The National University of Computer and Emerging Sciences

Logistic Regression

Machine Learning for Data Science

Dr. Akhtar Jamil

Department of Computer Science

Goals

- Review of Previous Lecture
- Today's Lecture
  - Logistic Regression

Review Of Previous Lecture

Linear regression with multiple features

- The idea of linear regression can be extended for multiple variables.
  - A set of multi features (X) will be input to the model
  - Y is continuous valued response (target variable y).

- The idea of linear regression can be extended for multiple variables.
  - A set of multi features (X) will be input to the model
  - Y is continuous valued response (target variable y).

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 \ldots + w_n x_n$$

$$\hat{y} = h_w(x) = W.x$$

Where x $= x_1, x_2, \ldots x_n$ and W = $w_0, w_1, \ldots w_n$

Where x $= x_1, x_2, \ldots x_n$ and W = $w_0, w_1, \ldots w_n$

Linear regression with multiple features

$$\hat{y}_1 = w_0 + w_1 x_1$$
$$\hat{y}_2 = w_0 + w_1 x_2$$
$$\hat{y}_3 = w_0 + w_1 x_3$$

$$\hat{y}_n = w_0 + w_1 x_n$$

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ . \\ . \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_1 \\ w_0 + w_1 x_2 \\ . \\ . \\ w_0 + w_1 x_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ & . \\ & . \\ 1 & x_n \end{bmatrix} \times \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Optimization: Gradient Descent

- Variants of Gradient Descent are available for optimization in ML
  - Batch Gradient Descent
  - Stochastic Gradient Descent(SGD)
  - Mini-batch Gradient Descent
- **Batch Gradient Descent**
  - Updates the weight vector over the full training data
  - It is very slow on very large training data.

$$w_1 = w_1 - \lambda \frac{\partial J}{\partial w_1}$$

$$w_1 = w_1 - \lambda \frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) x_i$$

- Variants of Gradient Descent are available for optimization in ML
  - Batch Gradient Descent
  - Stochastic Gradient Descent(SGD)
  - Mini-batch Gradient Descent
- **Batch Gradient Descent**
  - Updates the weight vector over the full training data
  - It is very slow on very large training data.

Optimization: Gradient Descent

- **Stochastic Gradient Descent(SGD)**
  - It updates the parameters for each training data, according to its own gradients:

Stochastic Gradient Descent(SGD)

– It updates the parameters for each training data, according to its own gradients:

$$w_1 = w_1 - \lambda \frac{\partial J}{\partial w_1}$$

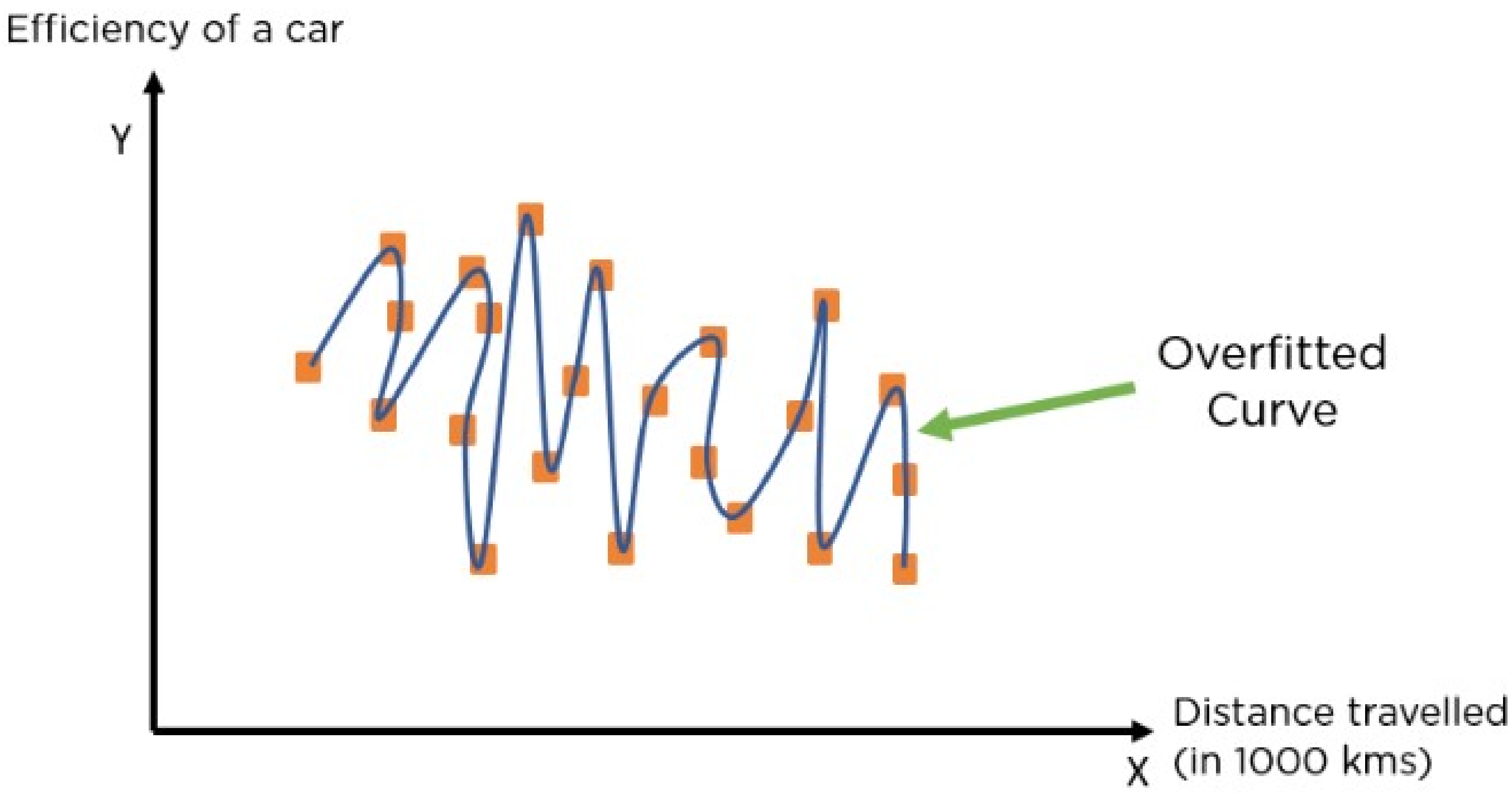
$$w_1 = w_1 - \lambda \, (y_i - \hat{y}_i) x_i$$

Optimization: Gradient Descent

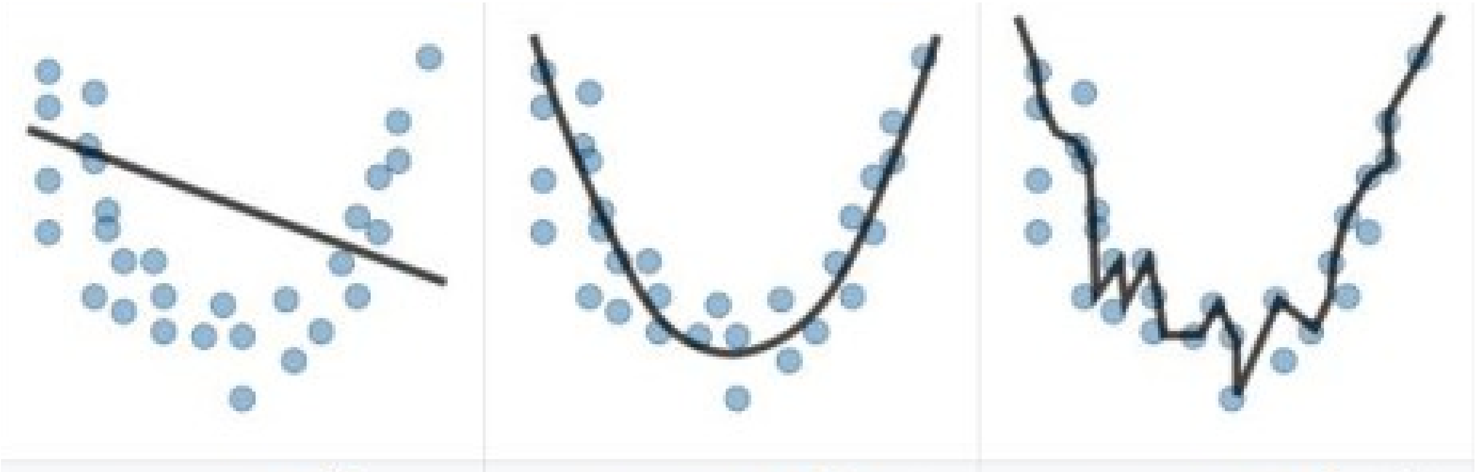
- **Mini-batch Gradient Descent**
  - It computes the gradients on small random sets of instances called mini-batches.
  - It has shown better performance than SGD
  - More robust stable than SGD

Generalization

Overfitting



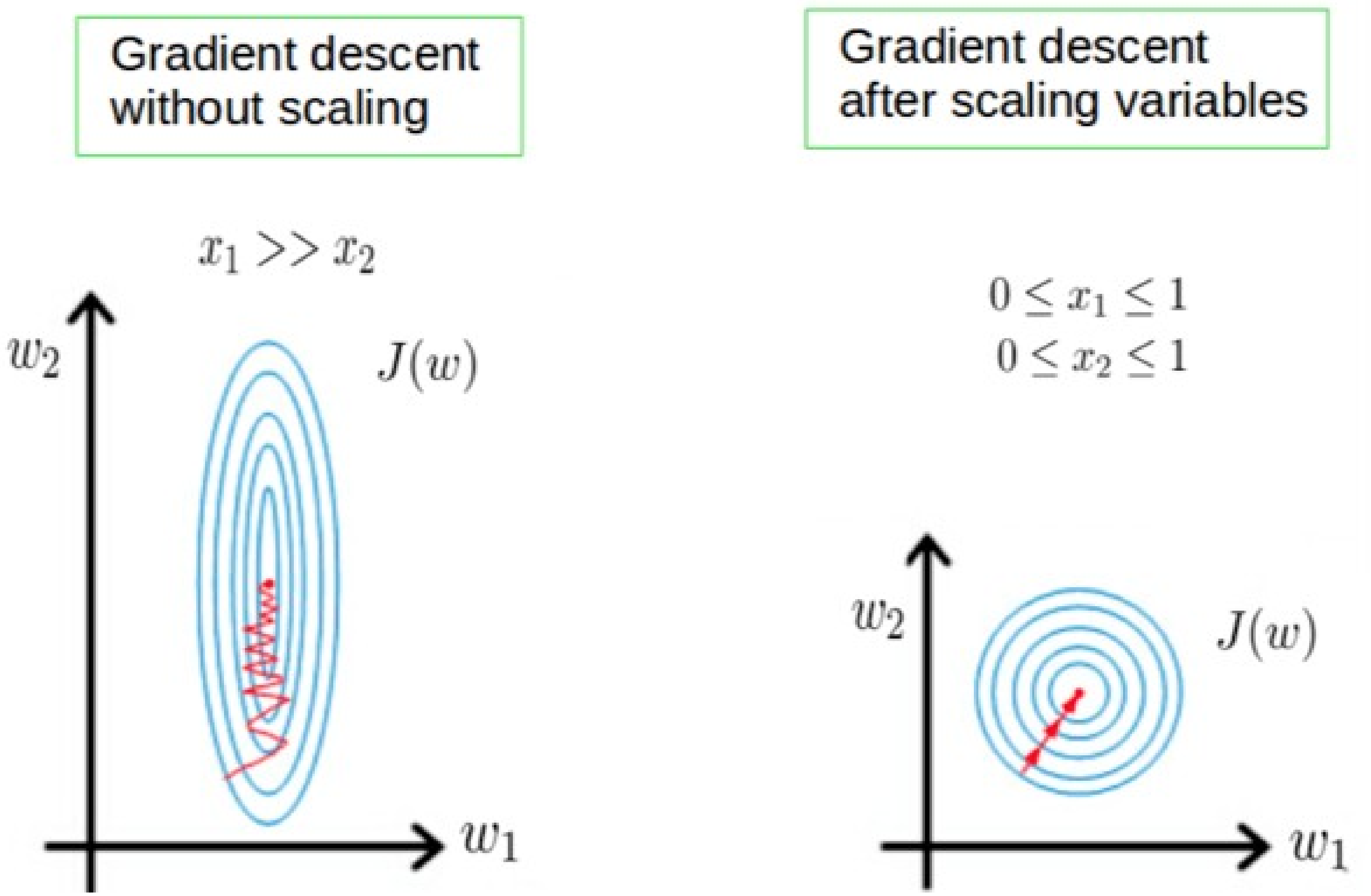
Underfitting vs Overfitting



Feature Scaling

- Feature scaling in machine learning is an important pre-processing steps
  - Affect performance of the model
- The difference in range of values of features may cause one feature to dominate other.
- The most commonly used techniques:
  - Normalization
  - Standardization.

Feature Scaling



Today's Lecture

Odds in Probability

- Example:
  - A survey of 250 customers was conducted for an automobile dealership. The customers were asked if they would recommend the service department to a friend. The number who responded Yes was 210.
  - The proportion (probability) of customers who recommend is

$$\hat{p} = \frac{210}{250} = 0.84$$

  - So, the proportion of customers who would not recommend the service department are:

$$1 - \hat{p} = 1 - 0.84 = 0.16$$

Odds in Probability

- The odds are simply the ratio of the proportions for the two possible outcomes.
- If p is the proportion for one outcome, then 1 - p is the proportion for the second outcome:

$$\text{odds} = \frac{\hat{p}}{1 - \hat{p}}$$

$$\text{odds} = \frac{\hat{p}}{1 - \hat{p}} = \frac{0.84}{0.16} = 5.25$$

Odds in Probability

- Odd usually use integers or fractions.
- 5.25 to 5.
- The odds are approximately 5 to 1 that a customer would recommend the service to a friend.
- In a similar way, we could describe the odds that a customer would not recommend the service as 1 to 5.

Logistic Regression

- Linear regression models the relationship between a response variable and one or more explanatory variables.
- For categorical response variable with two possible values, Similar Regression Models can also be used
  - Spam or Not Spam
  - Patient Dies or Survives
  - Tumor Benign or Malignant.

Logistic Regression

- Classification, like regression, is a predictive task
  - But one in which the outcome takes only values across discrete categories;
- Classification problems are very common (more common than regression problems!)
- The objective function should be modified
- Fundamentals will be same as regression

Logistic Regression

- **Logistic Regression** (also called Logit Regression) is commonly used to estimate the probability for each class
  - What is the probability that this email is spam?
- Can a binary classifier be constructed using probability?
  - If the estimated probability > 50%, the instance belongs positive class
  - Otherwise, it belongs to the negative class

Logistic Regression

- It is a supervised method for classification
- "Logit" = "Log Odds"
- p(y=0|x) or p(y=1|x)?

$$\log\left(\frac{p(x)}{1-p(x)}\right)$$

Logistic Regression

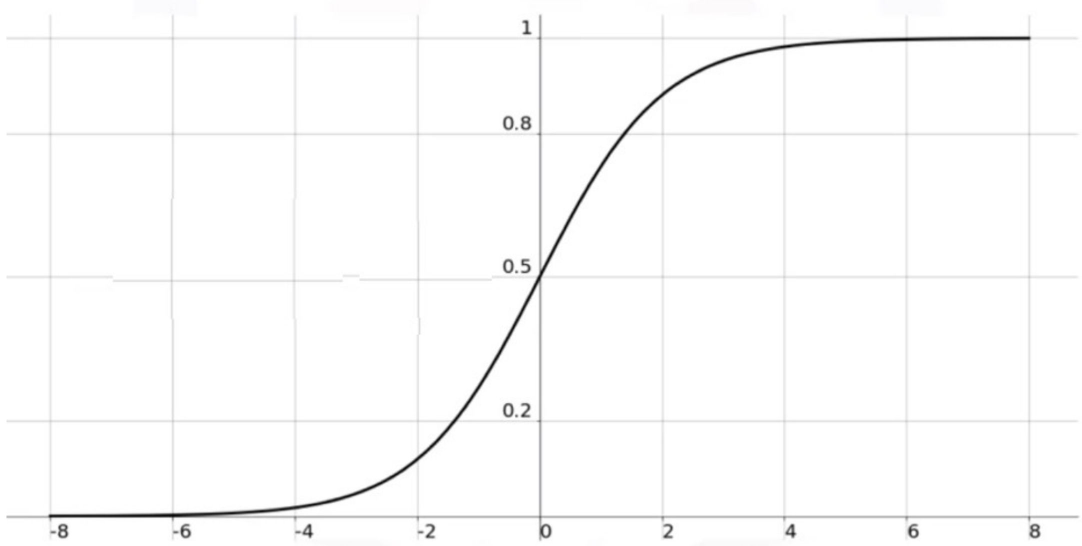
- Suppose p(y=1|x) = p(x)

- Suppose p(y=1|x) = p(x)

- Sigmoid Function:

- Sigmoid Function:

$$p(x) = \frac{1}{1+e^{-W^T X}}$$



Logistic Regression

- Linear Regression model, a Logistic Regression model computes a weighted sum of the input features (plus a bias term),

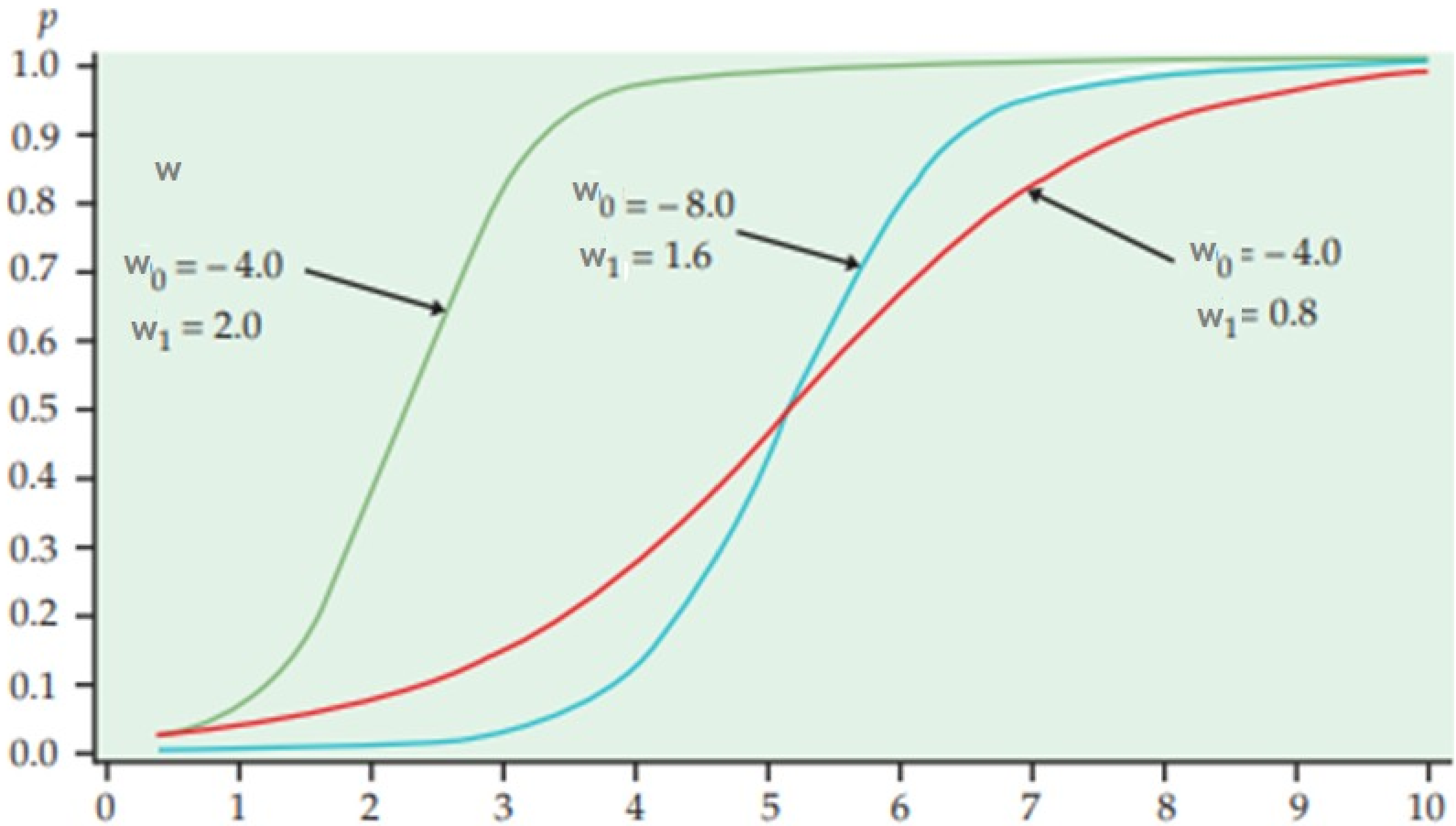
$$y = w_0 + w_1 x = W^T x$$

- Logistic Regression uses Sigmoid Function to model the relationship between input variable and output response.

$$p(x) = W^T x$$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \underline{W}^T x = w_0 + w_1 x$$

Logistic Regression



Logistic Regression

$$p(x) = h_{\boldsymbol{w}}(\mathbf{x}) = \sigma(\mathbf{x}^{\top}\boldsymbol{W})$$

- σ(·) is a sigmoid function
- Outputs a number between 0 and 1
- Logistic Regression model Prediction

- σ(·) is a sigmoid function
- Outputs a number between 0 and 1
- Logistic Regression model Prediction

$$\hat{y} = \begin{cases} 0 & \text{if } \mathrm{p(x)} < 0.5 \\ 1 & \text{if } \mathrm{p(x)} \geq 0.5 \end{cases}$$

Logistic Regression

- Goal: w_0 and w_1 must be estimated.
- Linear Regression uses Least Squared method
- Logistic Regression uses Maximum Likelihood estimation (MLE)
  - For a Binary classification:
  - M labeled samples with labels (0 or 1)
  - For class-1: Find values of W such that p(x) is close to 1
  - For class-0: Find values of W such that p(x) is close to 0 or 1-p(x) is close to 1

Logistic Regression

- Given samples $(x_i, y_i) \in \mathbb{R}^p \times \{0,1\}, i = 1, \dots m$

- Given samples

- Assume: $p(x_i) = p(y_i = 1 \mid x_i)$

- Assume:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = W^T x$$

- The optimal coefficients of W can be estimated using principle of maximum likelihood.

- The optimal coefficients of W can be estimated using principle of maximum likelihood.

Logistic Regression

$$\mathrm{p}(x) = h_{\boldsymbol{w}}(\mathbf{x}) = \sigma(\mathbf{x}^{\top}\boldsymbol{W})$$

$$\mathrm{p}(x) = \sigma(w_0 + w_1 x_1 + \cdots + w_n x_n)$$

- *Cost function:*

- *Cost function:*

$$Loss(\boldsymbol{W}, \boldsymbol{x}) = \begin{cases} -\log(\mathrm{p}(\mathrm{x})) & \text{if } y = 1 \\ -\log(1 - \mathrm{p}(\mathrm{x})) & \text{if } y = 0 \end{cases}$$

- The *log loss* can be written as:

- The *log loss* can be written as:

$$J(W) = -\frac{1}{m}\sum_{i=1}^{m} [y_i \log(p(x_i)) + (1 - y_i)\log(1 - p(x_i))]$$

Logistic Regression

- This cost function is convex
  - Gradient Descent is guaranteed to find the global minimum
- The weights can be updated using the partial derivative of the cost function according to W_i

$$w_i = w_i - \lambda \frac{\partial J}{\partial w_i}$$

$$\frac{\partial}{\partial w_j} \mathrm{J}(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \left(\sigma(\boldsymbol{W}^{\top} x_i) - y_i\right) x_j$$

Logistic Regression

- Logistic regression will find W such that it minimizes J(W)
- Make prediction using

- Logistic regression will find W such that it minimizes J(W)
- Make prediction using

$$p(x) = \frac{1}{1+e^{-W^T x}}$$

$$\hat{y} = \begin{cases} 0 & \text{if } p(x) < 0.5 \\ 1 & \text{if } p(x) \geq 0.5 \end{cases}$$

Multinomial regression

- Can we extend the Logistic Regression model for multiclass classification(more than two) problems?
  - Yes
- Multinomial model regression
  - We can use One-vs-Rest (One-vs-all)
  - Divide the problem into many sub problems (binary)
  - Train separate model for one class vs rest classes
  - Repeat for all possible combinations for every class

Multinomial regression

- Given K classes (k>2), the predictor Y can be obtained for each model:

$$p_1(y = 1|x) = \frac{1}{1+e^{-W_1^T X}}$$

$$p_2(y = 2|x) = \frac{1}{1+e^{-W2^T X}}$$

$$\vdots$$

$$p_k(y = k|x) = \frac{1}{1+e^{-Wk^T X}}$$

Multinomial regression

- For a new input data X will be passed through each model and get probabilities.

$$\hat{f}(x) = \underset{j=1,\ldots K}{\operatorname{argmax}} \; \hat{p}_j(x)$$

- The instance will be assigned to the class with maximum probability

Reference

- Chapter 5, Deep Learning MIT Press 2016, Ian Goodfellow
- Chapter 3 Pattern Recognition and Machine Learning, Christopher M. Bishop
- Some graphics from the internet:
  - https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35

Thank You