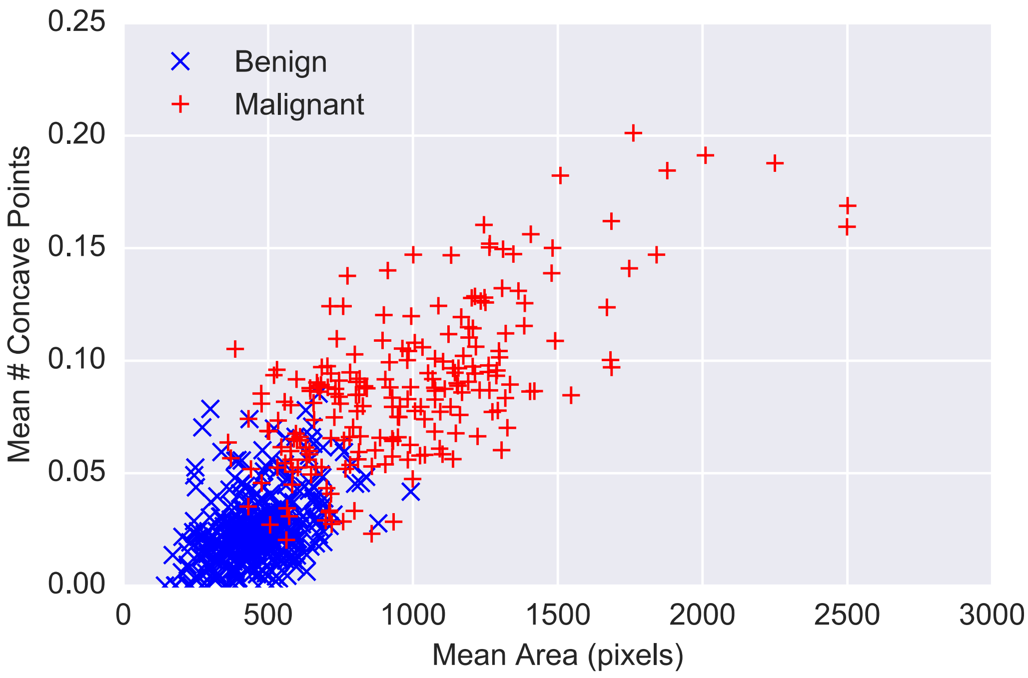
- Well-known classification example: using machine learning to diagnose whether a breast tumor is benign or malignant [Street et al., 1992]
- Setting: doctor extracts a sample of fluid from tumor, stains cells, then outlines several of the cells (image processing refines outline)



System computes features for each cell such as area, perimeter, concavity, texture (10 total); computes mean/std/max for all features

Example: Breast cancer classification

Plot of two features: mean area vs. mean concave points, for two classes

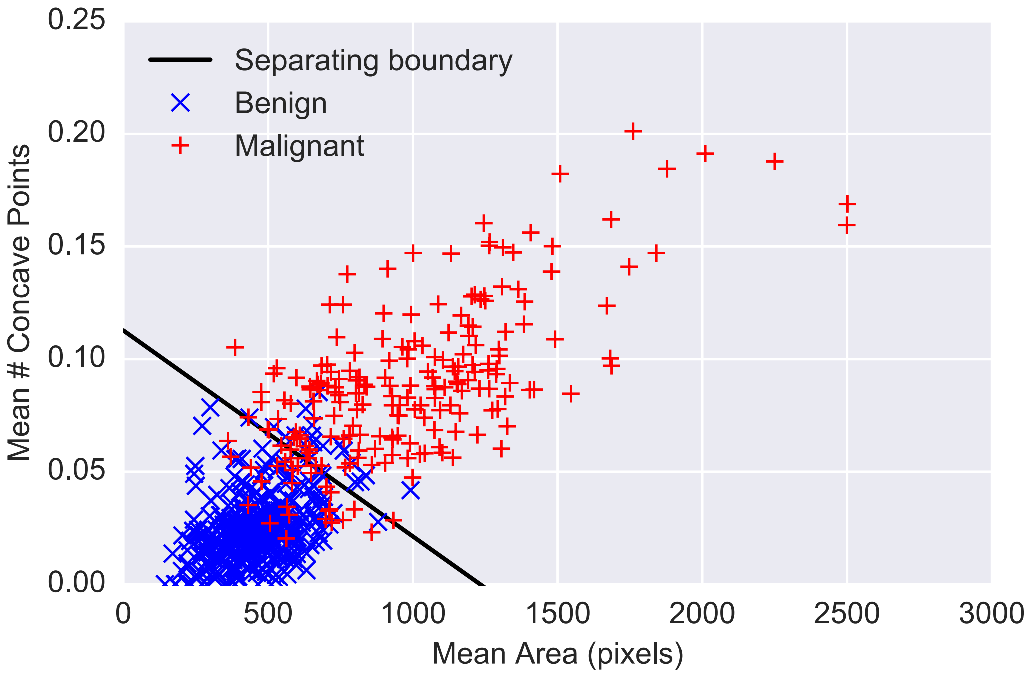


Slide credit: CMU AI Zico Kolter

Slides from Pat Virtue, CMU 10-315 Introduction to ML

Linear classification example

Linear classification: linear decision boundary

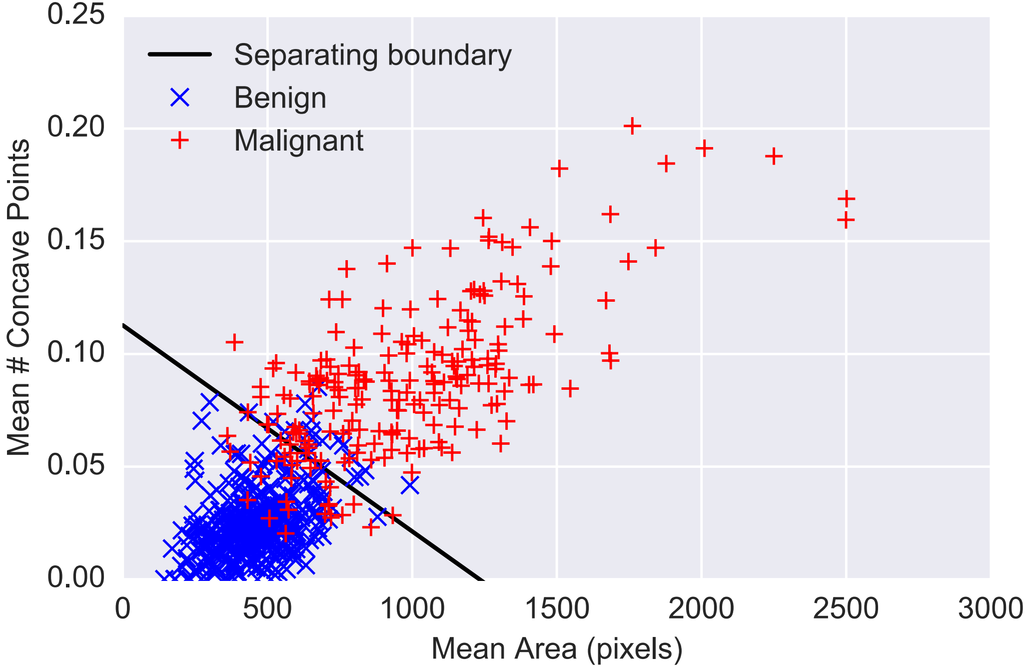


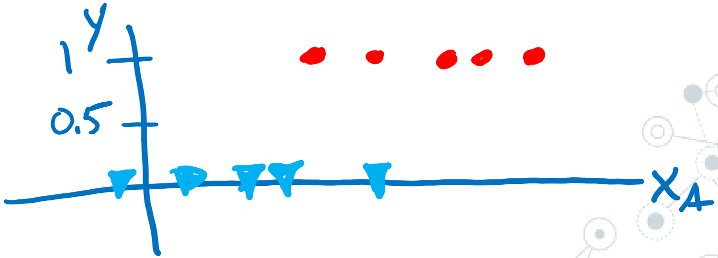
Slide credit: CMU AI Zico Kolter

Slides from Pat Virtue, CMU 10-315 Introduction to ML

Logistic regression for classification

- Linear classification: linear decision boundary
- Probabilistic classification: provide $P(Y = 1 \mid x)$ rather than just $\hat{y} \in \{0, 1\}$



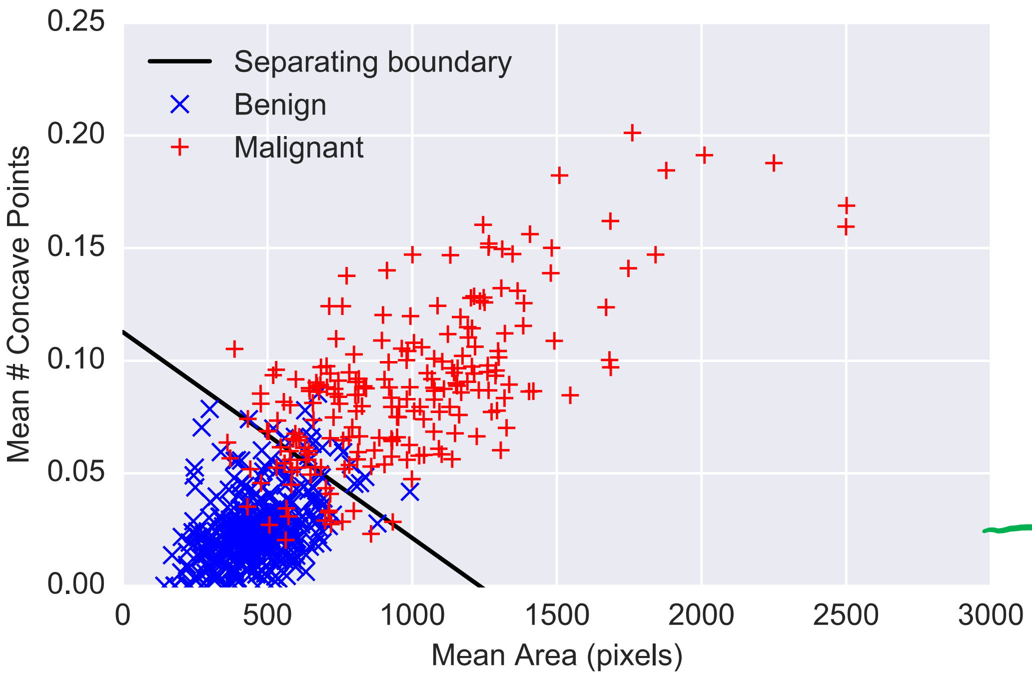


Slides from Pat Virtue, CMU 10-315 Introduction to ML

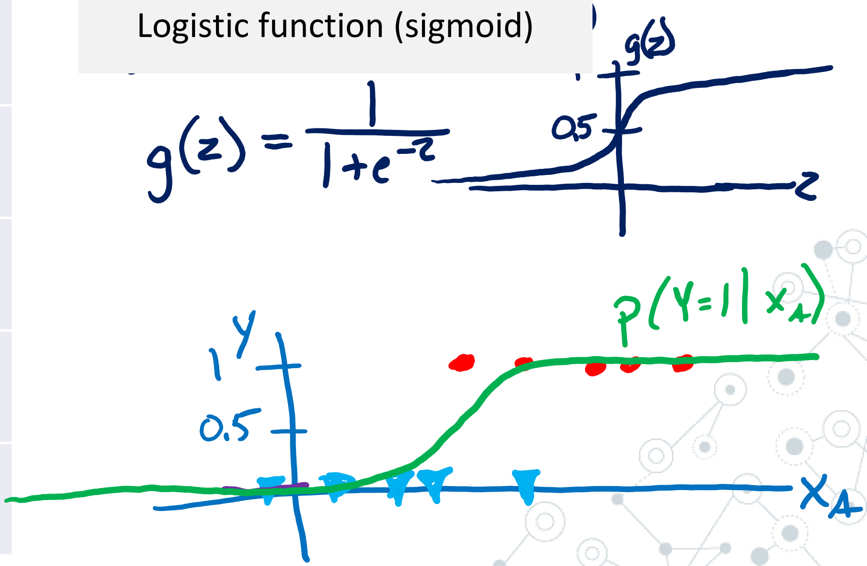
Slide credit: CMU AI Zico Kolter

Logistic regression for classification

- Linear classification: linear decision boundary
- Probabilistic classification: provide $P(Y = 1 \mid x)$ rather than just $\hat{y} \in \{0, 1\}$

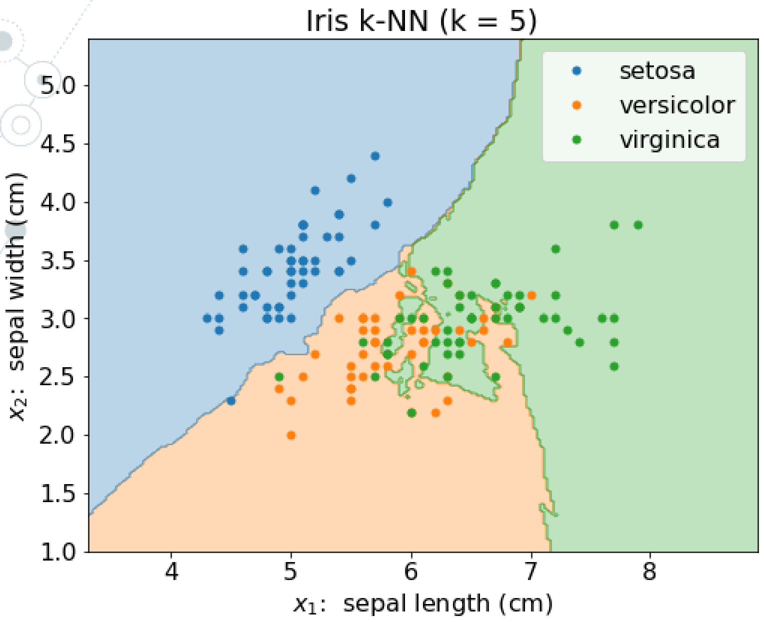


Logistic function (sigmoid)



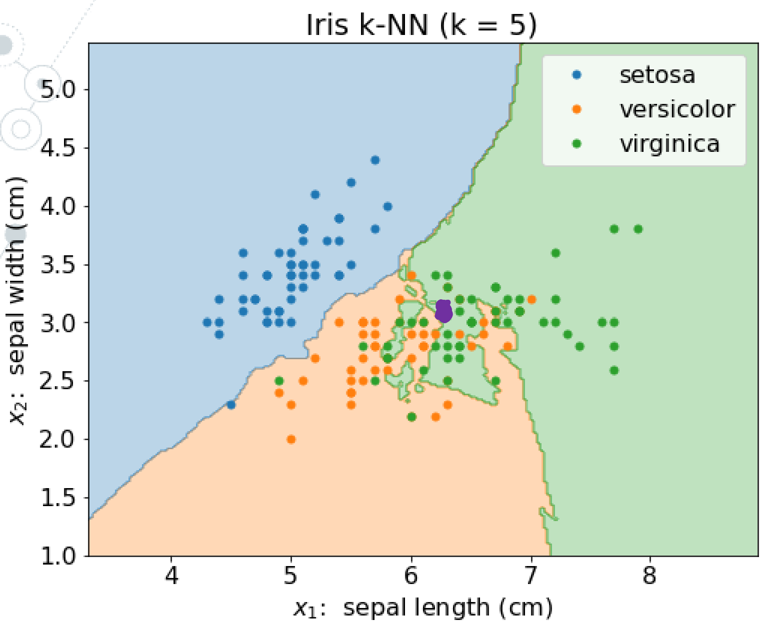
Classification Decisions

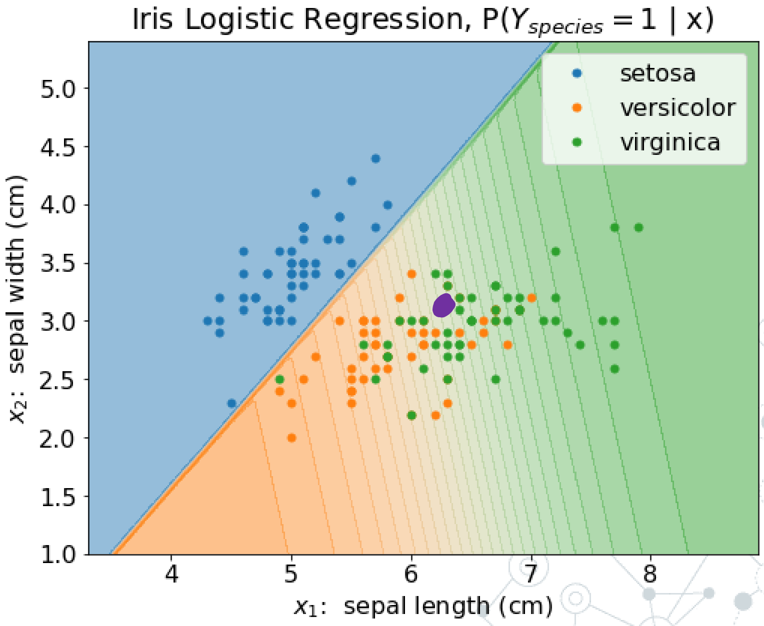
Predicting one specific class is troubling, especially when we know that there is some uncertainty in our prediction



Classification Probability

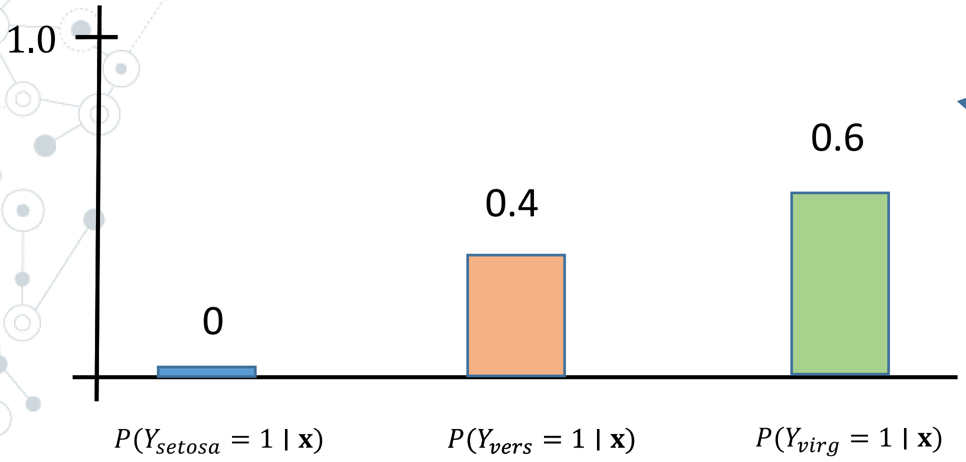
- Constructing a model than can return the probability of the output being a specific class could be incredibly useful

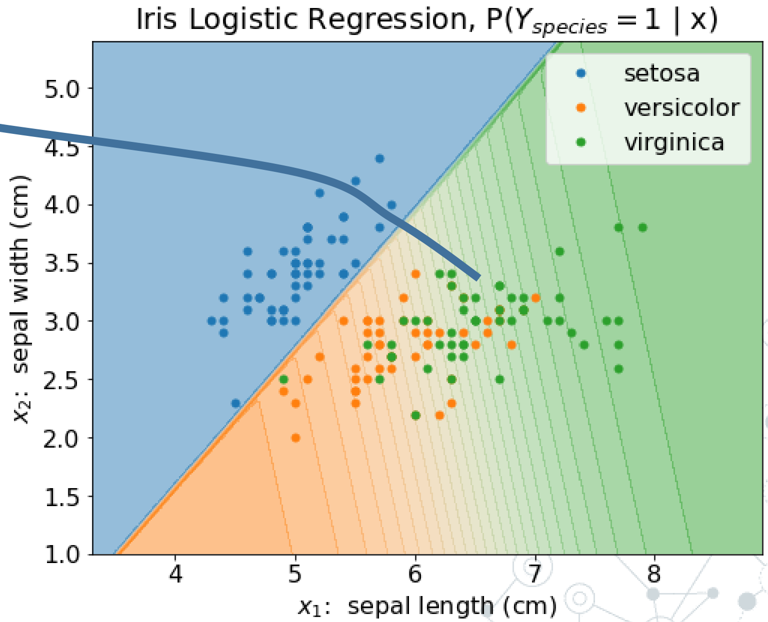




Classification Probability

- Constructing a model than can return the probability of the output being a specific class could be incredibly useful





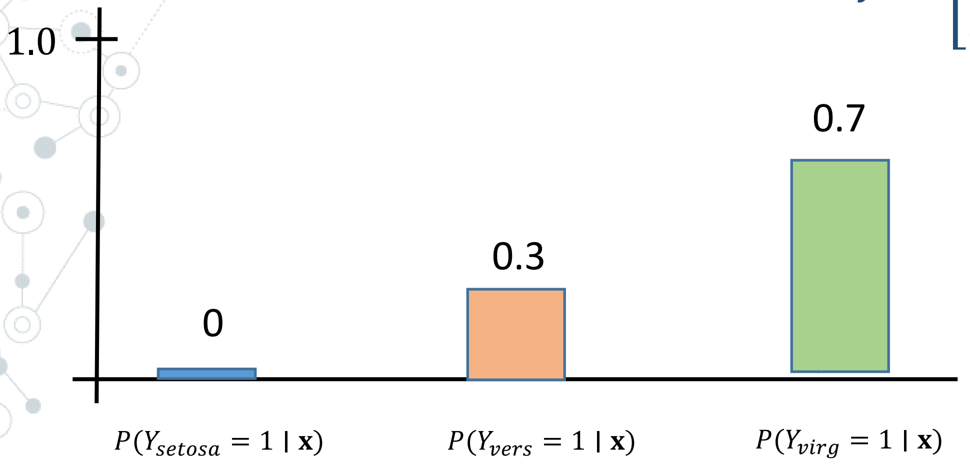
We can still make decisions, .e.g,

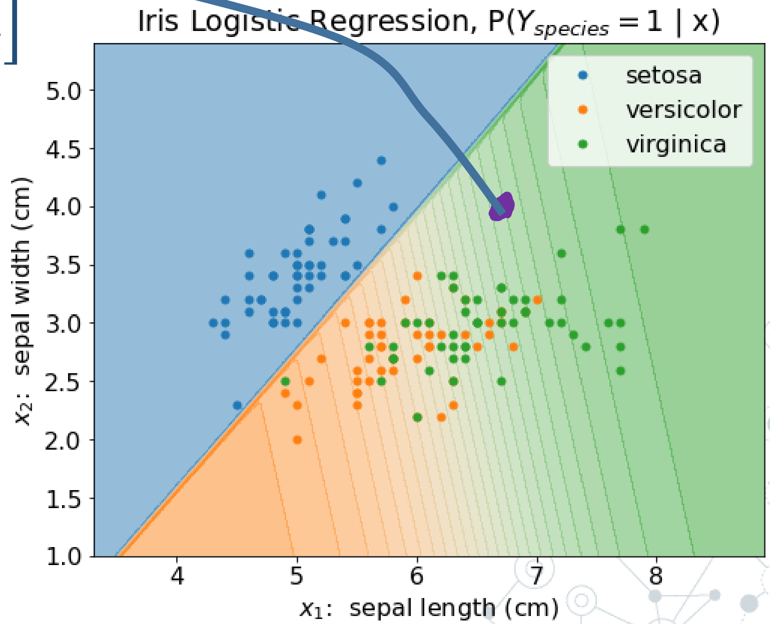
$$\underset{k}{\operatorname{argmax}}\, P(Y_k = 1 \mid \mathbf{x})$$

Loss for Probability Distributions

- We need a way to compare how good/bad each prediction is

$$\hat{y} = \begin{bmatrix} 0.0 \\ 0.3 \\ 0.7 \end{bmatrix}$$





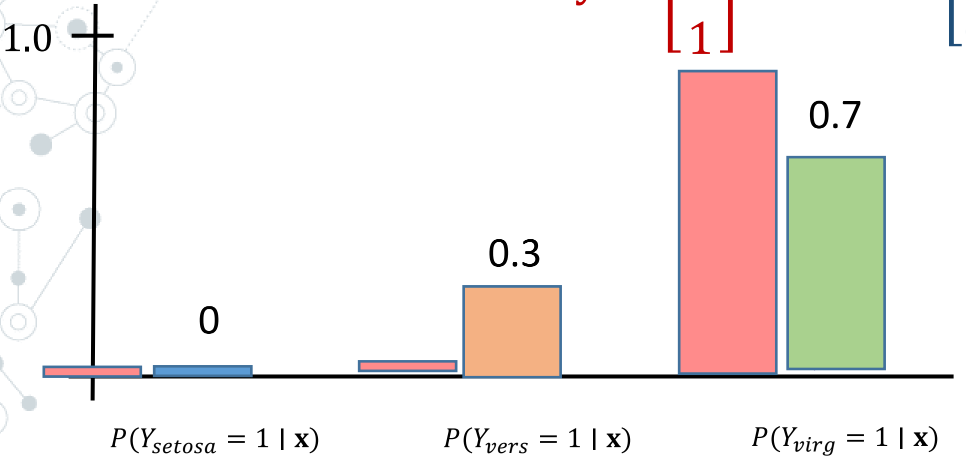
Cross-entropy loss

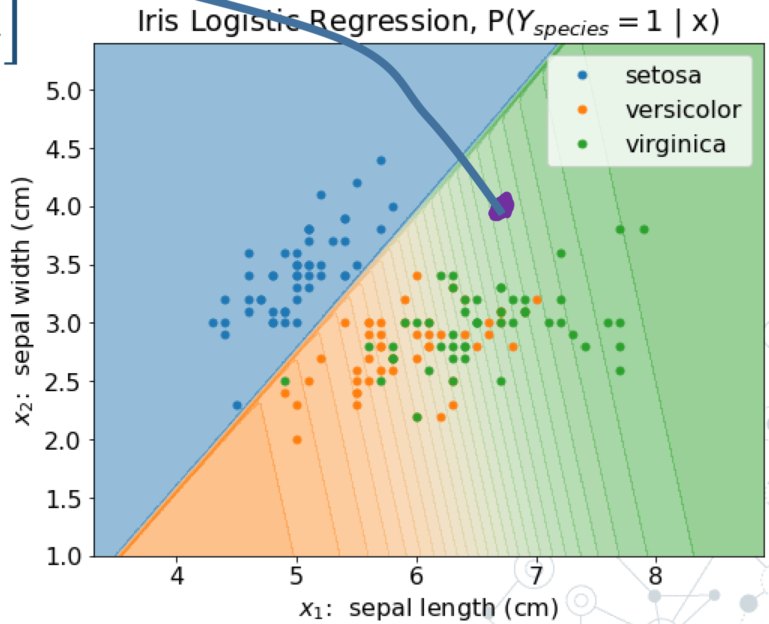
$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$

Loss for Probability Distributions

- We need a way to compare how good/bad each prediction is

$$y = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \hat{y} = \begin{bmatrix} 0.0 \\ 0.3 \\ 0.7 \end{bmatrix}$$





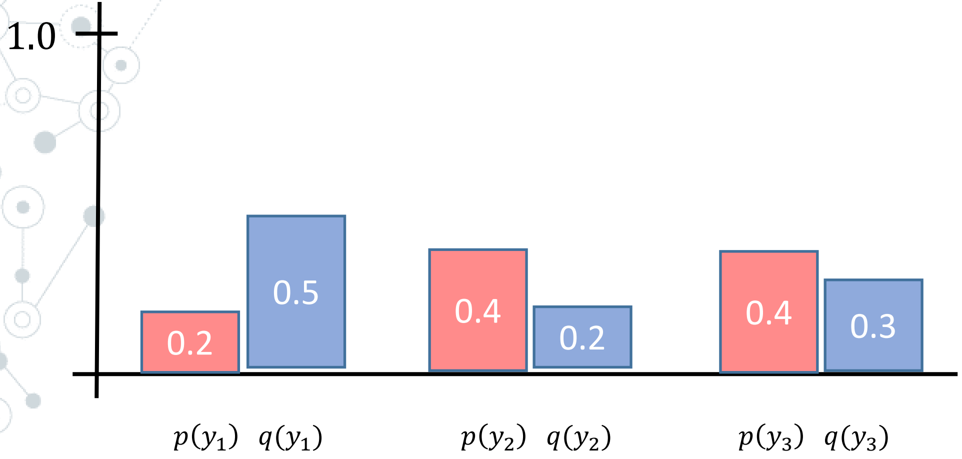
Cross-entropy loss

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$

Loss for Probability Distributions

Cross-entropy more generally is a way to compare any two probability distributions*

*when used in logistic regression
$\mathbf{y}$ is always a one-hot vector

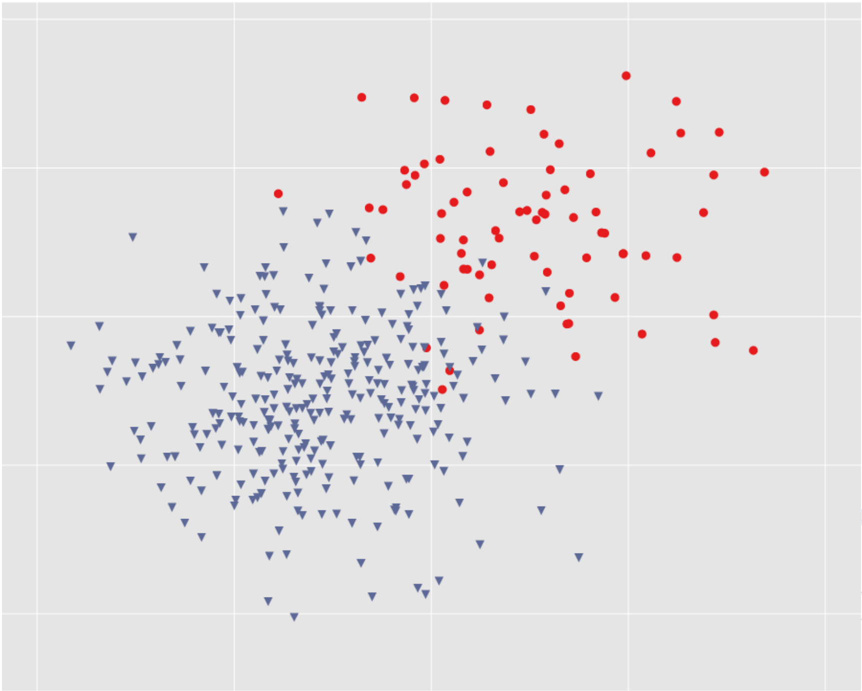


Cross-entropy loss

$$H(P, Q) = -\sum_{k=1}^{K} p(y_k) \log q(y_k)$$

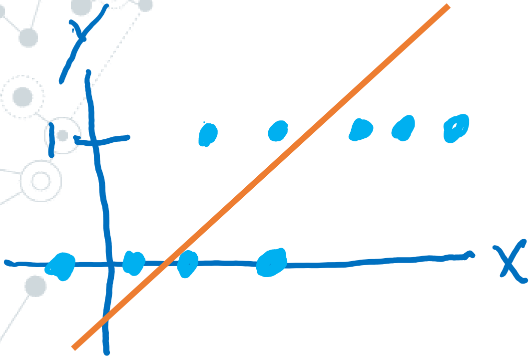
Linear models for classification

Learn to predict if a patient has cancer ($Y = 1$) or not ($Y = 0$) given the input of two test results, X_A and X_B.



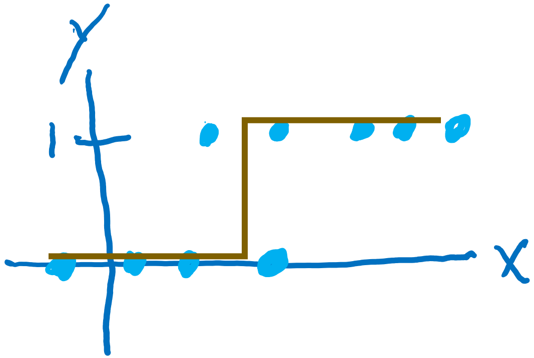
Building on a Linear Model

Linear



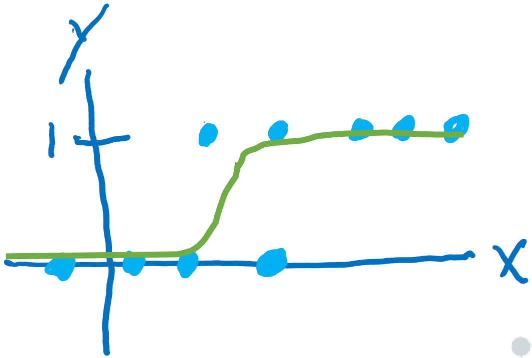
$\hat{y} = \theta^T X$

Thresholded Linear



$\hat{y} = g_{\text{thres}}(\theta^T X)$

Logistic Linear



$\hat{y} = g_{\text{logistic}}(\theta^T X)$

Logistic Regression

Linear model for classification

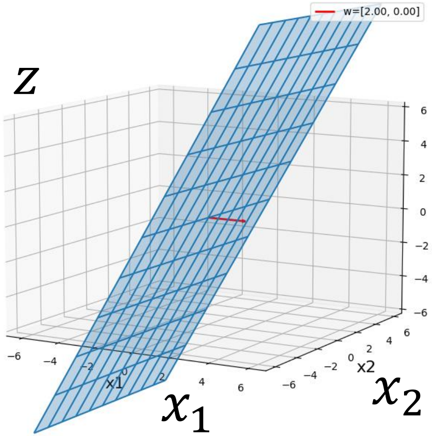
(with multiple input features)

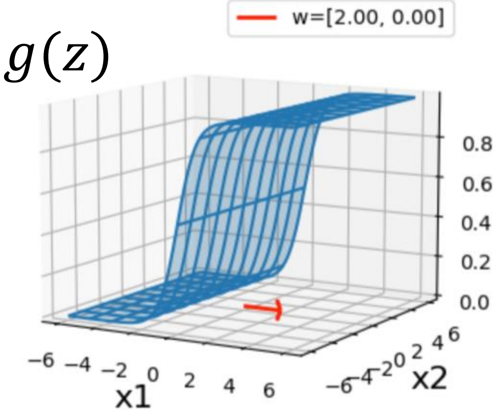
Building on a Linear Model

- With two input features, $\mathbf{x} = [x_1 \;\; x_2]^\top$, we have two weight parameters and one bias parameter, $\mathbf{w} = [\theta_1 \;\; \theta_2]^\top$ and b (θ_0), that control the slope and vertical offset of the following plane:

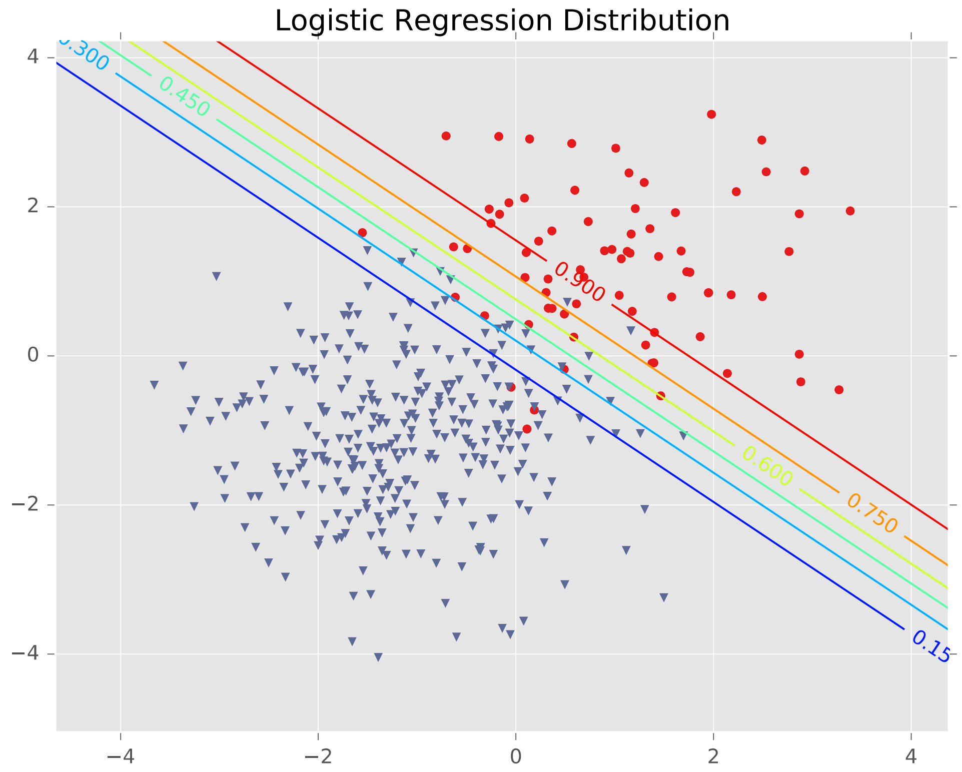
$$z = \boldsymbol{\theta}^\top \mathbf{x} + \theta_0$$

- The sigmoid function $\hat{y} = g(z)$ then squashed the plane such that any z values going to $+\infty$ go to 1 and z values going to $-\infty$ go to 0





Building on a Linear Model



Slide credit: CMU MLD Matt Gormley

Linear in Higher Dimensions

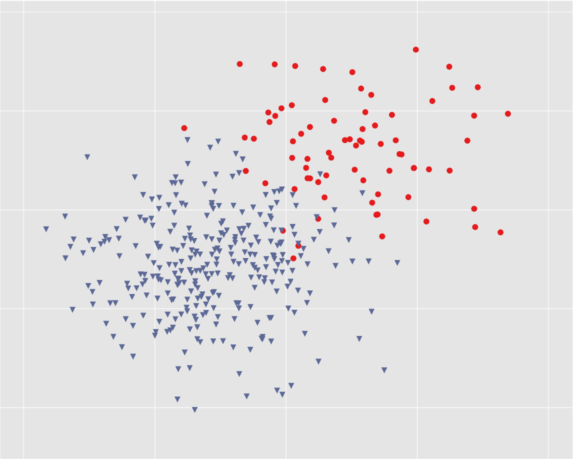
- What are these linear shapes called for 1-D, 2-D, 3-D, M-D input?

	$\boldsymbol{x} \in \mathbb{R}$	$\boldsymbol{x} \in \mathbb{R}^2$	$\boldsymbol{x} \in \mathbb{R}^3$	$\boldsymbol{x} \in \mathbb{R}^M$
$y = \boldsymbol{w}^T \boldsymbol{x} + b$	line	plane	hyperplane	hyperplane
$\boldsymbol{w}^T \boldsymbol{x} + b = 0$	point	line	plane	hyperplane
$\boldsymbol{w}^T \boldsymbol{x} + b \geq 0$	halfline	halfplane	halfspace	halfspace

Optimizing a Model for Cancer Diagnosis

Learn to predict if a patient has cancer ($Y = 1$) or not ($Y = 0$) given the input of two test results, X_A, X_B. Note: bias term included in $\mathbf{x}$.

$$p(\,Y = 1 \mid \mathbf{x}, \boldsymbol{\theta}\,) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

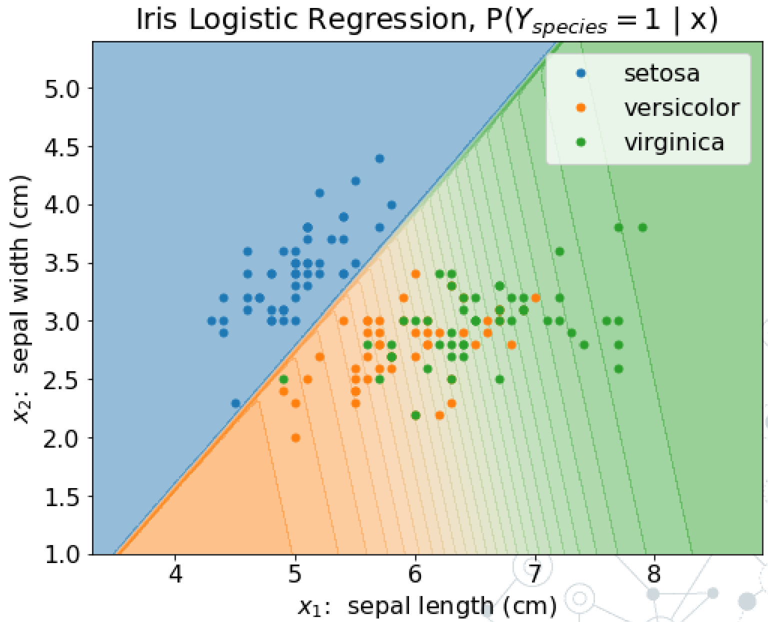


Empirical Risk Minimization

- Still doing empirical risk minimization, just with a cross-entropy loss

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \ \hat{R}(h)$$

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(y^{(i)}, h\left(x^{(i)}\right)\right)$$



Empirical Risk Minimization

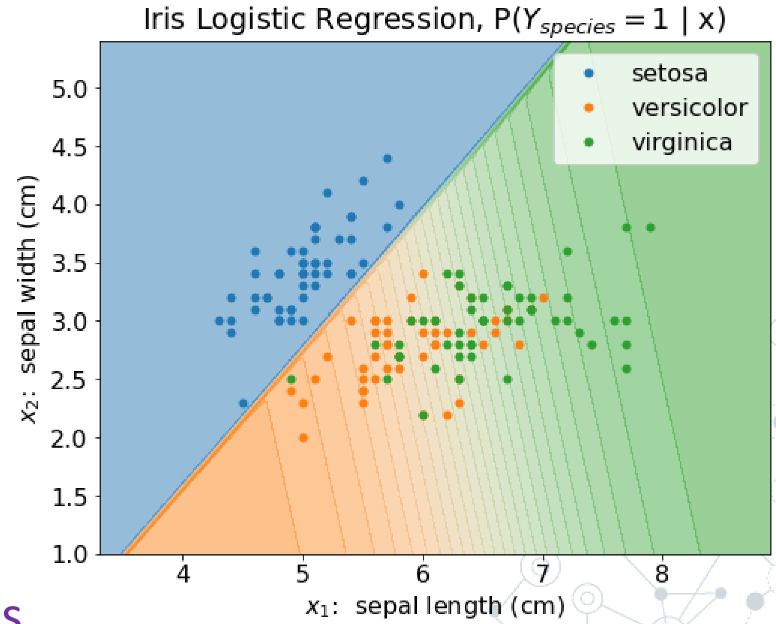
- Still doing empirical risk minimization, just with a cross-entropy loss

$$h^* = \underset{h \in \mathcal{H}}{\text{argmin}} \ \hat{R}(h)$$

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} l\big(Y_i, h(X_i)\big)$$

Cross-entropy loss

$\ell(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \ \log \hat{y}_k$



But now we need a model $h_{\boldsymbol{\theta}}(\mathbf{x})$ that returns values that look like probabilities

CS60050: Machine Learning
Autumn 2024

Sudeshna Sarkar

Linear Models for Classification

Logistic Regression contd

2 August 2024

Slides from Pat Virtue, CMU 10-315 Introduction to ML

Binary Logistic Regression

$$g(z) = \frac{1}{1 + e^{-z}}$$

- Objective: Special case for binary logistic regression

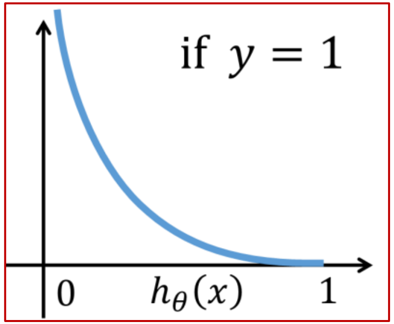
1) Model $\quad \hat{y} = g(\boldsymbol{\theta}^T \mathbf{x})$

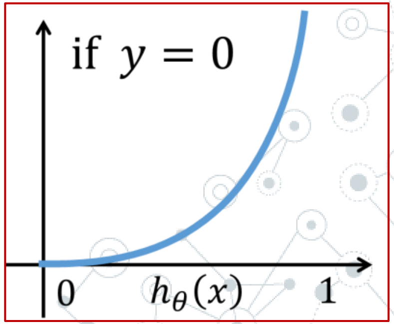
1) Objective function

$$J(\boldsymbol{\theta}) = -\frac{1}{m} \sum_i \sum_k y_k^{(i)} \log y_k^{(i)}$$

$$= -\frac{1}{m} \sum_i \left(y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right) \right)$$

1) Solve for $\widehat{\boldsymbol{\theta}}$





Solve Logistic Regression

$$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x}) \qquad g(z) = \frac{1}{1 + e^{-z}} \qquad \frac{dg}{dz} = g(z)\big(1 - g(z)\big)$$

$$J^{(i)}(\boldsymbol{\theta}) = -\left[y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right]$$

$$\frac{\partial J^{(i)}}{\partial \boldsymbol{\theta}} = -\left(y^{(i)} - \hat{y}^{(i)}\right) \mathbf{x}^{(i)}$$

Solve Logistic Regression

$$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x}) \qquad g(z) = \frac{1}{1+e^{-z}} \qquad \frac{dg}{dz} = g(z)\big(1 - g(z)\big)$$

$$J^{(i)}(\boldsymbol{\theta}) = -\left[y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right]$$

$$\frac{\partial J^{(i)}}{\partial \boldsymbol{\theta}} = -\left(y^{(i)} - \hat{y}^{(i)}\right) \mathbf{x}^{(i)}$$

Solve Logistic Regression

$$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x}) \qquad g(z) = \frac{1}{1+e^{-z}}$$

$$J^{(i)}(\boldsymbol{\theta}) = -\left[y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right]$$

$$J(\boldsymbol{\theta}) = -\frac{1}{N} \sum_i \left(y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right)$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\frac{1}{N} \sum_i \left(y^{(i)} - \hat{y}^{(i)}\right) \mathbf{x}^{(i)}$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0?$$

No closed form solution ☹

Back to iterative methods. Solve with (stochastic) gradient descent, Newton’s method, or Iteratively Reweighted Least Squares (IRLS)

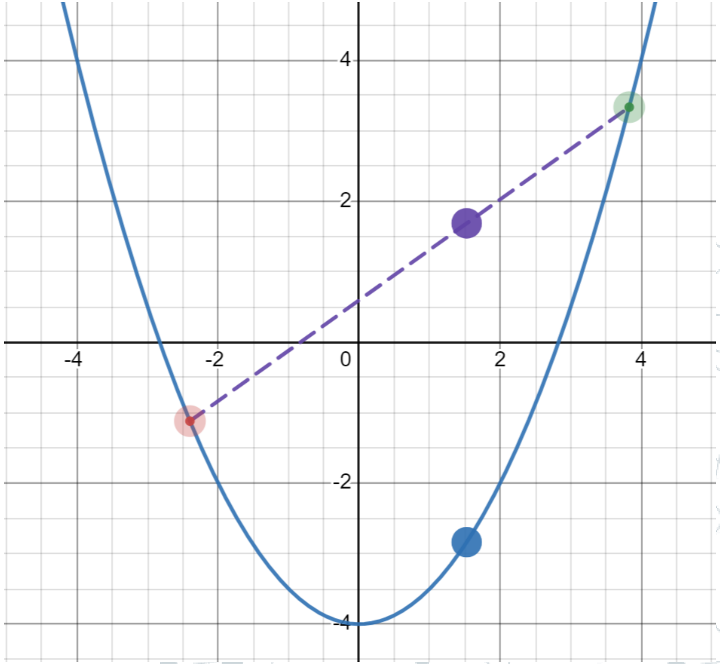
Good news: The logistic regression optimization function is convex!

Logistic Regression

Convexity

Optimization

- Convex function
- If $f(\boldsymbol{x})$ is convex, then:
  - $f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{z}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{z})$ $\forall\ 0 \leq \alpha \leq 1$



Optimization

- Convex function
- If $f(\boldsymbol{x})$ is convex, then:
  - $f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{z}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{z}) \quad \forall\, 0 \leq \alpha \leq 1$

Convex optimization

If second derivative is ≥ 0 everywhere then function is convex

If $f(\boldsymbol{x})$ is convex, then:

- Every local minimum is also a global minimum ☺

Optimization

Is $h(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{x} + b)$ convex?

But...what are we optimizing over in logistic regression?

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\sum_i \sum_k y_k^{(i)} \log y_k^{(i)}$$
$$= -\frac{1}{m}\sum_i \left(y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right)$$

Solve Logistic Regression

$$f_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}$$

$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x})$ $\qquad g(z) = \frac{1}{1+e^{-z}}$

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\sum_i \left(y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)\right)$$

Goal: $\min_\theta J(\theta)$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\frac{1}{m}\sum_i \left(y^{(i)} - \hat{y}^{(i)}\right) \mathbf{x}^{(i)}$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \left(y^{(i)} - f_\theta\left(x^{(i)}\right)\right) x_j^{(i)}$$

Gradient descent

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

(Simultaneously update all θ_j)

$$\boxed{\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}}$$

Gradient descent for Linear Regression

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

}

$$\boxed{f_\theta(x) = \theta^\top x}$$

Gradient descent for Logistic Regression

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

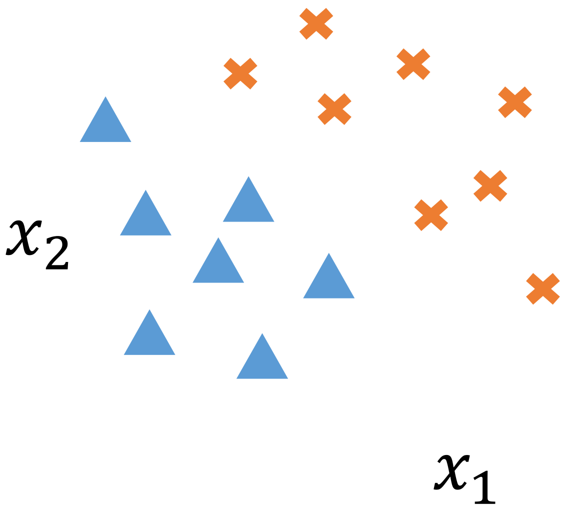
}

$$\boxed{f_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}}$$

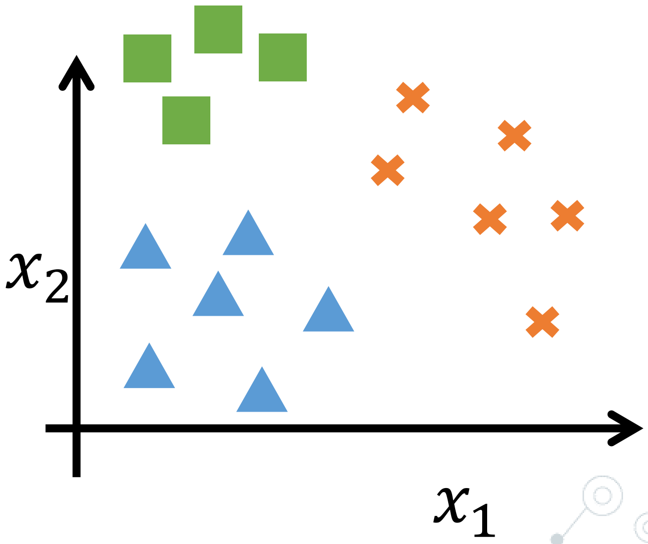
Multi-class Logistic Regression

Multiclass classification

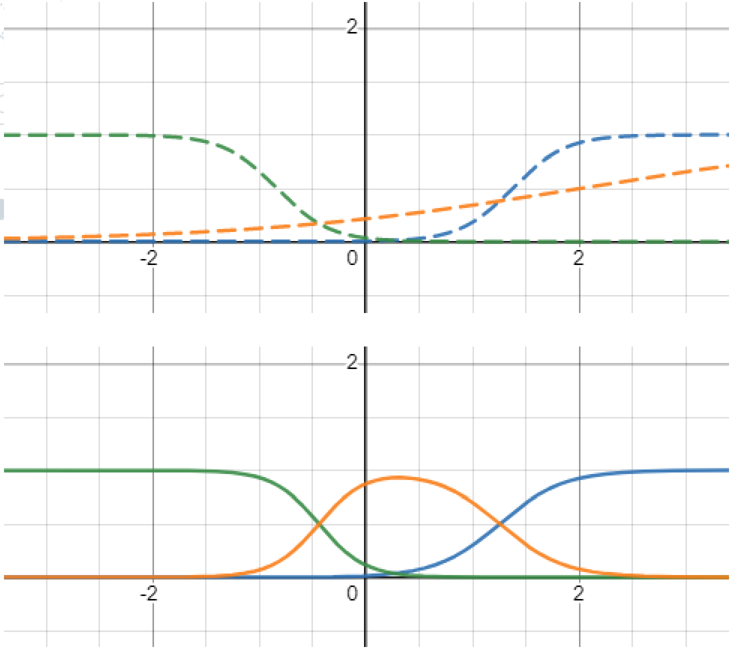
Binary classification



Multiclass classification



Multi-class Logistic Regression



Multi-class Logistic Regression

- Cross-entropy loss

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$

- Model

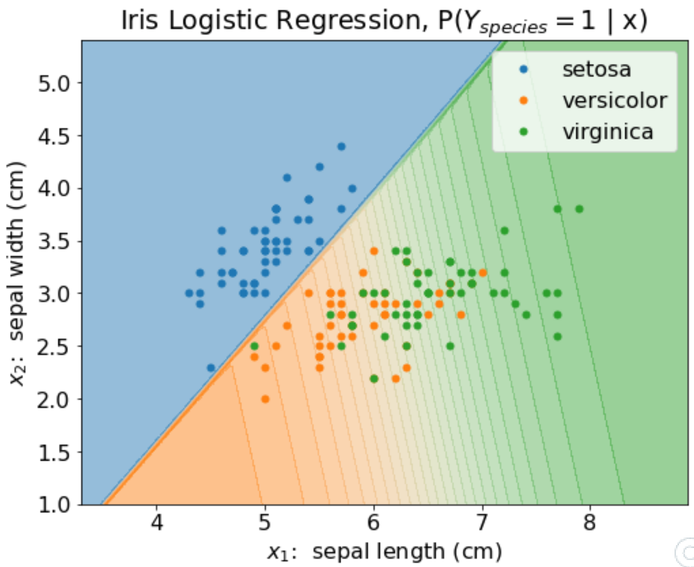
$$\hat{\mathbf{y}} = h(\mathbf{x}) = g_{\text{softmax}}(\mathbf{z})$$

$$\mathbf{z} = \Theta\mathbf{x}$$

$$z_k = \boldsymbol{\theta}_k \mathbf{x}$$

One vector of parameters for each class

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \qquad \boldsymbol{\theta}_k = \begin{bmatrix} b_k \\ w_{k,1} \\ w_{k,2} \end{bmatrix}$$



Stacked into a matrix of $K \times d$ parameters

$$\Theta = \begin{bmatrix} - & \boldsymbol{\theta}_1^\top & - \\ - & \boldsymbol{\theta}_2^\top & - \\ - & \boldsymbol{\theta}_3^\top & - \end{bmatrix} = \begin{bmatrix} b_1 & w_{1,1} & w_{1,2} \\ b_2 & w_{2,1} & w_{2,2} \\ b_3 & w_{3,1} & w_{3,2} \end{bmatrix}$$

Logistic Function

- Logistic (sigmoid) function converts value from $(-\infty, \infty) \rightarrow (0, 1)$

$$g(z) = \frac{1}{1 + \mathrm{e}^{-z}} = \frac{e^{z}}{e^{z} + 1}$$

- $g(z)$ and $1 - g(z)$ sum to one

    - Example 2 → $g(2) = 0.88$, 1-$g(2) = 0.12$

Softmax Function

- Softmax function convert each value in a vector of values from $(-\infty, \infty) \rightarrow (0, 1)$, such that they all sum to one.

$$g(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_K \end{bmatrix} \rightarrow \begin{bmatrix} e^{z_1} \\ e^{z_2} \\ \vdots \\ e^{z_K} \end{bmatrix} \cdot \frac{1}{\sum_{k=1}^{K} e^{z_k}}$$

Example $\begin{bmatrix} -1 \\ 4 \\ 1 \\ -2 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 0.0047 \\ 0.7008 \\ 0.0349 \\ 0.0017 \\ 0.2578 \end{bmatrix}$

Multiclass Predicted Probability

Multiclass logistic regression uses the parameters learned across all K classes to predict the discrete conditional probability distribution of the output Y given a specific input vector $\mathbf{x}$

$$\begin{bmatrix} p(Y = 1 \mid \mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \\ p(Y = 2 \mid \mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \\ p(Y = 3 \mid \mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \end{bmatrix} = \begin{bmatrix} e^{\boldsymbol{\theta}_1^T \mathbf{x}} \\ e^{\boldsymbol{\theta}_2^T \mathbf{x}} \\ e^{\boldsymbol{\theta}_3^T \mathbf{x}} \end{bmatrix} \cdot \frac{1}{\sum_{k=1}^{K} e^{\boldsymbol{\theta}_k^T \mathbf{x}}}$$

Multi-class Classification

- Multi-class Classification: y can take on K different values $\{1,2,\dots,k\}$
- $f_\theta(x)$ estimates the probability of belonging to each class

$P(y=k|x,\theta) \propto \exp(\theta_k^T x)$

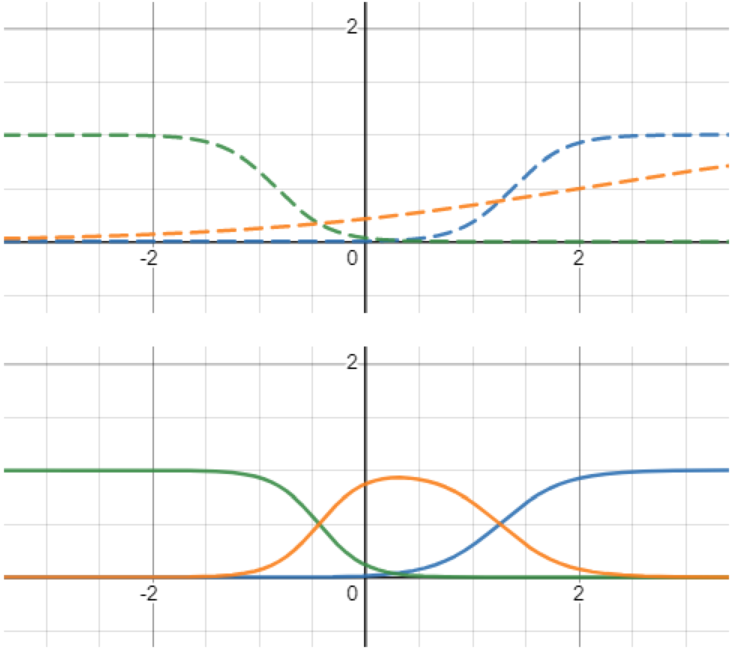
$$\theta = \begin{bmatrix} \vdots & \vdots & \vdots \\ \theta_1 & \theta_2 & \theta_k \\ \vdots & \vdots & \vdots \end{bmatrix}$$

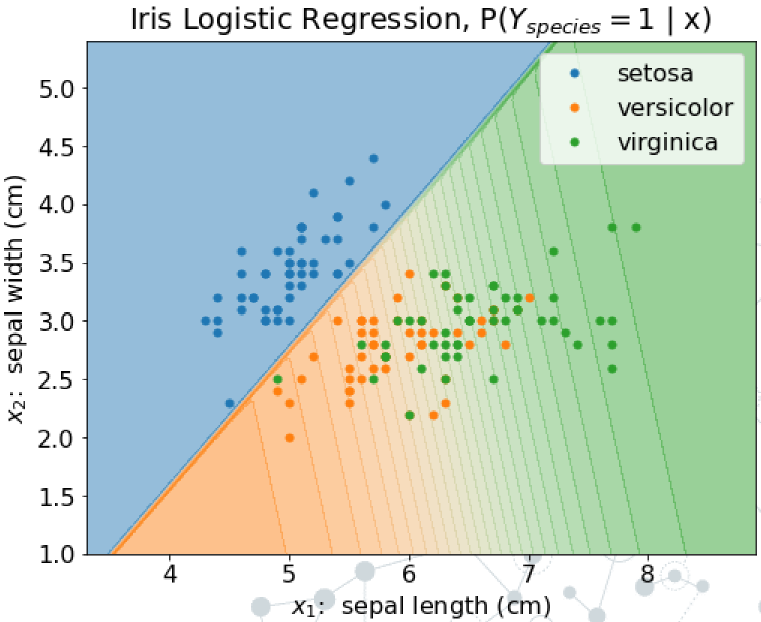
$$P(y=k|x,\theta) = \frac{\exp(\theta_k^T x)}{\sum_{j=1}^{K}\exp(\theta_j^T x)}$$

$$\boxed{\text{loss}(\theta) = -\left[\sum_{i=1}^{m}\sum_{j=1}^{K} 1\{y^{(i)}=k\}\log\frac{\exp(\theta_k^T x^{(i)})}{\sum_{j=1}^{K}\exp\left(\theta_j^T x^{(i)}\right)}\right]}$$

Multiclass Predicted Probability

- Multiclass logistic regression uses the parameters learned across all K classes to predict the discrete conditional probability distribution of the output Y given a specific input vector $\mathbf{x}$





END

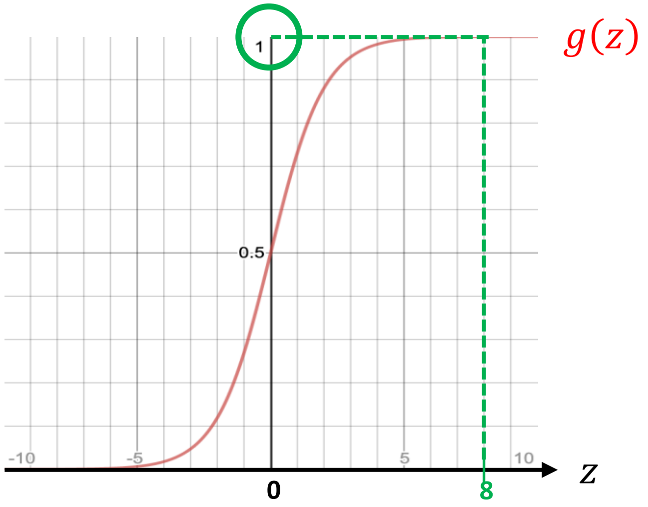
Regression vs. Classification

We want the possible outputs of $\boldsymbol{f_\theta(x)} = \boldsymbol{\theta^T x}$ to be discrete-valued

Use an ***activation function*** (e.g., ***sigmoid or logistic function***)

$$g(z) = \frac{1}{1+e^{-z}}$$

$z \in \mathbb{R}, but$
$g(z) \in [0,1]$



If y = **1**, we want $g(z) \approx 1$ (i.e., we want a correct prediction)
For this to happen, $\boldsymbol{z} \gg \boldsymbol{0}$

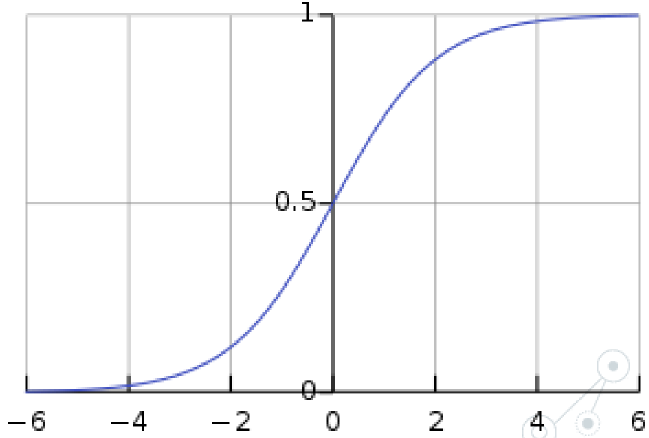
If y = **0**, we want $g(z) \approx 0$ (i.e., we want a correct prediction)
For this to happen, $\boldsymbol{z} \ll \boldsymbol{0}$

Classification

$\boldsymbol{x} = [\boldsymbol{x}_0, \dots, \boldsymbol{x}_m] \rightarrow \boldsymbol{g(\theta^T x)} \rightarrow \boldsymbol{f_\theta(x)} \in [0,1]$

$f_\theta(x) = g(\theta^\top x)$

$g(z) = \dfrac{1}{1 + e^{-z}}$



Thresholding:

predict “y = 1” if $f_\theta(x) \geq 0.5$

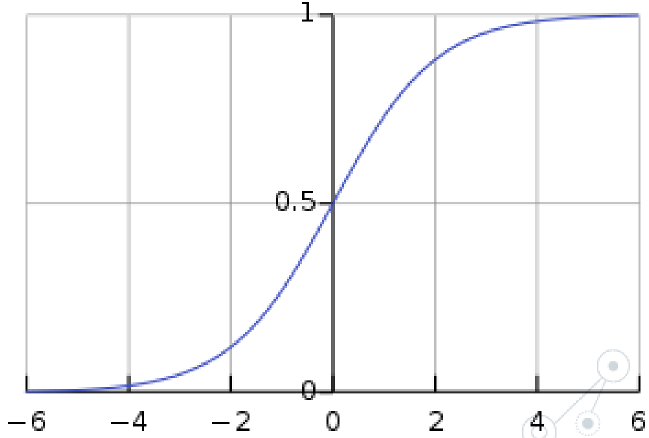
predict “y = 0” if $f_\theta(x) < 0.5$

Classification

$\boldsymbol{x = [x_0, \dots, x_m]} \rightarrow \boldsymbol{g(\theta^T x)} \rightarrow \boldsymbol{f_\theta(x)} \in [0,1]$

$f_\theta(x) = g(\theta^\top x)$

$g(z) = \dfrac{1}{1 + e^{-z}}$



Thresholding:

predict "y = 1" if $f_\theta(x) \geq 0.5$

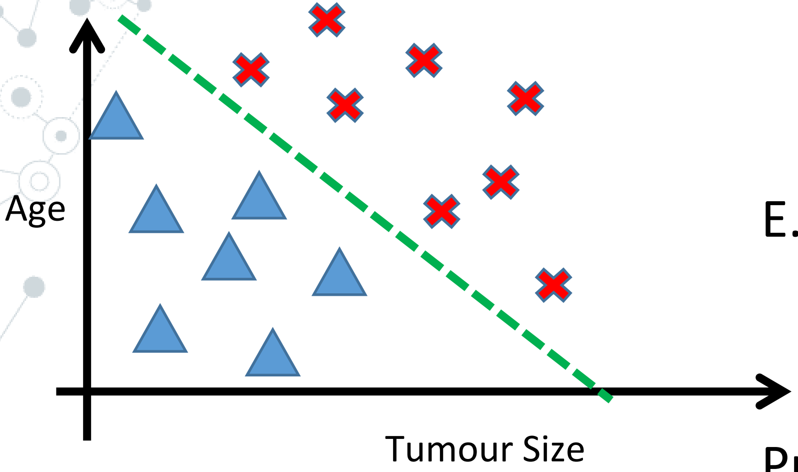
$\boldsymbol{z = \theta^\top x \geq 0}$

predict "y = 0" if $f_\theta(x) < 0.5$

$\boldsymbol{z = \theta^\top x < 0}$

Alternative Interpretation: $f_\theta(x) =$ estimated probability that $y = 1$ on input x

Decision boundary

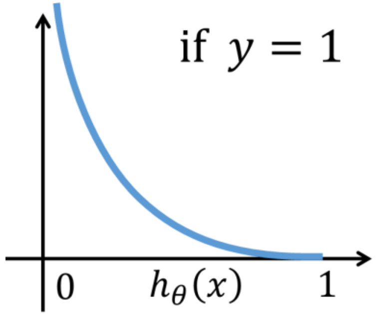


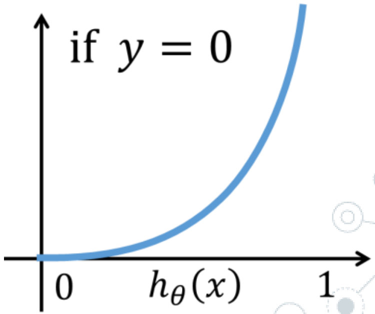
$$f_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

E.g., $\theta_0 = -3, \theta_1 = 1, \ \theta_2 = 1$

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

Cost function for Logistic Regression





Logistic Regression

$$\mathrm{Cost}(f_\theta(x), y) = \begin{cases} -\log\big(f_\theta(x)\big) & \text{if } y = 1 \\ -\log\big(1 - f_\theta(x)\big) & \text{if } y = 0 \end{cases}$$

$$= \boldsymbol{-y\ \log\big(f_\theta(x)\big) - (1 - y)\log\big(1 - f_\theta(x)\big)}$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(\boldsymbol{f}_\theta(x^{(i)}), y^{(i)})$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\ \log\left(\boldsymbol{f}_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)})\log\left(1 - \boldsymbol{f}_\theta\left(x^{(i)}\right)\right)\right]$$

Multi-class Classification

- Multi-class Classification: y can take on K different values $\{1,2,\ldots,k\}$
- $f_\theta(x)$ estimates the probability of belonging to each class

$$P(y=k|x,\theta) \propto \exp\left(\theta_k^T x\right)$$

$$\theta = \begin{bmatrix} \vdots & \vdots & \vdots \\ \theta_1 & \theta_2 & \theta_k \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$P(y=k|x,\theta) = \frac{\exp\left(\theta_k^T x\right)}{\sum_{j=1}^K \exp\left(\theta_j^T x\right)}$$

$$\boxed{\mathrm{loss}(\theta) = -\left[\sum_{i=1}^m \sum_{j=1}^K 1\{y^{(i)}=k\} \log \frac{\exp\left(\theta_k^T x^{(i)}\right)}{\sum_{j=1}^K \exp\left(\theta_j^T x^{(i)}\right)}\right]}$$