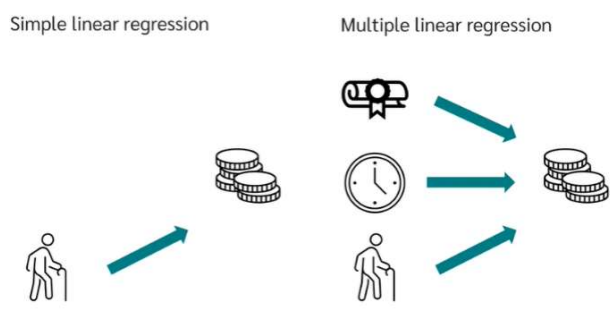
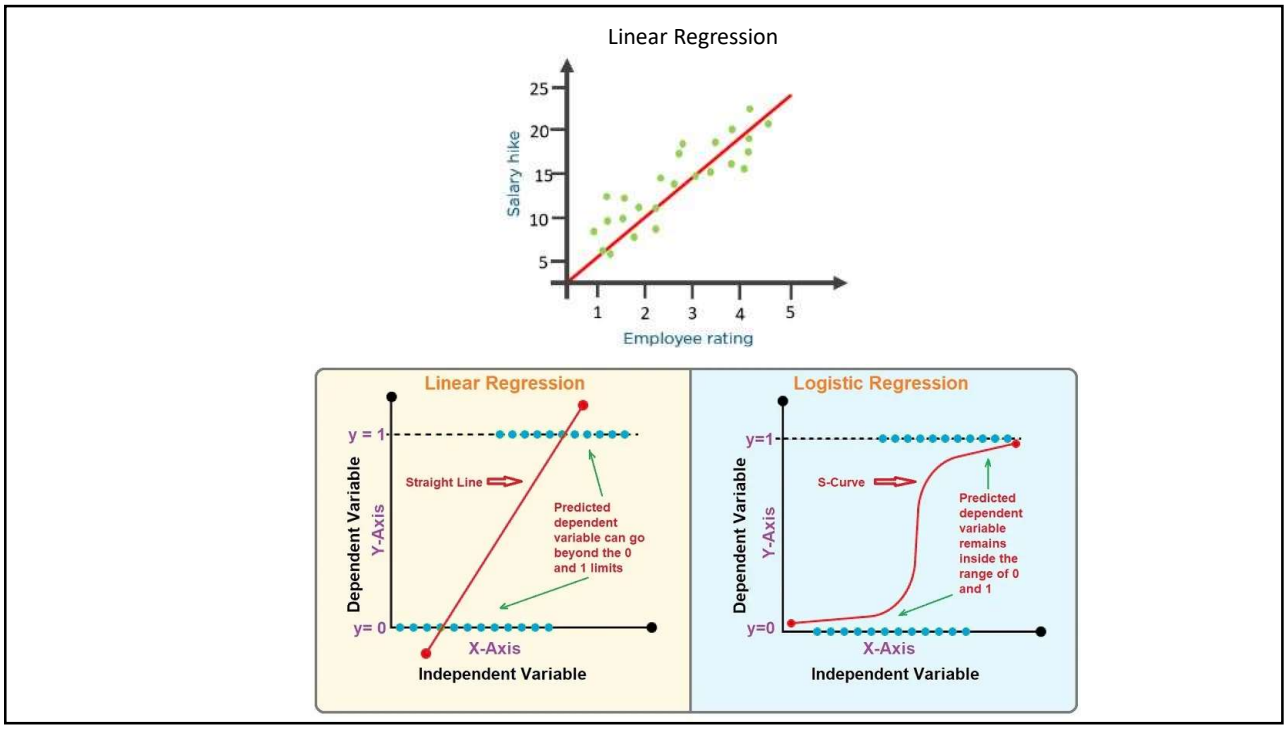
Machine Learning
Logistic Regression

1



2



3

Logistic Regression

Name is somewhat misleading. It is really a **technique for classification**, ~~**not regression**~~.

"Regression" comes from fact that we fit a linear model to the feature space.

Involves a more probabilistic view of classification. In regression analysis, logistic regression or logit regression is estimating the parameters of a logistic model.

Models relationship between set of input variables $\boldsymbol{X_i}$ (**categorical or continuous**) and the response variable **Y** (**categorical**).

The dependent variable is binary rather than continuous and it can also be applied to ordered categories (ordinal data).

4

4

Logistic Regression

Logistic regression is a special case of regression analysis and is calculated when the dependent variable is nominally or ordinally scaled.

Business example:

For an online retailer, you need to predict which product a particular customer is most likely to buy. For this, you receive a data set with past visitors and their purchases from the online retailer.

Medical example:

You want to investigate whether a person is susceptible to a certain disease or not. For this purpose, you receive a data set with diseased and non-diseased persons as well as other medical parameters.

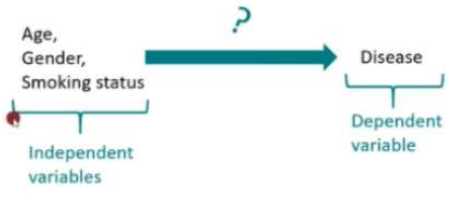
Political example:

Would a person vote for party A if there were elections next weekend?

5

- **Dichotomous variables (0 or 1)** can be predicted by using a logistic regression.
- The **probability of occurrence** of characteristic 1 (=characteristic present) is estimated.

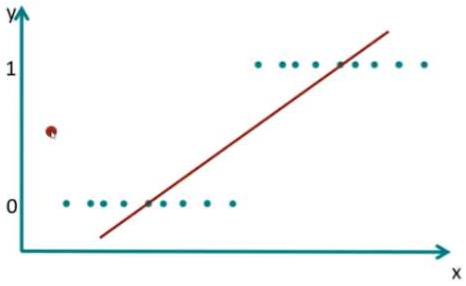
- In medicine, for example, a common goal is to find out which variables have an impact on a disease.
- In this case, 0 could stand for "not diseased" and 1 for "diseased" and the influence of age, gender and smoking status on this particular disease is examined.



6

Why not just use linear regression?

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$



- The graph shows that values between **plus and minus infinity** can now occur.
- The goal of logistic regression is to estimate the **probability of occurrence**, not the value of the variable itself.
- The range of values for the prediction is restricted to the range between 0 and 1.
- Since only values between 0 and 1 are possible, the logistic function f is used.

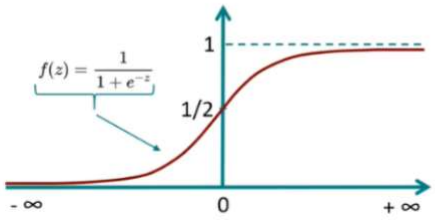
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \varepsilon$$

$$Y = \begin{cases} 1, \text{ if YES} \\ 0, \text{ NO} \end{cases}$$

7

Logistic function

The logistic model is based on the logistic function.
The important thing about the logistic function is, that only values **between 0 and 1** are possible.



$$f(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}$$

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

8

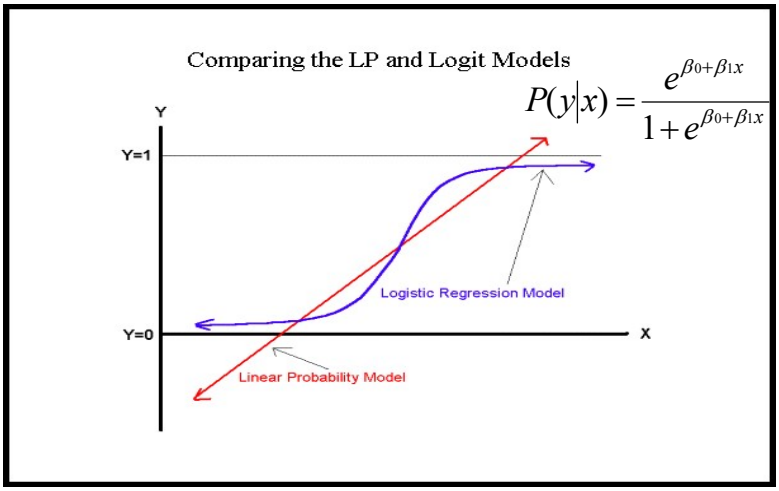
Estimating the Logit Model

- We use method of *maximum likelihood estimation (MLE)* to estimate the parameters of the *logistic regression model*:

$$\ell n\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

9

Comparing Linear and Logistic Regression Models



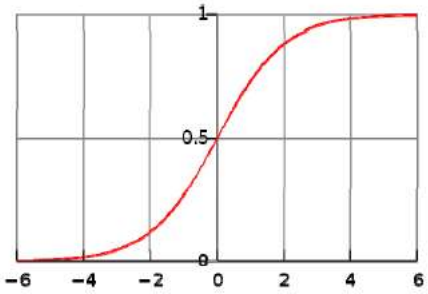
10

10

$$z = \log\left(\frac{p}{1-p}\right)$$ logit function

$$\frac{p}{1-p} = e^{z}$$ **Odds**

$$p = \frac{e^{z}}{1+e^{z}} = \frac{1}{1+e^{-z}}$$ logistic function



Standard Logistic Function

11

11

The term "Odds"

- Popular in horse races, sports, gambling etc.

- Instead of talking about the *probability* of winning or contacting a disease, people talk about the ***odds* of winning or affecting** with the disease.

e.g.:

- **Probability statement**: The probability of a person contracting the flu this season is 20%.
- **Odds statement**: The odds of a person contracting the flu this season are 1 to 4.

12

Logit Function in Logistic Regression

In logistic regression, the dependent variable is a logit, which is the natural log of the odds, that is,

$$odds = \frac{P}{1-P}$$

$$\log(\text{odds}) = \text{logit}(P) = \ln\left(\frac{P}{1-P}\right)$$

13

Math behind Logistic Regression

So a logit is a log of odds.
Odds is a function of P. P is the probability of a 1.
In Logistic Regression, we find : **logit(P) = *a + bX***

$$\ln\left(\frac{P}{1-P}\right) = a + bX$$

$$\frac{P}{1-P} = e^{a+bX}$$

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

14

Different ways of expressing Probabilities

- Consider a two-outcome probability space, where:
  - $p(O_1) = p$
  - $p(O_2) = 1 - p = q$
- Can express probability of O_1 as:

	notation	range equivalents		
standard probability	p	0	0.5	1
odds	p / q	0	1	+ ∞
log odds (logit)	log(p / q)	- ∞	0	+ ∞

15

Odds vs. Probability

- What is probability of A: P(A)?
- $Odds\ ratio = \frac{Probability\ of\ event\ occuring}{Probability\ of\ event\ not\ occuring}$
- Odds ratio = $\frac{P}{1-P}$
- Probability= $\frac{Odds\ ratio}{1+Odds\ ratio}$

A bag contains **10 balls**: **3 red balls, 4 blue balls, 3 green balls**
- Probability of drawing a red ball = 30% (or 0.3)
- Odds of drawing a red ball = 3 to 7 (**Odds= favorable outcomes/ unfavorable outcomes**)

Disease Example
•**Probability:** *The probability of a person getting infected with Disease X is 25%.*
•**Odds:** *The odds of getting infected with Disease X are 1 to 3 (since 25% means 25 people get infected out of 100, and 75 do not, so 75:25 simplifies to 3:1).*
Sports Example
•**Probability:** *The probability of a soccer team winning a match is 40%.*
•**Odds:** *The odds of the team winning are 2 to 3 (since 40% means 40 wins out of 100, and 60 losses/draws, so 60:40 simplifies to 3:2, or 2:3 in favor of losing).*
Coin Toss Example
•**Probability:** *The probability of flipping heads on a fair coin is 50% (or 0.5).*
•**Odds:** *The odds of flipping heads are 1 to 1 (since for every 1 heads, there's 1 tails).*

16

Math Behind Logistic Regression

- Predict likelihood or probability
- Predicted value P should lie between 0 and 1
- Use Sigmoid function to achieve this

$$Probability, P = \frac{e^z}{1+e^z}$$

$z = \beta_0 + \beta_1 x$

$$Odds\ Ratio = \frac{P}{1-P}$$

Substituting for P, $Odds\ Ratio = \frac{P}{1-P} = e^z = e^{(\beta_0+\beta_1 x)}$

$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x$$

Log(Odds) takes the form of linear regression
intercept β_0 and slope β_1
β_0 and slope β_1 estimated using maximum likelihood estimation

17

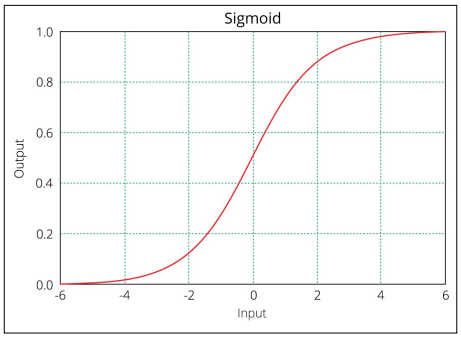
Equation of Logistic Regression

log - odds or **odds ratio** or **logit** function and is the link function for Logistic Regression

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x$$

Regression intercept & coefficient

This link function follows a sigmoid function which limits its range of probabilities between 0 and 1.



18

Logistic Regression

- A multidimensional feature space (features can be categorical or continuous).
- – Outcome is **discrete**, not continuous.

We'll focus on case of two classes.
– A linear decision boundary (hyperplane) will give good predictive accuracy.

19

19

Using Logistic Regression Model

- Model consists of a vector β in *d*-dimensional feature space
- For a point **x** in feature space, project it onto β to convert it into a real number *z* in the range - ∞ to + ∞

$$z = \alpha + \boldsymbol{\beta} \cdot \mathbf{x} = \alpha + \beta_1 x_1 + \cdots + \beta_d x_d$$

- Map *z* to the range 0 to 1 using the logistic function

$$p = 1/(1+e^{-z})$$

- Overall, logistic regression maps a point **x** in *d*-dimensional feature space to a value in the range 0 to 1

Logit Transformation

The logistic regression model is given by

$$P(Y \mid X) = \frac{e^{\beta_o + \beta_1 X}}{1 + e^{\beta_o + \beta_1 X}}$$

which is equivalent to

$$\ln\left(\frac{P(Y \mid X)}{1 - P(Y \mid X)}\right) = \beta_o + \beta_1 X$$

This is called the Logit Transformation

20

20

Using Logistic Regression Model

Can interpret prediction from a logistic regression model as:

- – A probability of class membership
- – A class assignment, by **applying threshold to probability**
- Threshold represents decision boundary in feature space

21

21

Training a Logistic Regression Model

Optimization of model parameters:

Need to optimize $\boldsymbol{\beta}$ so the model gives the best possible reproduction of training set labels

– Usually done by numerical approximation or maximum likelihood

– On really large datasets, may use stochastic gradient descent

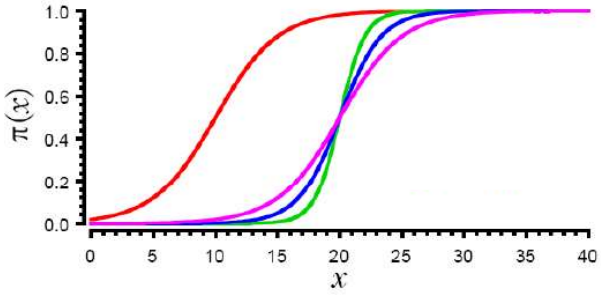
22

22

Parameters of the Model and Shape of the Function

- Parameters control shape and location of sigmoid curve
  - α controls location of midpoint
  - β controls slope of rise

$$\pi(x) = \exp(\alpha + \beta x) / (1 + \exp(\alpha + \beta x))$$



23

23

The Logistic Regression Model

The "logit" model solves these problems:

$\ln[p/(1-p)] = \beta_0 + \beta_1 X$

p is the probability that the event Y occurs, *p(Y=1)*
[range=0 to 1]

p/(1-p) is the "odds ratio" [range=0 to ∞]

ln[p/(1-p)]: log odds ratio, or <u>logit</u> [range=-∞ to +∞]

24

Making Prediction with Sigmoid

- Using our knowledge of sigmoid functions and decision boundaries, we can now write a prediction function.
- A prediction function in logistic regression returns the probability of our observation being positive, True, or "Yes".
- We call this class 1 and its notation is P(class=1).
- As the probability gets closer to 1, our model is more confident that the observation is in class 1.

25

Making Prediction with Sigmoid

$$z = W_0 + W_1 Studied + W_2 Slept$$

Transform the output using the sigmoid function to return a probability value between 0 and 1.

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

If the model returns .4 it believes there is only a 40% chance of passing. If our decision boundary was .5, we would categorize this observation as "Fail.""

26

Cost Function

- Unfortunately, we can't (or at least shouldn't) use the same cost function MSE as we did for linear regression. **Why?**
- Because our prediction function is non-linear (due to sigmoid transform).
- Squaring this prediction as we do in MSE results in a non-convex function with many local minimums. If our cost function has many local minimums, gradient descent may not find the optimal global minimum.
- Instead of Mean Squared Error, we use a cost function called **Cross-Entropy,** also known as Log Loss. Cross-entropy loss can be divided into two separate cost functions: one for 0 and one for 1.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \quad \text{if } y = 1$$
$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \quad \text{if } y = 0$$

- The benefits of taking the logarithm reveal themselves when you look at the cost function graphs for y=1 and y=0. These smooth monotonic functions (always increasing or always decreasing) make it easy to calculate the gradient and minimize cost.

27

What is the use of Maximum Likelihood Estimator?

The primary objective of Maximum Likelihood Estimation (MLE) in machine learning, particularly in the context of logistic regression, is to identify parameter values that maximize the likelihood function.

If for this experiment a random variable X is defined such that it takes value 1 when S occurs and 0 if F occurs, then X follows a Bernoulli Distribution.

probability density function

$$P(n) = \begin{cases} 1-p & \text{for } n = 0 \\ p & \text{for } n = 1, \end{cases}$$

which can also be written

$$P(n) = p^n (1-p)^{1-n}.$$

28

1. Logistic Regression Model

Logistic regression models the probability of a binary outcome $y \in \{0, 1\}$ given input features x. The model assumes that:

$$P(y = 1 \mid x; \theta) = \sigma(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

where:

- θ is the vector of model parameters (weights),
- x is the feature vector,
- $\sigma(z)$ is the **sigmoid function**.

For $y = 0$:

$$P(y = 0 \mid x; \theta) = 1 - \sigma(\theta^T x)$$

29

2. Likelihood Function

Given a dataset $\{(x_i, y_i)\}_{i=1}^m$, the likelihood function is the joint probability of all samples:

$$L(\theta) = \prod_{i=1}^{m} P(y_i \mid x_i; \theta)$$

Substituting the probability from the logistic regression model:

$$L(\theta) = \prod_{i=1}^{m} \sigma(\theta^T x_i)^{y_i} (1 - \sigma(\theta^T x_i))^{1 - y_i}$$

3. Log-Likelihood Function

Since it's easier to work with sums rather than products, we take the logarithm:

$$\log L(\theta) = \sum_{i=1}^{m} \left[y_i \log \sigma(\theta^T x_i) + (1 - y_i) \log(1 - \sigma(\theta^T x_i)) \right]$$

This is called the **log-likelihood function**, and it is the function we want to **maximize** to find the best parameters θ.

30

4. Cost Function for Gradient Descent

Instead of maximizing the log-likelihood, we define a cost function $J(\theta)$ as the **negative log-likelihood** (to convert maximization into minimization):

$$J(\theta) = -LL(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log \sigma(\theta^T x_i) + (1-y_i)\log(1-\sigma(\theta^T x_i))\right]$$

This function is convex and can be **minimized using gradient descent**.

5. Gradient Descent Update Rule

To minimize $J(\theta)$, we compute its gradient with respect to θ:

$$\frac{\partial J}{\partial \theta}$$

Using **gradient descent**, we update the parameters iteratively:

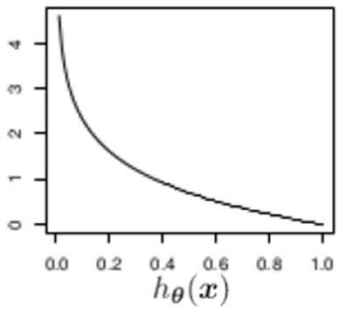
$$\theta := \theta - \alpha \cdot \frac{\partial J}{\partial \theta}$$

where α is the learning rate.

31

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

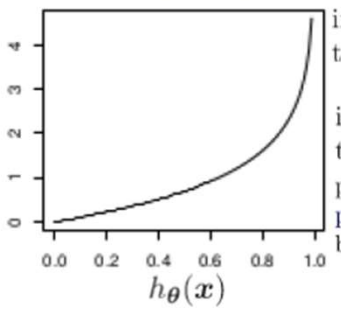
if $y = 1$



if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 1$
then cost $= 0$

if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 0$
then cost $\to \infty$
predicted
$\text{prob}(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$
but $y = 1$

if $y = 0$

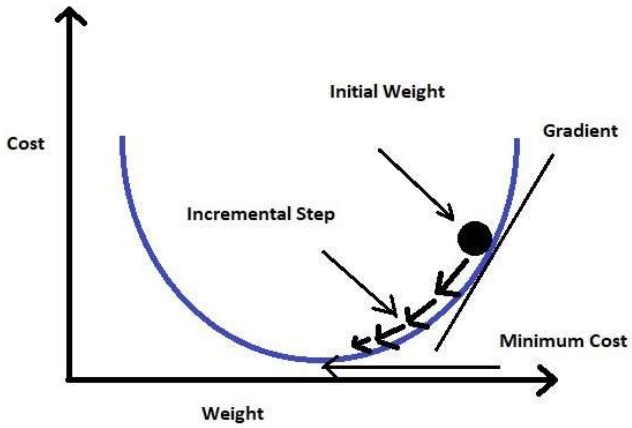


if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$
then cost $= 0$

if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 1$
then cost $\to \infty$
predicted
$\text{prob}(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$
but $y = 0$

32

Gradient Descent Optimization



33

Step-1: Use chain rule and break the partial derivative of log-likelihood.

$$\partial J(\theta)/\partial \theta j = -\frac{\partial LL(\theta)}{\partial \theta_j} = -\frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial \theta} \qquad where\ p = \sigma\left[\theta^T x\right]$$

$$= -\frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial \theta_j} \qquad where\ z = \theta^T x$$

Step-2: Find derivative of log-likelihood w.r.t p

We know,

$$LL(\theta) = y\ \log(p) + (1-y)\log(1-p) \qquad \text{where } p = \sigma\left[\theta^T x\right]$$

$$\frac{\partial LL(\theta)}{\partial p} = \frac{y}{p} - \frac{(1-y)}{(1-p)}$$

34

We can summarize the gradient descent algorithm as: $\theta_{new} = \theta_{old} - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$

Derivation of Cost Function:

$$\frac{d\sigma(x)}{dx} = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = \frac{d}{dx}(1+e^{-x})^{-1}$$

$$\Rightarrow -(1+e^{-x})^{-2} \times \frac{d}{dx}(1+e^{-x})$$

$$\Rightarrow -(1+e^{-x})^{-2} \times [0 + \frac{d}{dx}(e^{-x})]$$

$$\Rightarrow -(1+e^{-x})^{-2} \times [e^{-x} \times \frac{d(-x)}{dx}]$$

$$\Rightarrow (1+e^{-x})^{-2} \times [e^{-x} \times 1]$$

$$\Rightarrow e^{-x}(1+e^{-x})^{-2}$$

$$\Rightarrow \frac{e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}+1-1}{(1+e^{-x})(1+e^{-x})}$$

$$\Rightarrow \frac{(1+e^{-x})-1}{(1+e^{-x})(1+e^{-x})} = \frac{1}{(1+e^{-x})}[\frac{(1+e^{-x})}{(1+e^{-x})} - \frac{1}{(1+e^{-x})}]$$

$$\Rightarrow \frac{1}{(1+e^{-x})}[1 - \frac{1}{(1+e^{-x})}]$$

35

Step-3: Find derivative of ***'p'*** w.r.t ***'z'***

$$p = \sigma(z)$$

$$\frac{\partial p}{\partial z} = \frac{\partial[\sigma(z)]}{\partial z}$$

We know the derivative of sigmoid function is $\sigma\left[\theta^T x\right]\left[1-\sigma\left(\theta^T x\right)\right]$

$$\Rightarrow \frac{\partial p}{\partial z} = \sigma[z][1-\sigma(z)]$$

36

Step-4: Find derivate of z w.r.t θ

$$z = \theta^{\mathrm{T}} \mathrm{x}$$

$$\frac{\partial z}{\partial \theta_j} = x_j$$

Step-5: Put all the derivatives

$$-\frac{\partial \mathrm{LL}(\theta)}{\partial \theta_{\mathrm{j}}} = -\frac{\partial \mathrm{LL}(\theta)}{\partial \mathrm{p}} \cdot \frac{\partial \mathrm{p}}{\partial \mathrm{z}} \cdot \frac{\partial \mathrm{z}}{\partial \theta_{\mathrm{j}}}$$

$$-\frac{\partial \mathrm{LL}(\theta)}{\partial \theta_{\mathrm{j}}} = -\left[\frac{y}{p} - \frac{(1-y)}{(1-p)}\right].\sigma(z)\sigma(1-(z)).x_j$$

$$= -\left[\frac{y}{p} - \frac{(1-y)}{(1-p)}\right].p[1-p].x_j \qquad \text{since } p = \sigma[z]$$

$$= -[\mathrm{y}(1-\mathrm{p}) - \mathrm{p}(1-\mathrm{y})].\mathrm{x}_{\mathrm{j}} \qquad = -[y-p].x_j$$

Hence the derivative of our cost function is:

$$\Rightarrow [p-y].x_j = [\sigma(\theta^T x) - y].x_j$$

$$\theta_{new} = \theta_{old} - \alpha\left[\sigma\left(\theta^T x\right) - y\right].x_j$$

37

Cost Function

- To minimize our cost, we use Gradient Descent just like before in Linear Regression.
- There are other more sophisticated optimization algorithms out there such as conjugate gradient like BFGS, but you don't have to worry about these.
- One of the neat properties of the sigmoid function is its derivative is easy to calculate.

$$\sigma'(z) = \sigma(z)(1-\sigma(z))$$

- Which leads to an equally beautiful and convenient cost function derivative:

$$C' = x(\sigma(z) - y)$$

- C' is the derivative of cost with respect to weights
- y is the actual class label (0 or 1)
- $\sigma(z)$ is your model's prediction
- x is your feature or feature vector.

Notice how this gradient is the same as the MSE gradient of Linear Regression, the only difference is the **hypothesis function**.

38

Multi-class Classification

- Instead of y=0,1 we will expand our definition y=0,1,2,3...n-1 (n-class classification problem).
- Basically, we rerun binary classification multiple times, once for each class.

Procedure:

1. Divide the problem into **n binary classification problems**.
2. For each class we do the following:
   - Predict the probability the observations are in that single class.
   - Prediction = max {probability of all the classes}
   - For each sub-problem, we select one class (YES) and dump all the others into a second class (NO).
   - Then we take the class with the highest predicted value.

39

Softmax Classification

- The softmax function (*softargmax* or *normalized exponential function*) is a function that takes as input a vector of ***K*** real numbers; and normalizes it into a probability distribution consisting of ***K*** probabilities proportional to the exponentials of the input numbers.
- That is, prior to applying softmax, some vector components could be negative, or greater than one; and might not sum to 1; but after softmax, each component will be in the interval [0, 1], and the components will add up to 1, so that they can be interpreted as probabilities.
- The standard softmax function is defined by formula

$$\sigma(z_i) = \frac{e^{z_{(i)}}}{\sum_{j=1}^{K} e^{z_{(j)}}} \quad for\ i = 1, ., ., ., K\ and\ z = z_1, ., ., ., z_K$$

- In other words, we apply the standard exponential function to each element z_i of the input vector z and normalize these values by dividing the sum of all these exponentials; this normalization ensures that the sum of the components of the output vector σ(z) is 1.

40

Logistic Regression

Advantages:

– Makes no assumptions about distributions of classes in feature space

– Easily extended to multiple classes (multinomial regression)

–Can interpret model coefficients as indicators of feature importance

Disadvantages:

- Linear Decision Boundary

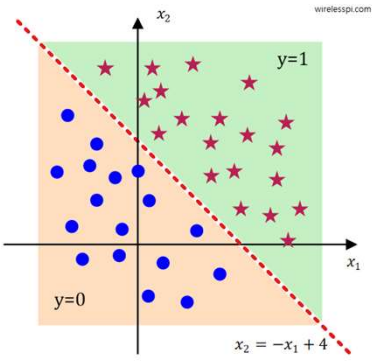
41

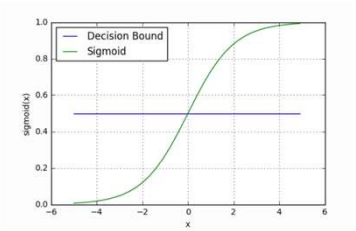
41

- **DECISION BOUNDARY:**

- While training a classifier on a dataset, using a specific classification algorithm, it is required to define a set of hyper-planes, called Decision Boundary, that separates the data points into specific classes, where the algorithm switches from one class to another.

- On one side of a decision boundary, data points are more likely to be called class A — on the other side of the boundary, it's more likely to be called class B.





42

IMPORTANCE OF DECISION BOUNDARY

- A decision boundary separates data points belonging to different class labels.
- Decision boundaries span the entire feature space we trained on, allowing the model to predict values for any possible combination of inputs.
- If the training data is not diverse, the model may generalize poorly to new instances.
- It's important to analyze models suitable for diverse datasets before using them in production.
- Examining decision boundaries helps understand how training data affects performance and generalization.
- Visualization of decision boundaries shows how sensitive models are to different datasets, helping understand algorithms and their limitations.

43