Logistic Regression

Machine Learning (AIM 5002-41)

Joon Hee Choi
Sungkyunkwan University

Classification

Email: Spam / Not Spam?

Online Transactions: Fraudulent (Yes / No)?

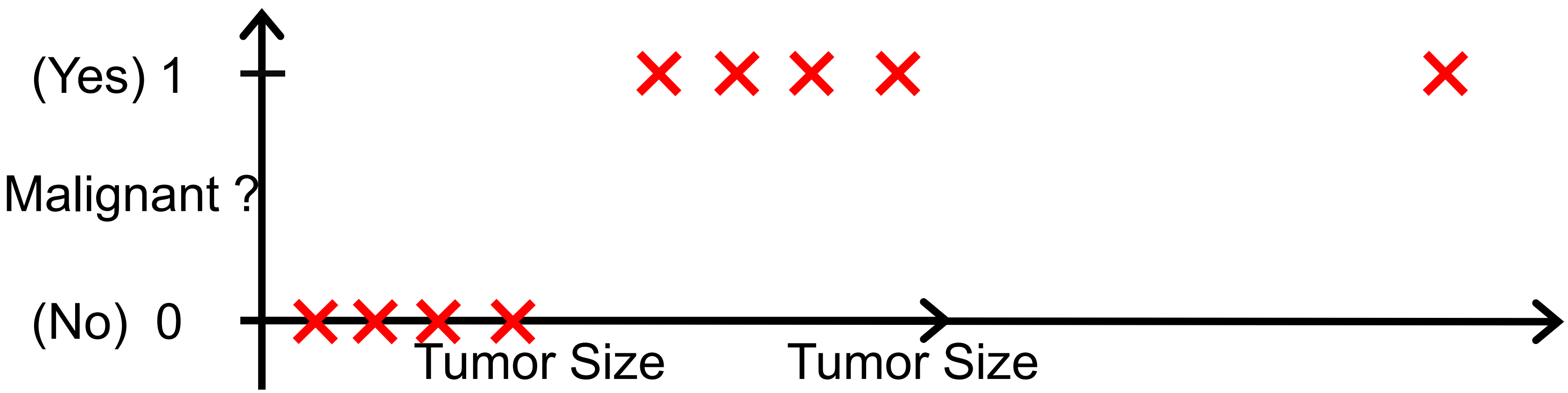
Tumor: Malignant / Benign ?

$y \in \{0, 1\}$

0: “Negative Class” (e.g., benign tumor)

1: “Positive Class” (e.g., malignant tumor)

Classification



Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Classification

Classification: y = 0 or 1

$h_\theta(x)$ can be > 1 or < 0

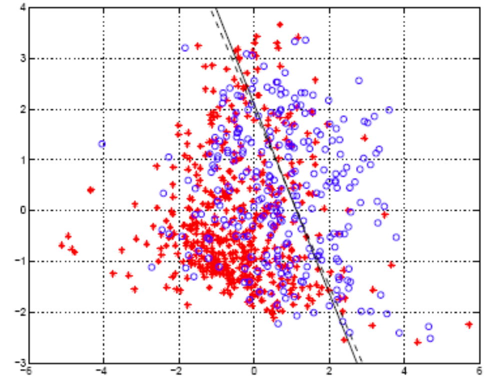
Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Classification Based on Probability

- Instead of just predicting the class, give the probability of the instance being that class
  - i.e., learn $p(y|x)$

- Comparison to perceptron:
  - Perceptron doesn't produce probability estimate
  - Perceptron (and other discriminative classifiers) are only interested in producing a discriminative model

- Recall that:
  - $0 \leq p(event) \leq 1$
  - $p(event) + p(\neg event) = 1$



Logistic Regression

- Takes a probabilistic approach to learning discriminative functions (i.e., a classifier)

- $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ should give $p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta})$
  - Want $0 \leq h_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$

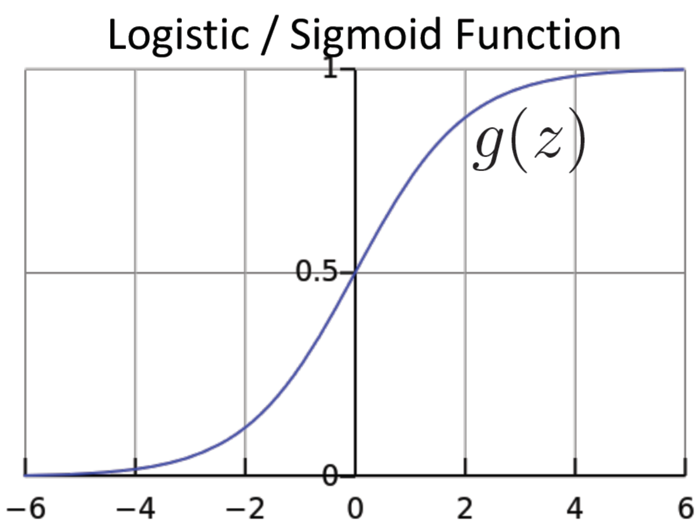
Can't just use linear regression with a threshold

- Logistic regression model:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}\right)$$

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

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



Interpretation of Hypothesis Output

- $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ = estimated $p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta})$

Example: Cancer diagnosis from tumor size

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0.7$$

→ Tell patient that 70% chance of tumor being malignant

- Note that: $p(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) + p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 1$

  Therefore, $p(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 1 - p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta})$

Another Interpretation

- Equivalently, logistic regression assumes that

$$\log \boxed{\frac{p(y=1 \mid \boldsymbol{x}; \boldsymbol{\theta})}{p(y=0 \mid \boldsymbol{x}; \boldsymbol{\theta})}} = \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d$$

odds of $y = 1$

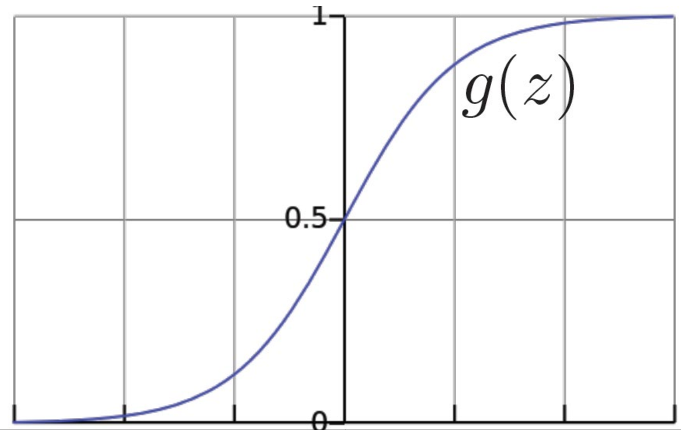
> **Side Note**: the odds in favor of an event is the quantity *p/(1−p)*, where *p* is the probability of the event
>
> E.g., If I toss a fair dice, what are the odds that I will have a 6?

- In other words, logistic regression assumes that the log odds is a linear function of $\boldsymbol{x}$

Logistic Regression

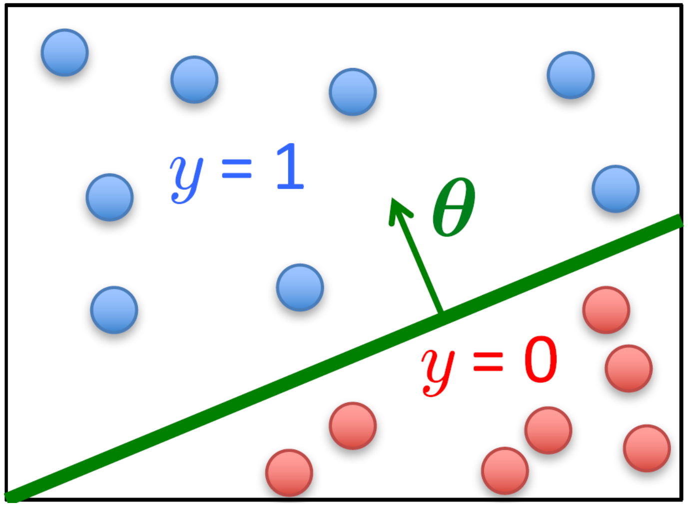
$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}\right)$$

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

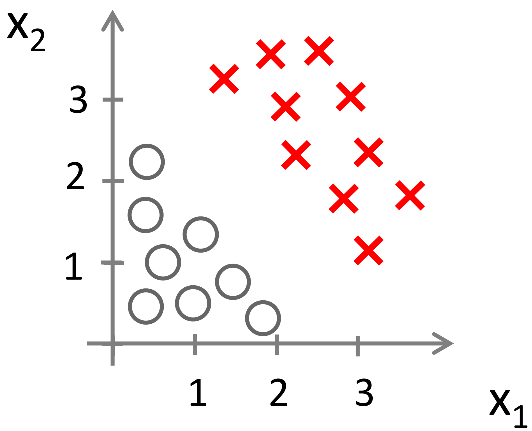


$\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}$ should be large negative values for negative instances	$\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}$ should be large positive values for positive instances

- Assume a threshold and...
  - Predict $y = 1$ if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5$
  - Predict $y = 0$ if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5$



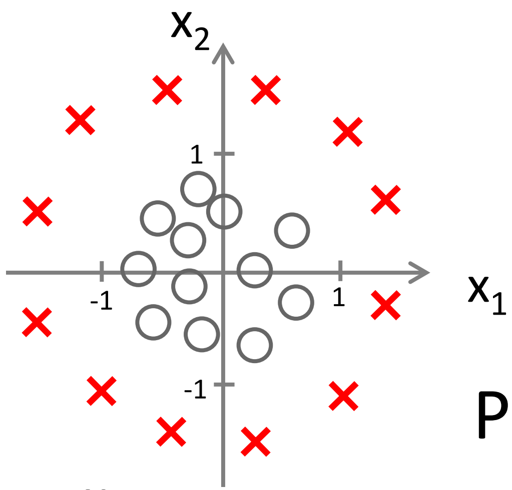
Decision Boundary



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

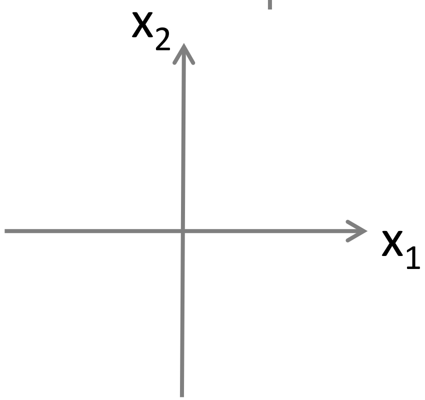
Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

Non-linear Decision Boundary



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \dots)$$

Logistic Regression

- Given $\left\{\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \left(\boldsymbol{x}^{(2)}, y^{(2)}\right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}$
  where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \{0, 1\}$

- Model:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x}\right)$$

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

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \qquad \boldsymbol{x}^{\mathbf{T}} = \begin{bmatrix} 1 & x_1 & \cdots & x_d \end{bmatrix}$$

Logistic Regression Objective Function

- Can't just use squared loss as in linear regression:

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

  - Using the logistic regression model

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

    results in a non-convex optimization

Deriving the Cost Function via Maximum Likelihood Estimation

- Likelihood of data is given by: $l(\boldsymbol{\theta}) = \prod_{i=1}^{n} p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta})$

- So, looking for the $\boldsymbol{\theta}$ that maximizes the likelihood

$$\boldsymbol{\theta}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

- Can take the log without changing the solution:

$$\begin{aligned}\boldsymbol{\theta}_{\text{MLE}} &= \arg\max_{\boldsymbol{\theta}} \log \prod_{i=1}^{n} p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta})\end{aligned}$$

Deriving the Cost Function via Maximum Likelihood Estimation

- Expand as follows:

$$\boldsymbol{\theta}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left[y^{(i)} \log p(y^{(i)} = 1 \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) + \left(1 - y^{(i)}\right) \log \left(1 - p(y^{(i)} = 1 \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta})\right) \right]$$

- Substitute in model, and take negative to yield

Logistic regression objective:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

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

Intuition Behind the Objective

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

- Cost of a single instance:

$$\mathrm{cost}(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} -\log\left(h_{\boldsymbol{\theta}}(\boldsymbol{x})\right) & \text{if } y = 1 \\ -\log\left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\right) & \text{if } y = 0 \end{cases}$$

- Can re-write objective function as

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \mathrm{cost}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right), y^{(i)}\right)$$

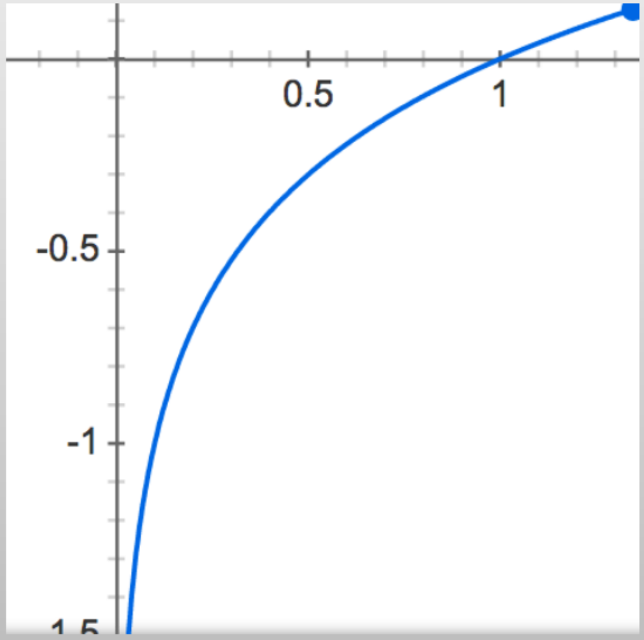
Compare to linear regression:

$$J(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

Intuition Behind the Objective

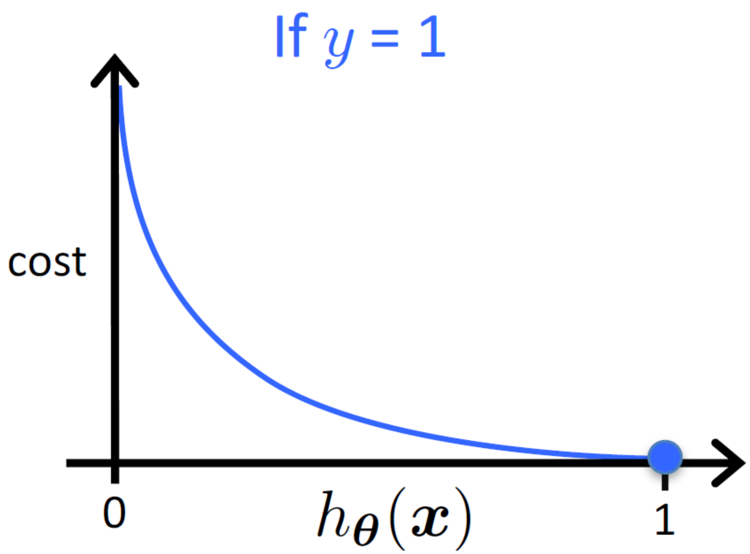
$$\text{cost}(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} -\log\big(h_{\boldsymbol{\theta}}(\boldsymbol{x})\big) & \text{if } y = 1 \\ -\log\big(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\big) & \text{if } y = 0 \end{cases}$$

Aside: Recall the plot of log(z)



Intuition Behind the Objective

$$\text{cost}(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} \boxed{-\log\big(h_{\boldsymbol{\theta}}(\boldsymbol{x})\big) \quad \text{if } y = 1} \\ -\log\big(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\big) \quad \text{if } y = 0 \end{cases}$$

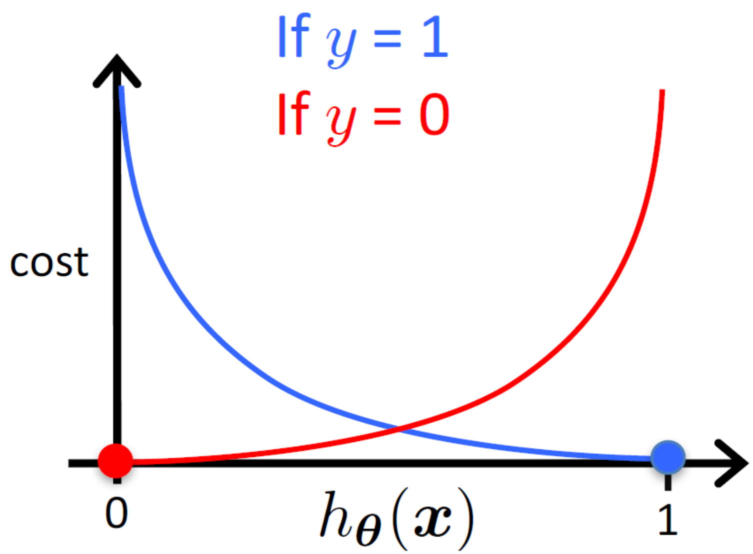


If $y = 1$

- Cost = 0 if prediction is correct
- As $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 0, \text{cost} \to \infty$
- Captures intuition that larger mistakes should get larger penalties
  - e.g., predict $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$, but $y = 1$

Intuition Behind the Objective

$$\mathrm{cost}(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} \boxed{-\log\left(h_{\boldsymbol{\theta}}(\boldsymbol{x})\right) \quad \text{if } y = 1} \\ \boxed{-\log\left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\right) \quad \text{if } y = 0} \end{cases}$$



If $y = 0$

- Cost = 0 if prediction is correct
- As $\left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\right) \to 0, \mathrm{cost} \to \infty$
- Captures intuition that larger mistakes should get larger penalties

Regularized Logistic Regression

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

- We can regularize logistic regression exactly as before:

$$\begin{aligned} J_{\text{regularized}}(\boldsymbol{\theta}) &= J(\boldsymbol{\theta}) + \frac{\lambda}{2} \sum_{i=1}^{n} \theta_j^2 \\ &= J(\boldsymbol{\theta}) + \frac{\lambda}{2} \left\| \boldsymbol{\theta}_{[1:d]} \right\|_2^2 \end{aligned}$$

Gradient Descent for Logistic Regression

$$J_{\text{reg}}(\boldsymbol{\theta}) = -\sum_{i=1}^{n}\left[y^{(i)}\log\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right)\right) + \left(1-y^{(i)}\right)\log\left(1-h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right)\right)\right] + \frac{\lambda}{2}\left\|\boldsymbol{\theta}_{[1:d]}\right\|_2^2$$

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for $j = 0 \ldots d$)

$$\theta_0 \leftarrow \theta_0 - \alpha\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)$$

$$\theta_j \leftarrow \theta_j - \alpha\left[\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \lambda\theta_j\right]$$

Gradient Descent for Logistic Regression

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for $j = 0 \ldots d$)

$$\theta_0 \leftarrow \theta_0 - \alpha \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)} \right)$$

$$\theta_j \leftarrow \theta_j - \alpha \left[\sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)} \right) x_j^{(i)} + \lambda \theta_j \right]$$

This looks IDENTICAL to linear regression!!!

- Ignoring the $1/n$ constant
- However, the form of the model is very different:

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

Stochastic Gradient Descent

Consider Learning with Numerous Data

- Logistic regression objective:

$$J(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\underbrace{\left[y_i \log\left(h_{\boldsymbol{\theta}}(\mathbf{x}_{\boldsymbol{i}})\right) + (1 - y_i)\log\left(1 - h_{\boldsymbol{\theta}}(\mathbf{x}_{\boldsymbol{i}})\right)\right]}_{\mathrm{cost}_{\theta}(\mathbf{x}_i, y_i)}$$

- Fit via gradient descent:

$$\theta_j \leftarrow \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i\right) x_{ij}$$

- What is the computational complexity in terms of n?

Gradient Descent

Batch Gradient Descent

Initialize θ

Repeat {

$$\underbrace{\theta_j \leftarrow \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i \right) x_{ij}}_{\frac{\partial}{\partial \theta_j} J(\theta)} \qquad \text{for } j = 0, \ldots, d$$

}

Stochastic Gradient Descent

Initialize θ

Randomly shuffle dataset

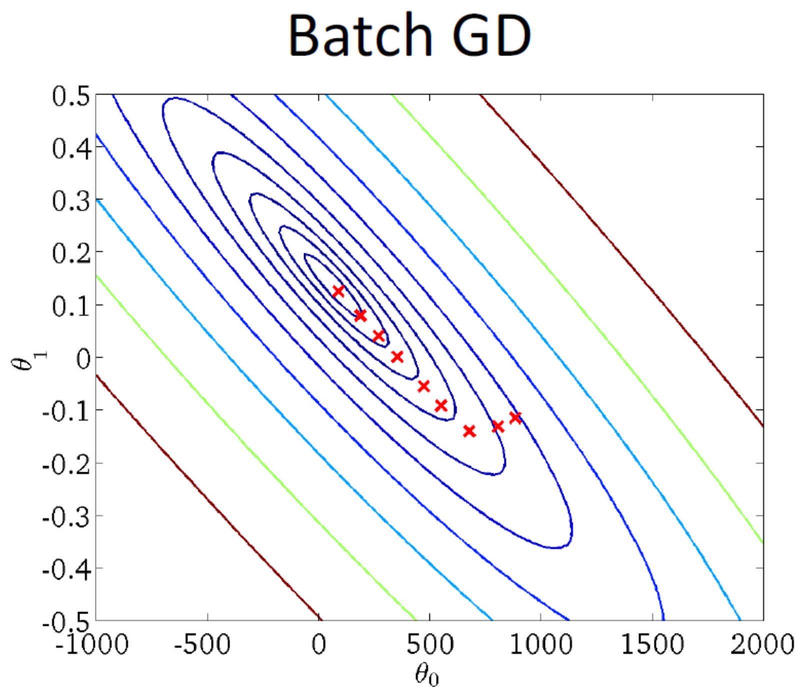
Repeat { (Typically 1 - 10x)

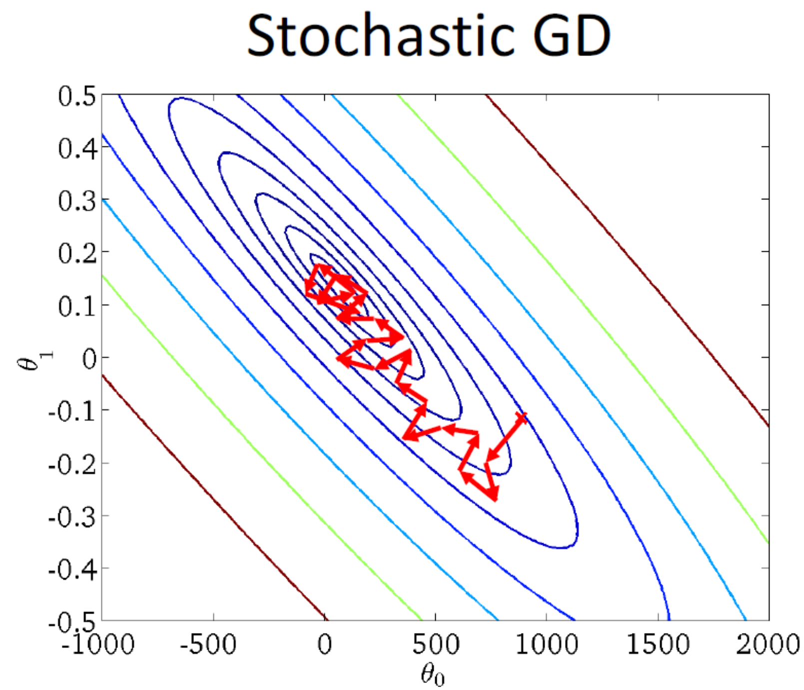
For $i = 1, \ldots, n$ do

$$\underbrace{\theta_j \leftarrow \theta_j - \alpha \left(h_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i \right) x_{ij}}_{\frac{\partial}{\partial \theta_j} \mathrm{cost}_{\theta}(\mathbf{x}_i, y_i)} \qquad \text{for } j = 0, \ldots, d$$

}

Batch GD vs Stochastic GD





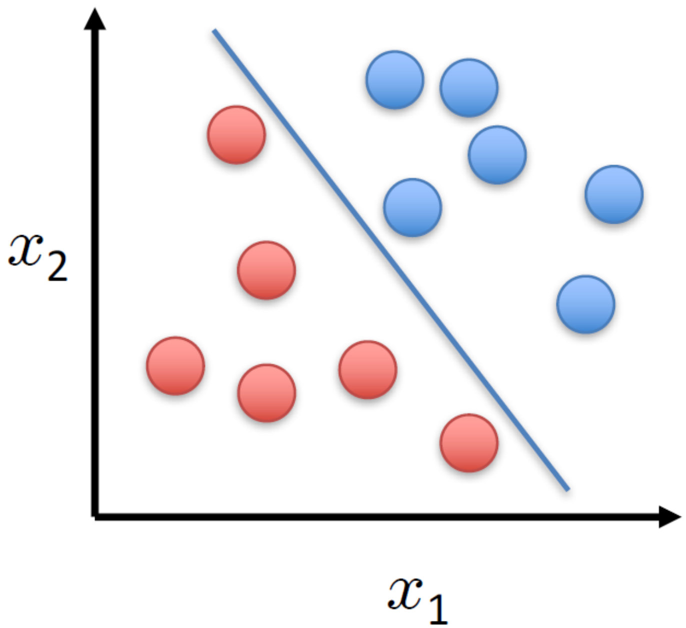
- Learning rate α is typically held constant
- Can slowly decrease α over time to force θ to converge:

$$\text{e.g., } \alpha_t = \frac{\text{constant1}}{\text{iterationNumber} + \text{constant2}}$$

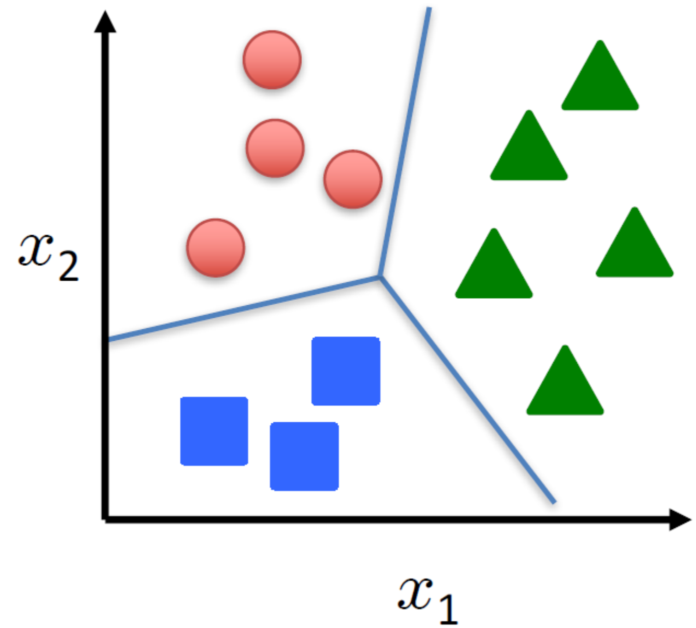
Multi-Class Classification

Multi-Class Classification

Binary classification:



Multi-class classification:



Disease diagnosis:	healthy / cold / flu / pneumonia
Object classification:	desk / chair / monitor / bookcase

Multi-Class Logistic Regression

- For 2 classes:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x})} = \frac{\exp\left(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}\right)}{\boxed{1} + \boxed{\exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x})}}$$

weight assigned to $y = 0$

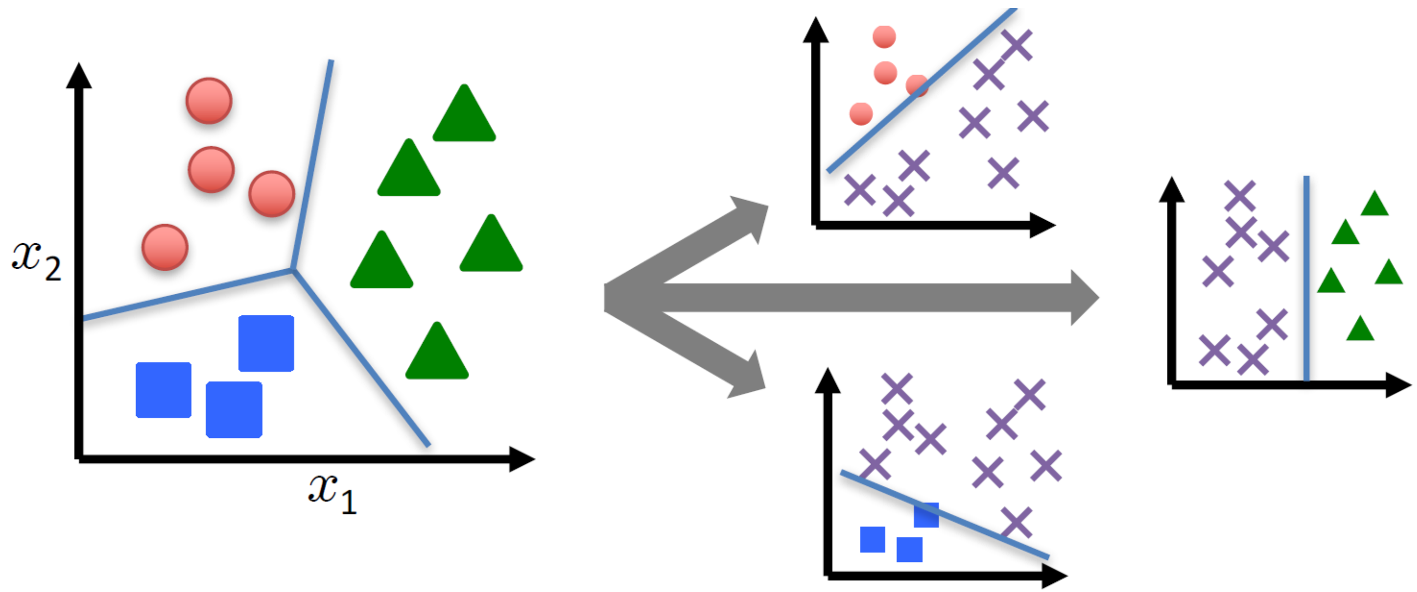
weight assigned to $y = 1$

- For C classes $\{1, \ldots, C\}$:

$$p(y = c \mid \boldsymbol{x}; \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_c) = \frac{\exp\left(\boldsymbol{\theta}_{\mathrm{c}}^{\mathrm{T}}\,\boldsymbol{x}\right)}{\sum_{c=1}^{C} \exp\left(\boldsymbol{\theta}_{\mathrm{c}}^{\mathrm{T}}\,\boldsymbol{x}\right)}$$

  - Called the **softmax** function

Multi-Class Logistic Regression



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

$$h_c(\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta}_c^T \boldsymbol{x})}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^T \boldsymbol{x})}$$

Implementing Multi-Class Logistic Regression

- Use $h_c(\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta}_\mathrm{c}^\mathrm{T}\,\boldsymbol{x})}{\sum_{c=1}^{C}\exp(\boldsymbol{\theta}_\mathrm{c}^\mathrm{T}\,\boldsymbol{x})}$ as the model for class c

- Gradient descent simultaneously updates all parameters for all models
  - Same derivative as before, just with the above $h_c(\boldsymbol{x})$

- Predict class label as the most probable label

$$\max_{\boldsymbol{c}} h_c(\boldsymbol{x})$$

Reference

- https://www.seas.upenn.edu/~cis519
- Andrew NG's slides