Logistic Regression

Data Mining

Prof. Sujee Lee

Department of Systems Management Engineering

Sungkyunkwan University

Logistic Regression

Logistic Regression

- **(Recap.) Linear Regression**
  - In linear regression, y is predicted by

$$\hat{y} = f(\boldsymbol{x}) = \mathbf{w}^T\boldsymbol{x} + b = w_1\, x_1 + \cdots + w_d x_d + b$$

  - here, $y, \hat{y} \in \mathbb{R}$

- **Logistic Regression**
  - Logistic Regression extends the ideas of linear regression for classification problem
    - i.e., the labels are binary $y = 0 \text{ or } 1$
  - we will use linear model $\mathbf{w}^T\boldsymbol{x} + b$, but $f(\boldsymbol{x})$ to be a probability

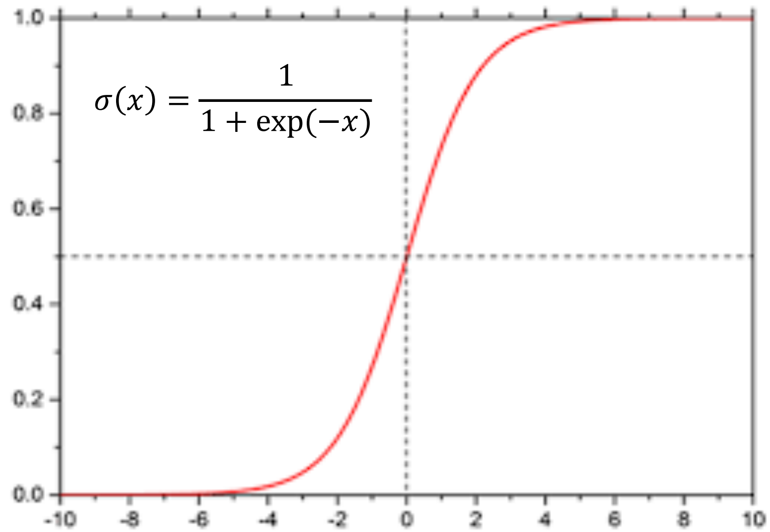
$$\hat{y} = f(\boldsymbol{x}) = \sigma(\mathbf{w}^T\boldsymbol{x} + b) = \frac{1}{1 + \exp(-\boldsymbol{w}^T\boldsymbol{x} - b)}$$

$$\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d, \qquad y \in \mathbb{B}, \qquad 0 \le \hat{y} \le 1$$

Logistic Regression

$$\hat{y} = f(\boldsymbol{x}) = \sigma(\mathbf{w}^T\boldsymbol{x} + b) = \frac{1}{1 + \exp(-\boldsymbol{w}^T\boldsymbol{x} - b)}$$

- here $\sigma(\cdot)$ is called as a **sigmoid function** or a **logistic function**, and results in between 0 and 1

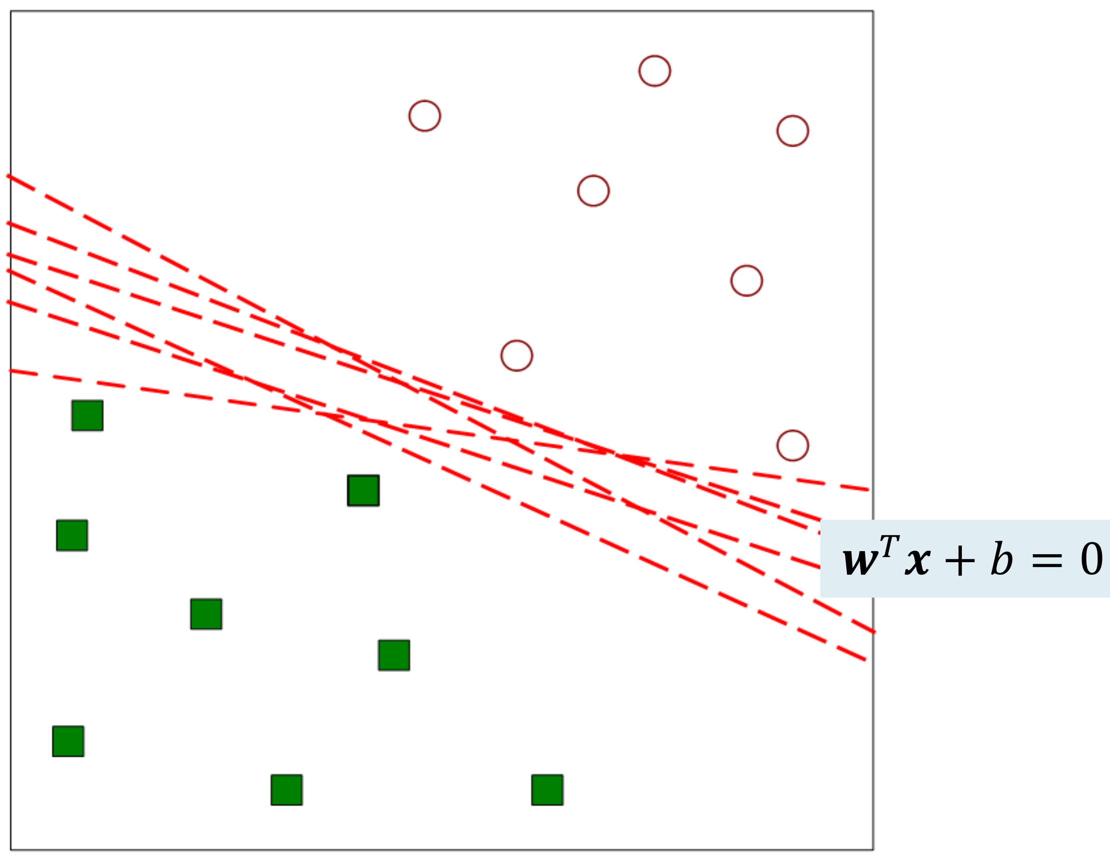


- then, we can consider the class probabilities as

$$P(y = 1) = \frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}-b)}, \; P(y = 0) = \frac{\exp(-\boldsymbol{w}^T\boldsymbol{x}-b)}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}-b)}$$

Logistic Regression

- **This algorithm is a linear classifier**
  - For linear models for binary classification, the **decision boundary** (hyperplane) that separates two classes is a **linear function** of input features.



Learning a Logistic Regression Model

- **Logistic Regression**
  - Goal : Given data pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, find the optimal parameter $\boldsymbol{w}^*$that minimizes the training error. *(as we did in linear regression)*
  - here, we use *"binary cross-entropy"* loss to define classification error

$$\min_{w} \sum_{i=1}^{n} \{-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)\}$$

$-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$: If $y_i = 1,\ -\log \hat{y}_i$ If $y_i = 0, -\log(1 - \hat{y}_i)$

$\hat{y}_i = 1$

$\hat{y}_i = 0$

  - There is no "closed-form". How can we solve the above problem? => *"gradient descent"*

Gradient Descent

Optimization Problem

- **Optimization Problem**
  - (Recap) Linear Regression
    - Ordinary Least Squares : $\min_{w} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||^2$
  - (Further)
    - Logistic Regression : $\min_{w} \sum_{i=1}^{n} \{-y_i \log \widehat{y_i} - (1 - y_i) \log(1 - \widehat{y_i})\}$
    - Neural Network : ...
  - All are about ***minimizing loss function*** (cost function or error function)

$$\min_{w} L(w; \boldsymbol{X}, \boldsymbol{y})$$

Optimization Problem

- **Function Minimization / Maximization Problem**

$$\min_{x} f(x)$$

  - When can we find an (global) **optimal solution?** => when f is convex

  - Note: How to check the convexity : $f''(x)$ or $\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x})$ (Hessian matrix)
    - f is convex if and only if $f''(x) \geq 0$ for all x
    - f is convex if and only if $\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}) \succeq 0$ (positive semi-definite) for all $\boldsymbol{x}$

Optimization Problem

- When f is differentiable and convex, a necessary and sufficient condition for a point x^* to be optimal is $\nabla f(x^*) = 0$.
- If you can find an *analytical solution* (closed-form solution) for $\nabla f(x^*) = 0$, please do!
- If you can't, $\nabla f(x^*) = 0$ usually can be solved by an *iterative algorithm*
  - it means, you will update your solutions as the following until convergence

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

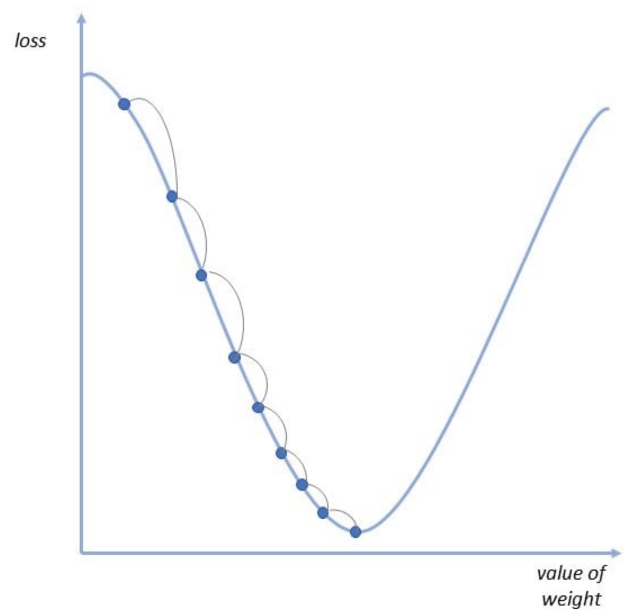
Gradient Descent

Gradient Descent

- Gradient descent (GD) is an iterative first-order optimization algorithm used to find a local minimum/maximum of a given function.

- To find $\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} f(\boldsymbol{x})$, iteratively update

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \alpha \nabla_{\boldsymbol{x}} f(\boldsymbol{x}), \text{ i.e., } x_j \leftarrow x_j - \alpha \frac{\partial}{\partial x_j} f(x)$$

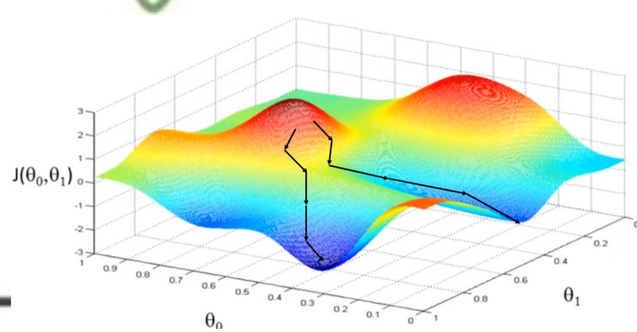


Gradient Descent

- The gradient descent algorithm guides the search for values that minimize the function at a local/global minimum

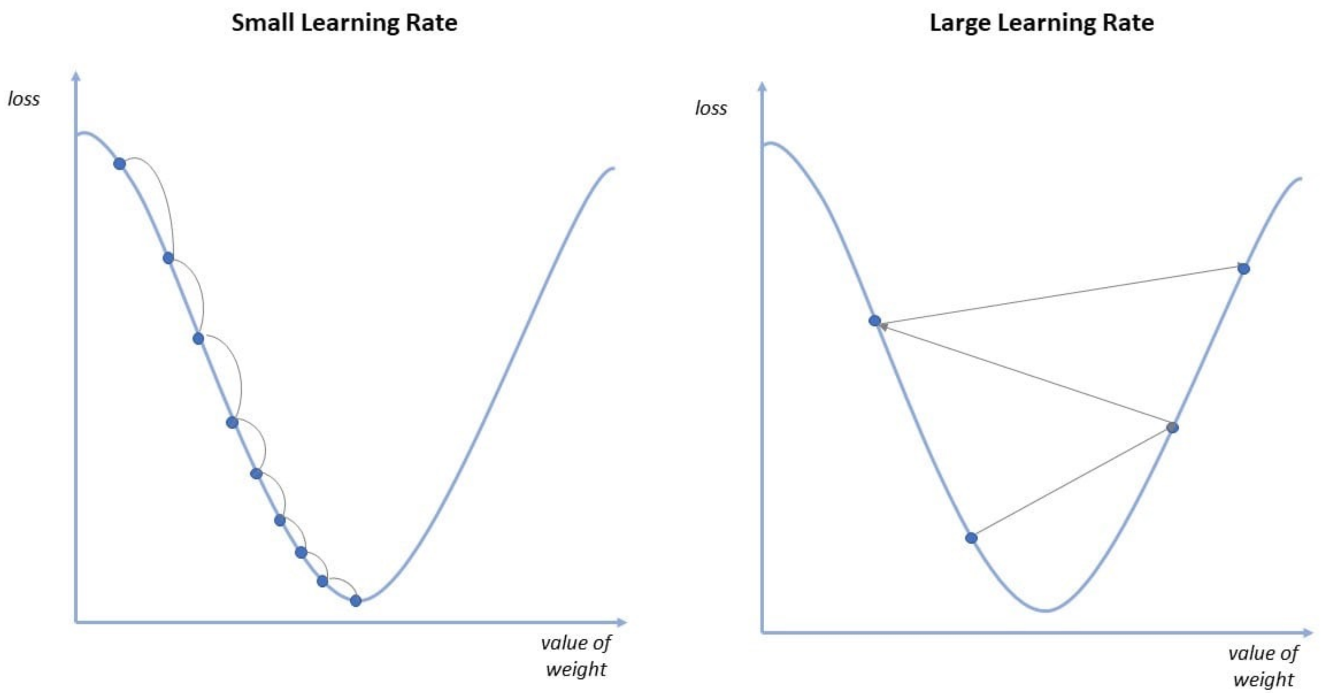
$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \alpha \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$$

- **Local** Minimum: The minimum parameter values within a specified range or sector of the cost function.
- **Global** Minimum: This is the smallest parameter value within the entire cost function domain.



Gradient Descent

- Role of the learning rate α
  - The learning rate determines the size of the steps taken towards the minimum of the loss function. A higher learning rate means larger steps, while a lower learning rate means smaller steps.



* warning : gradient descent might “overshoot” if step size is chosen incorrectly (α has to be small enough relative to curvature of the function)

Logistic Regression (again)

Learning a Logistic Regression Model

- **Logistic Regression**
  - Goal : Given data pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, find the optimal parameter $\boldsymbol{w}^*$that minimizes the training error. *(as we did in linear regression)*
  - here, we use *"binary cross-entropy"* loss to define classification error

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \{-y_i \log \widehat{y}_i - (1 - y_i) \log(1 - \widehat{y}_i)\}$$

  - There is no "closed-form". How can we solve the above problem? => *"gradient descent"*

Learning a Logistic Regression Model

- **Check the convexity**
  - Let $L(\boldsymbol{w})$ be

$$L(\boldsymbol{w}) = \sum_{i=1}^{n}\{-y_i \log \hat{y}_i - (1 - y_i)\log(1 - \hat{y}_i)\}$$

  - You want to obtain a global optimal solution $\boldsymbol{w}^*$ for $\min_{\boldsymbol{w}} L(\boldsymbol{w})$ using Gradient Descent

  - To make sure that you can find it, you have to check that $L(\boldsymbol{w})$ is convex
  - Note: To do so,
    - Calculate $\nabla_{\boldsymbol{w}}^2 L(\boldsymbol{w})$
    - Show that $\nabla_{\boldsymbol{w}}^2 L(\boldsymbol{w})$ is positive semi-definite.
    - If $\nabla_{\boldsymbol{w}}^2 L(\boldsymbol{w})$ is positive semi-definite => then, $L(\boldsymbol{w})$ is convex

Learning a Logistic Regression Model

- **Find the optimal $\boldsymbol{w}^*$**
  - By the gradient decent on $L(\boldsymbol{w})$,

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \nabla_{\boldsymbol{w}} L(\boldsymbol{w}) \quad (\alpha : \text{learning rate})$$

    - Repeat the above until convergence
  - $L(\boldsymbol{w}) = \sum_{i=1}^{n} \{-y_i \log \widehat{y_i} - (1 - y_i) \log(1 - \widehat{y_i})\}$

$$= \sum_{i=1}^{n} \{-y_i \log \sigma(\boldsymbol{w}^T \boldsymbol{x}) - (1 - y_i) \log(1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}))\}$$

  - $\nabla_{\boldsymbol{w}} L(\boldsymbol{w}) =$

  - it will converge to the optimal $\boldsymbol{w}^*$

Another View

Probabilistic Approach for Logistic Regression

- **Probabilistic Approach for Logistic Regression**
  - Start from a generative model:
    - Assume that y_i's are generated probabilistically from $\boldsymbol{x}_i$'s and a parameter $\mathbf{w}$ via

$$P(y=1) = \frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}-b)}, \qquad P(y=0) = \frac{\exp(-\boldsymbol{w}^T\boldsymbol{x}-b)}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}-b)}$$

  - Let's perform MLE to find a formula for $\mathbf{w}$:

$$L_w(\boldsymbol{X},\boldsymbol{y}) = \prod_{i=1}^{n}\left(\frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}_i-b)}\right)^{y_i}\left(\frac{\exp(-\boldsymbol{w}^T\boldsymbol{x}_i-b)}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}_i-b)}\right)^{(1-y_i)}$$

  - Maximizing $L_w(\boldsymbol{X},\boldsymbol{y})$ with respect to $\mathbf{w}$ is equivalent to

$$\min_{w}\sum_{i=1}^{n}\left\{-y_i\log\left(\frac{1}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}_i-b)}\right)-(1-y_i)\log\left(\frac{\exp(-\boldsymbol{w}^T\boldsymbol{x}_i-b)}{1+\exp(-\boldsymbol{w}^T\boldsymbol{x}_i-b)}\right)\right\}$$