Machine Learning

| Logistic Regression

Prof. Dr. Mostafa Elhosseini

https://youtube.com/@ElhosseiniAcademy

Agenda

- Logistic Regression
- Sigmoid Function
- Model Equations
- Cost function
- Log Loss
- Model Training

Intro

- **Classification** techniques are an essential part of machine learning and data mining applications.
- Approximately **70%** of problems in Data Science are **classification** problems.
- There are lots of classification models that are available
- some regression algorithms can be used for classification

Intro

- The **logistics regression** is **common** and is a useful regression method for solving the **binary classification** problem – **dichotomous**
  - Spam detection
  - Customer purchase or churn
  - Loan application
  - User will click on a given advertisement or not
  - cancer detection problems

Intro

- Perceptron – Sentence was happy or sad
- Some sentences are happier than others
  - I’m good
  - Today was the most wonderful day in my life!
- Nice to have a classifier that **not only predicts** if sentences are happy or sad **but** that **gives a rating** for how happy sentences are
- A classifier that tells us that the first sentence is 60% happy and the second one is 95% happy

Logistic Classifier

- Continuous perceptrons: return an answer that can be any number in the interval between 0 and 1.

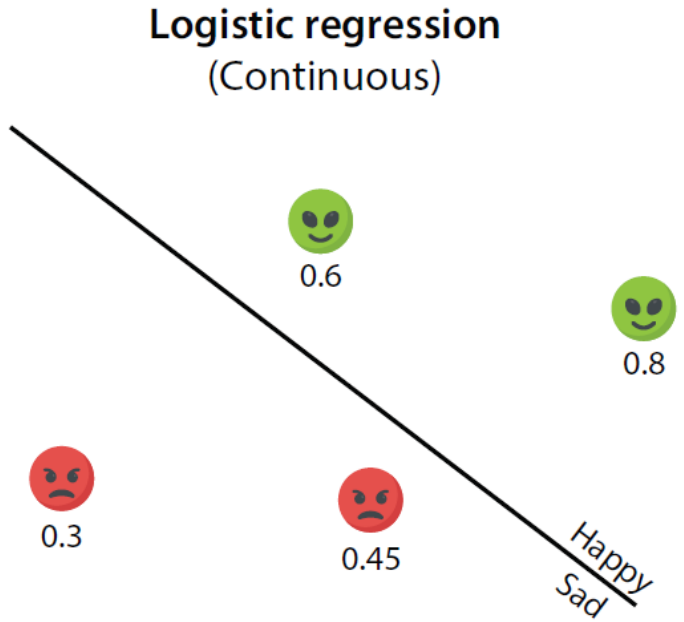
- The output of a logistic classifier can be interpreted as a score.

- The goal of the logistic classifier is to **assign scores as close as** possible to the **label** of the points
  - Points with label 0 should get scores close to 0, and
  - Points with label 1 should get scores close to 1.

Logistic regression

- Logistic Regression (also called Logit Regression) is commonly used to **estimate the probability** that an instance belongs to a particular class
  - what is the probability that this email is spam?

- Points in the positive zone get scores larger than 0.5
- The points even farther away from the 0.5 line in the positive direction get values closer to 1.
- No point gets a value of 1 or 0



Sigmoid function

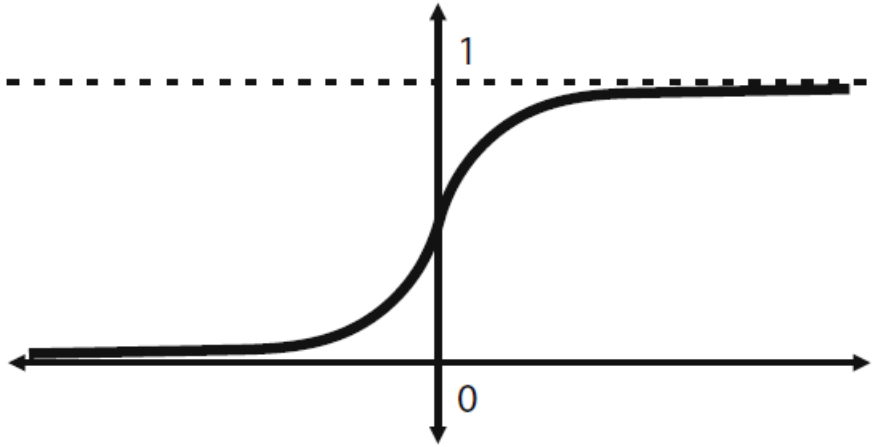
- The step function returns a 1 if the score was nonnegative and a 0 if it was negative.

Sigmoid function

- We take a function that receives the score as the input and outputs a number between 0 and 1.
- The number is close to 1 if the score is positive and close to zero if the score is negative.
- If the score is zero, then the output is 0.5

Sigmoid function

- $\sigma(x) = \frac{1}{1+e^{-x}}$: Crunching the real number line into the interval (0,1)
- Having continuous predictions gives us **more information** than discrete predictions.
- Has a much nicer derivative than the step function
- $\lim_{x\to\infty} \sigma(x) = 1 \quad \lim_{x\to-\infty} \sigma(x) = 0$
- $\lim_{x\to 0} \sigma(x) = 0.5$



- for large negative inputs, the output is close to 0, whereas for large positive inputs, the output is close to 1

Model Equation

- Logistic Regression model computes a **weighted sum** of the input features (plus a bias term), but instead of outputting the result directly like the Linear Regression model does, it **outputs the logistic** of this result

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

- $\hat{y} = \sigma(b + \sum_{i=1}^{n} \theta_i x_i)$
- $\hat{y} = \sigma(b + \sum_{i=1}^{n} w_i x_i)$

- $\hat{y} = h_\theta(x) = \sigma(\theta^T . X)$
  - $\hat{y}$: probability that an instance x is belongs to a class
  - θ : model parameters that determine boundary decision
  - X: features
  - σ: logistic (sigmoid – Logit) S-shaped function

Cost function

What properties would you like a good error function to have?

- If a point is correctly classified, the error is a small number.
- If a point is incorrectly classified, the error is a large number.
- The error of a classifier for a set of points is the sum (or average) of the errors at all the points.

Log Loss

- Most of the error functions have the word error in their name, but this one **instead** has the word **loss** in its name.

- The **log** part in the name comes from a **natural logarithm** that we use in the formula.

- However, the **real soul** of the log loss is **probability**.

Log Loss

- The **outputs** of a continuous perceptron are numbers between **0 and 1**, so they can be considered **probabilities**.
- The model **assigns a probability to every data point**, and that is the **probability that the point is (class 1 – Positive class)**.
- From this, we can infer the probability that the point is **(class 0 – Negative class)**, which is **1 minus** the probability of being happy
- if the prediction is 0.75, that means the model believes the point is happy with a probability of 0.75 and sad with a probability of 0.25

Log Loss – Sentiment Analysis

- The goal of the model is to assign high probabilities to the happy points (those with label 1) and low probabilities to the sad points (those with label 0).

Point 1: Label = 0 (sad)

- Prediction (probability of being happy) = 0.953
- Probability of being its label: 1 – 0.953 = **0.047**

Point 2: Label = 1 (happy)

- Prediction (probability of being happy) = 0.731
- Probability of being its label: 0.731

Point 3: Label = 1 (happy)

- Prediction (probability of being happy) = 0.119
- Probability of being its label: **0.119**

Point 4: Label = 0 (sad)

- Prediction (probability of being happy) = 0.119
- Probability of being its label: 1 – 0.119 = 0.881

Log Loss – Sentiment Analysis

Point 1: Label = 0 (sad)

- Prediction (probability of being happy) = 0.953
- Probability of being its label: 1 – 0.953 = **0.047**

Point 2: Label = 1 (happy)

- Prediction (probability of being happy) = 0.731
- Probability of being its label: 0.731

Point 3: Label = 1 (happy)

- Prediction (probability of being happy) = 0.119
- Probability of being its label: **0.119**

Point 4: Label = 0 (sad)

- Prediction (probability of being happy) = 0.119
- Probability of being its label: 1 – 0.119 = 0.881

- points **2 and 4** are the points that are **well classified**, and the model assigns a **high probability** that they are their own label.
- In contrast, **points 1 and 3** are **poorly** classified, and the model assigns a **low probability** that they are their own label

Log Loss – Sentiment Analysis

- The goal of the logistic classifier is to **assign** to each point the **highest** possible **probability** of **having the correct label**. This classifier assigns the probabilities 0.047, 0.731, 0.119, and 0.881 to the four labels. Ideally, we'd like these numbers to be higher

Point 1: Label = 0 (sad)

- Prediction (probability of being happy) = 0.953
- Probability of being its label: 1 – 0.953 = **0.047**

Point 2: Label = 1 (happy)

- Prediction (probability of being happy) = 0.731
- Probability of being its label: 0.731

Point 3: Label = 1 (happy)

- Prediction (probability of being happy) = 0.119
- Probability of being its label: **0.119**

Point 4: Label = 0 (sad)

- Prediction (probability of being happy) = 0.119
- Probability of being its label: 1 – 0.119 = 0.881

Log Loss – Sentiment Analysis

How do we measure these four numbers?

- One way would be to **add** them or **average** them. But because they are **probabilities**, the natural approach is to **multiply them**.
- When events are **independent**, the probability of them occurring simultaneously is the **product of their probabilities**.
- The probability that this model assigns to the labels “sad, happy, happy, sad” is the **product of the four** numbers, which is $0.047 \cdot 0.731 \cdot 0.119 \cdot 0.881 = 0.004$. This is a very small probability
- Our hope would be that a **model that fits** this dataset better would result in a **higher probability**

Log Loss – Sentiment Analysis

- Probability we just calculated seems like a good measure for our model, but it has some problems.
  - Products of many small numbers tend to be tiny
  - Imagine if our dataset had one million points. The probability would be a product of one million numbers, all between 0 and 1.
  - This number may be so small that a computer may not be able to represent it. Also, manipulating a product of one million numbers is extremely difficult.

Log Loss – Sentiment Analysis

- Is there any way that we could perhaps **turn it into** something easier to manipulate, like a **sum**?
- use the logarithm – it turns products into sums
- the logarithm of a product of two numbers is the sum of the logarithms of the numbers
- $\ln(a.b) = \ln(a) + \ln(b)$
- We can use logarithms in base 2, 10, or e
- we use the natural logarithm, which is on base e

Log Loss – Sentiment Analysis

- $\ln(a.b) = \ln(a) + \ln(b)$
- $\ln(20) = 2.9957$ $\ln(1) = 0$
- The logarithm of a number between 0 and 1 is always negative

- $\ln(0.9) = -0.10536$ $\ln(0.6) = -0.510826$ $\ln(0.1) \approx -2.3026$

- if a number x is close to 0, $-\ln(x)$ is a large number, but if x is close to 1, then $-\ln(x)$ is a small number

Log Loss – Sentiment Analysis

- $\ln(0.047 \times 0.731 \times 0.119 \times 0.881) = -5.616$

- if we take the negative logarithm of the product of probabilities, it is always a positive number.

- The **log loss** is defined as the **negative logarithm of the product** of probabilities, which is also the **sum of the negative logarithms** of the probabilities

Log Loss – Sentiment Analysis

Point	Label	Predicted label	Probability of being its label	Log loss
1	0 (Sad)	0.953	0.047	–ln(0.047) = 3.049
2	1 (Happy)	0.731	0.731	–ln(0.731) = 0.313
3	1 (Happy)	0.119	0.119	–ln(0.119) = 2.127
4	0 (Sad)	0.119	0.881	–ln(0.881) = 0.127

- if a number x is close to 0, $-\ln(x)$ is a large number, but if x is close to 1, then $-\ln(x)$ is a small number
- the well-classified points (2 and 4) have a small log loss, and the poorly classified points have a large log loss

Log Loss

- If the label is 0: log loss $= -\ln(1 - \hat{y})$
- If the label is 1: log loss $= -\ln(\hat{y})$
- Because the label is y, the previous if statement can be condensed into the following formula
- log loss = $-y\ln(\hat{y}) - (1 - y)\ln(1 - \hat{y})$
- We use the term **log loss** when we refer to the **log loss of a point** or of a **whole dataset**.
- The log loss of a dataset is the sum of the log losses at every point.

Model Training

- The objective of training is to set the parameter vector θ so that the model estimates high probabilities for positive instances ($y = 1$) and low probabilities for negative instances ($y = 0$).
- log loss = $-y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y})$

- This cost function makes sense because $-\ln(t)$ grows very large when t approaches 0, so the cost will be large if the model estimates a probability close to 0 for a positive instance, and it will also be very large if the model estimates a probability close to 1 for a negative instance

Training and cost function

- The cost function over the whole training set

$$L = -\frac{1}{M}\sum_{i=1}^{M}\left[y^{(i)}\ln(\hat{y}^{(i)}) + (1-y^{(i)})\ln(1-\hat{y}^{(i)})\right]$$

- Cost function is convex, so iterative Gradient Descent (or any other optimization algorithm) is guaranteed to find the global minimum (if the learning rate is not too large and you wait long enough)

Gradient Descent of Log Loss

- $L(y, \hat{y}) = -y \ln \hat{y} - (1 - y) \ln(1 - \hat{y})$

- $\hat{y} = \sigma(z)$
- $\hat{y} = \sigma(\boldsymbol{w}^T \boldsymbol{x})$

- $\acute{\sigma}(z) = \ \sigma(z)(1 - \sigma(z))$
- To find the gradient of the log loss with respect to the weights $\boldsymbol{w}$
- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w}$

Gradient Descent of Log Loss

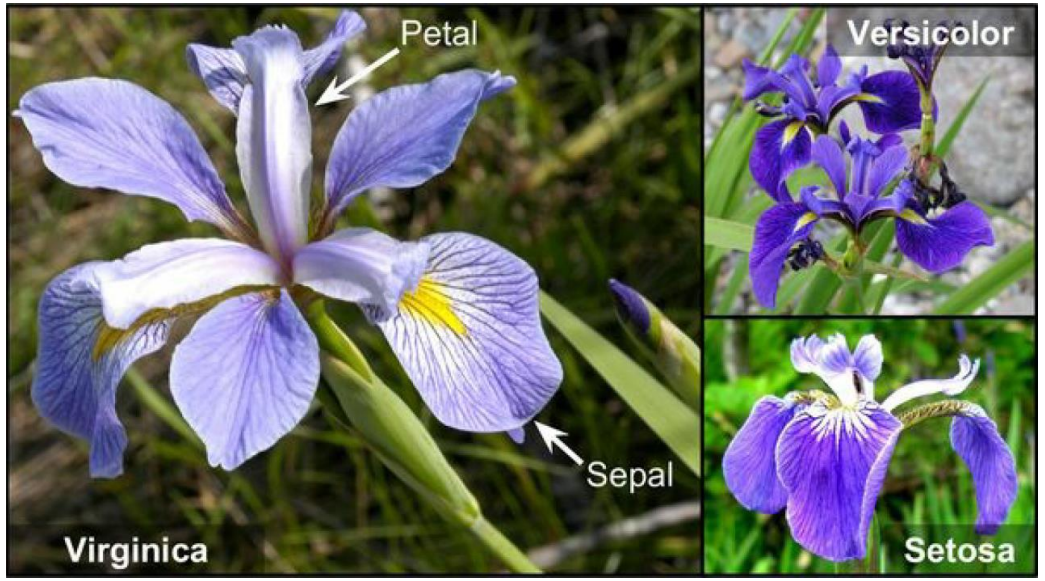
- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w} = \frac{-(y-\hat{y})}{\hat{y}(1-\hat{y})} \hat{y}(1-\hat{y})\, \boldsymbol{x}$

- $\frac{\partial L}{\partial w} = -(y-\hat{y})\boldsymbol{x}$

- To update the weights $\boldsymbol{w}$ to minimize the log loss, we adjust $\boldsymbol{w}$ in the opposite direction of the gradient of the loss:
- $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta(y-\hat{y})\boldsymbol{x}$

IRIS dataset

- Dataset that contains the **sepal** (الكأس) and **petal** (التويج) length and width of 150 iris flowers of three different species: Iris-**Setosa**, Iris-**Versicolor**, and Iris-**Virginica**
  - 4 numeric attributes
  - 50 instance per each class
  - One class is linearly separable from the other two



Pima Indians Diabetes Database

- The datasets consists of several medical predictor variables and one target variable.
- Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Appendix 01 - Gradient Descent of Log Loss

- $L(y, \hat{y}) = -y \ln \hat{y} - (1 - y) \ln(1 - \hat{y})$

- $\hat{y} = \sigma(z)$
- $\hat{y} = \sigma(\boldsymbol{w}^T \boldsymbol{x})$

- $\acute{\sigma}(z) = \ \sigma(z)(1 - \sigma(z))$
- To find the gradient of the log loss with respect to the weights $\boldsymbol{w}$
- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w}$

Appendix 01 - Gradient Descent of Log Loss

- $L(y, \hat{y}) = -y \ln \hat{y} - (1 - y) \ln(1 - \hat{y})$
- $\hat{y} = \sigma(z)$
- $\hat{y} = \sigma(\boldsymbol{w}^T \boldsymbol{x})$
- $\acute{\sigma}(z) = \ \sigma(z)(1 - \sigma(z))$
- To find the gradient of the log loss with respect to the weights $\boldsymbol{w}$
- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w}$
- $\frac{\partial L}{\partial \hat{y}}$??

Appendix 01 - Gradient Descent of Log Loss

- $L(y, \hat{y}) = -y \ln \hat{y} - (1 - y) \ln(1 - \hat{y})$
- $\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}} = \frac{-y((1-\hat{y})+\hat{y}(1-y)}{\hat{y}(1-\hat{y})} = \frac{-(y-\hat{y})}{\hat{y}(1-\hat{y})}$

- $\frac{\partial \hat{y}}{\partial z}$???
- $\because \quad \hat{y} = \sigma(z)$
- $\frac{\partial \hat{y}}{\partial z} = \hat{y}(1 - \hat{y})$

- $\frac{\partial z}{\partial w}$???
- $z = \boldsymbol{w}^T \boldsymbol{x}$
- $\frac{\partial z}{\partial w} = \boldsymbol{x}$

- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w} = \frac{-(y-\hat{y})}{\hat{y}(1-\hat{y})} \hat{y}(1 - \hat{y}) \boldsymbol{x}$

Appendix 01 - Gradient Descent of Log Loss

- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w} = \frac{-(y-\hat{y})}{\hat{y}(1-\hat{y})} \hat{y}(1-\hat{y})\, \boldsymbol{x}$

- $\frac{\partial L}{\partial w} = -(y-\hat{y})\boldsymbol{x}$

- To update the weights $\boldsymbol{w}$ to minimize the log loss, we adjust $\boldsymbol{w}$ in the opposite direction of the gradient of the loss:
- $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta(y-\hat{y})\boldsymbol{x}$

Appendix 02

- Prove That: $\acute{\sigma}(z) = \sigma(z)(1 - \sigma(z))$

- $\hat{y} = \sigma(\boldsymbol{w}^T \boldsymbol{x}) = \sigma(z)$ $\quad\quad$ $z = \boldsymbol{w}^T \boldsymbol{x}$
- $\sigma(z) = \frac{1}{1+e^{-z}} = (1 + e^{-z})^{-1}$

1. Differentiate $u = 1 + e^{-z}$ with respect to z:
   $\frac{du}{dz} = -e^{-z}$
2. Apply the chain rule to $\sigma(z) = u^{-1}$:
   $\frac{d\sigma}{dz} = \frac{d}{dz}(u^{-1}) = -u^{-2} \cdot \frac{du}{dz}$
   $\frac{d\sigma}{dz} = -(1 + e^{-z})^{-2} \cdot (-e^{-z})$
   $\frac{d\sigma}{dz} = \frac{e^{-z}}{(1+e^{-z})^2}$

Appendix

1. **Differentiate $u = 1 + e^{-z}$ with respect to z:**
   $\frac{du}{dz} = -e^{-z}$
2. **Apply the chain rule to $\sigma(z) = u^{-1}$:**
   $\frac{d\sigma}{dz} = \frac{d}{dz}(u^{-1}) = -u^{-2} \cdot \frac{du}{dz}$
   $\frac{d\sigma}{dz} = -(1 + e^{-z})^{-2} \cdot (-e^{-z})$
   $\frac{d\sigma}{dz} = \frac{e^{-z}}{(1+e^{-z})^2}$
3. **Simplify the expression:**
   To simplify, notice that $e^{-z} = \frac{1}{e^z}$ and rewrite it in terms of $\sigma(z)$:
   $\sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{e^z+1}$
   So:
   $e^{-z} = \frac{1}{1+e^z}$
   $\frac{e^{-z}}{(1+e^{-z})^2} = \frac{\frac{1}{e^z}}{\left(1+\frac{1}{e^z}\right)^2} = \frac{1}{e^z} \cdot \frac{1}{\left(\frac{e^z+1}{e^z}\right)^2} = \frac{1}{e^z+1} \cdot \frac{e^{2z}}{(e^z+1)^2}$
   $= \frac{e^z}{(e^z+1)^2}$

   Notice that:
   $\sigma(z) = \frac{e^z}{e^z+1}$
   So:
   $1 - \sigma(z) = 1 - \frac{e^z}{e^z+1} = \frac{e^z+1-e^z}{e^z+1} = \frac{1}{e^z+1}$

   Thus:
   $\frac{d\sigma}{dz} = \sigma(z) \cdot (1 - \sigma(z))$