BITS Pilani
Dubai Campus

Machine Learning

CS F464

Dr. Pranav M. Pawar

Contents

- Logistic Regression
- Gradient Descent

Logistic Regression (1)

- Logistic regression is a statistical and machine learning technique for classifying records of a dataset based on the values of the input fields.
- It is classification algorithm for categorical data.
- Example: A telecommunication dataset (For analysis of which customers might leave us next month.)

- Other application example
  - Predict the probability of a person having a heart attack within a specified time period.
    - Based on information such as age, sex, and body mass index.
  - Predict the likelihood of a homeowner defaulting on a mortgage.

Independent variables | Dependent variable

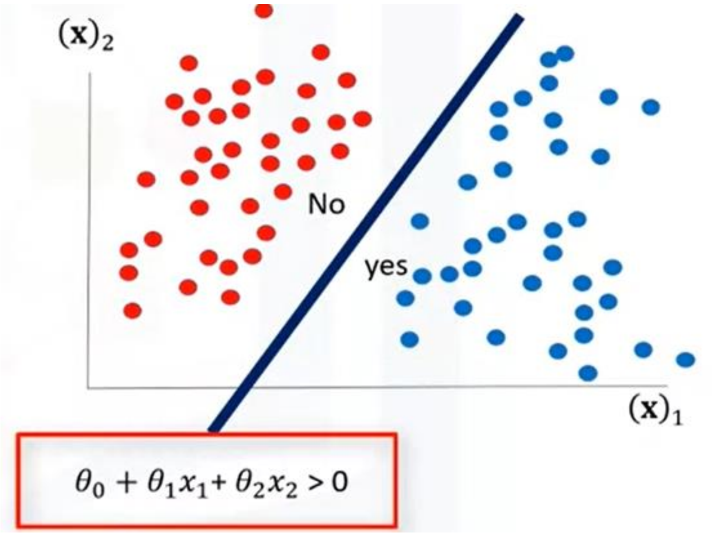
	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	Yes
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	Yes
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	No
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	No
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	?

Categorical Variable

Continuous/Categorical variables

Logistic Regression (2)

- When we should use logistic regression?
  - Target field in your data is categorical (specifically binary).
    - zero/one, yes/no, positive/negative etc.
  - If need probability of prediction.
  - If your data is linearly separable.
    - Decision boundary of logistic regression is a line or a plane or a hyper plane.
  - If you need to understand the impact of any feature.
- Two class logistic regression
  - *X* is a data set in space of real numbers *m x n*.
  - y is class which we want to predict.



$$X \in \mathbb{R}^{m \times n}$$
$$y \in \{0,1\}$$

$$\boxed{\hat{y} = P(y=1|x)}$$

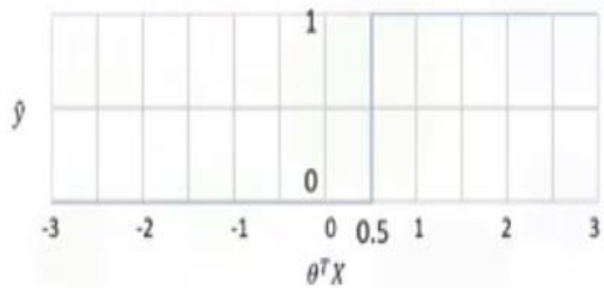
$$P(y=0|x) = 1 - P(y=1|x)$$

Linear Regression vs Logistic Regression

innovate achieve lead

- **Linear Regression for classification problem**

$$\theta^T X = \theta_0 + \theta_1 x_1 + \cdots$$

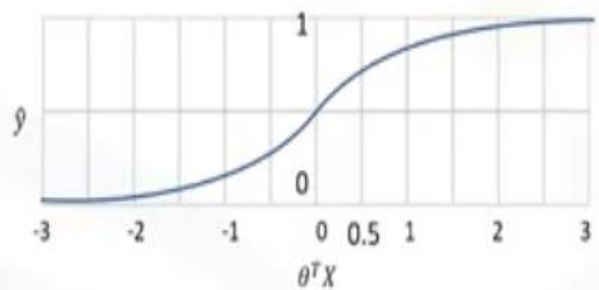


$$\hat{y} = \begin{cases} 0 & \text{if } \theta^T X < 0.5 \\ 1 & \text{if } \theta^T X \geq 0.5 \end{cases}$$

Step Function

- **Logistic Regression**

$$\sigma(\theta^T X) = \sigma(\theta_0 + \theta_1 x_1 + \cdots)$$

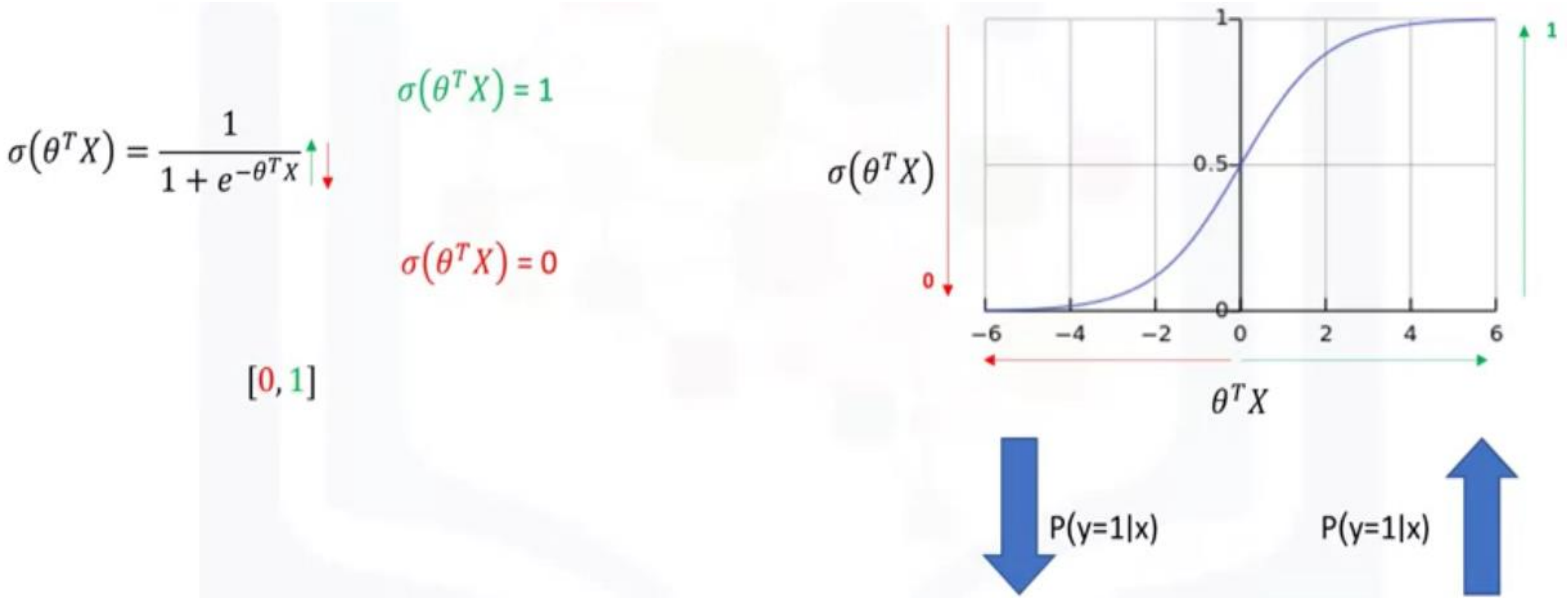


$$\hat{y} = \sigma(\theta^T X)$$

P(y=1|x)

Sigmoid (σ) of Theta transpose x gives us the probability of a point belonging to a class (in case of logistic regression) instead of the value of y (in case of linear regression).

Sigmoid Function (Logistic Function)



- Theta transpose *x* gets very big, the value of the sigmoid function gets closer to *1*.
- If Theta transpose *x* is very small, the sigmoid function gets closer to *0*.
- When the outcome of the sigmoid function gets closer to *1*, the probability of *y* equals *1* given *x* goes up.
- In contrast, when the sigmoid value is closer to *0*, the probability of *y* equals *1* given *x* is very small.

Continue with Telecom Example

- Output of model

$$P(Y=1|X)$$

$$P(y=0|X) = 1 - P(y=1|x)$$

  - Example

$$P(Churn=1|income,age) = 0.8$$

$$P(Churn=0|income,age) = 1 - 0.8 = 0.2$$

$$\boxed{\sigma(\theta^T X) \longrightarrow P(y=1|x)}$$

$$\boxed{1 - \sigma(\theta^T X) \longrightarrow P(y=0|x)}$$

To do good estimate of probabilities we need to find optimized values of *Θ. How to do it?*

Training Process

innovate achieve lead

$\sigma(\theta^T X) \longrightarrow$ P(y=1|x)

1. Initialize θ.
2. Calculate $\hat{y} = \sigma(\theta^T X)$ for a customer.
3. Compare the output of $\hat{y}$ with actual output of customer, y, and record it as error.
4. Calculate the error for all customers.
5. Change the θ to reduce the cost.
6. Go back to step 2.

$\theta = [-1,2]$

$\hat{y} = \sigma([-1,2] \times [2,5]) = 0.7$

Error = 1-0.7 = 0.3

$Cost = J(\theta)$

θ_{new}

- How to find values of Θ which will help to reduce cost across iterations?
- When we should stop iterations?

Cost Function

$$Cost(\hat{y}, y) = \frac{1}{2}\left(\sigma(\theta^T X) - y\right)^2$$

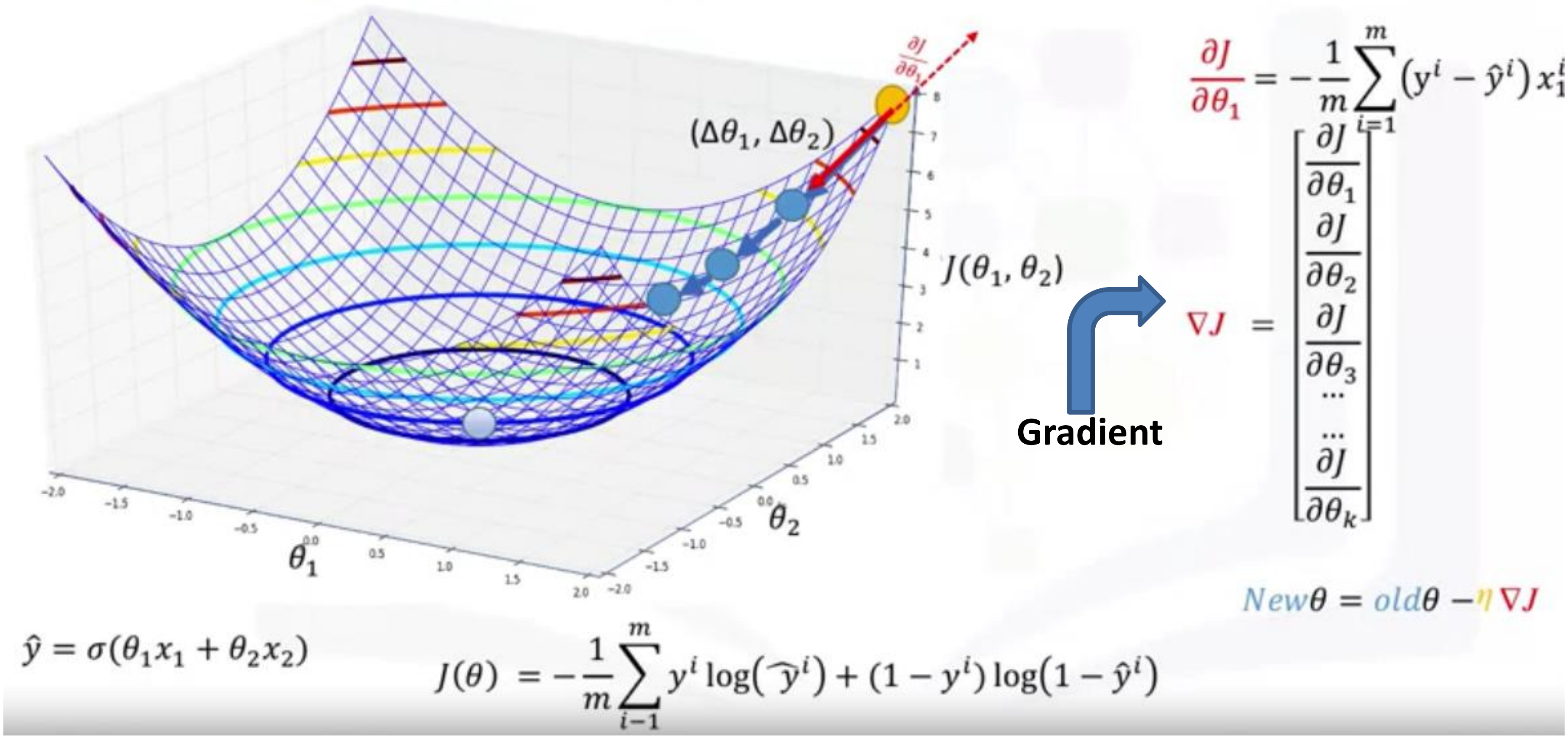
$$Cost(\hat{y}, y) = \begin{cases} -\log(\hat{y}) & if\ y = 1 \\ -\log(1-\hat{y}) & if\ y = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m}\sum_{i-1}^{m} Cost(\hat{y}, y)$$

$$\boxed{J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1-y^i)\log(1-\hat{y}^i)}$$

- Minimising cost functions help to find best parameters for model
- How to minimize it ? => Gradient Descent (GD) ?

Minimizing cost using GD



- Gradient descent is a technique to use derivative of a cost function to change the parameter values, to minimize the cost.
- More efficient and principal way for navigating a error surfaces.

Training with GD

1. initialize the parameters randomly.
2. Feed the cost function with training set, and calculate the error.
3. Calculate the gradient of cost function.
4. Update weights with new values.
5. Go to step 2 until cost is small enough.
6. Predict the new customer X.

$$\theta^T = [\theta_0, \theta_1, \theta_2, ...]$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

$$\nabla J = [\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \frac{\partial J}{\partial \theta_3}, \ldots, \frac{\partial J}{\partial \theta_k}]$$

$$\theta_{new} = \theta_{prv} - \eta \nabla J$$

$$\boxed{\mathrm{P(y=1|x)} = \sigma(\theta^T X)}$$

Types of Gradient Descent

innovate achieve lead

- Momentum based GD
- Nestrov accelerated GD
- Stochastic and Mini-batch GD
- Gradient Descent with Adaptive Learning (Adagrad)
- RMS prop (Root means square propagation)
- Adam (Adaptive moment estimation)

DL

Sources

- Chapter 3, Christopher M Bishop: Pattern Recognition & Machine Leaning, 2006 Springer.
- “Machine learning ” course, Andrew Ng
- “Machine learning with Python course”, IBM
- Chapter 3, Christopher M Bishop: Pattern Recognition & Machine Leaning, 2006 Springer.
- https://rmartinshort.jimdofree.com/2019/02/17/overfitting-bias-variance-and-leaning-curves/
- https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a

Thank You!