NUS Fintech Society

Machine Learning

Logistic Regression

Logistic Regression

Recap

- Two types of learning
    - Supervised and Unsupervised Learning

- Topic of focus for training wing: Supervised Learning

- Two broad kinds of problems within supervised machine learning
    - Numerical predictions
    - Classification problems

- Numerical problems covered (to some extent) with Linear Regression.

This week...

- Today we look at classification problems

- Examples
  - Is an image a hotdog or not a hotdog https://www.youtube.com/watch?v=pqTntG1RXSY
  - On a more serious note, logistic regression can be used to identify if a tumor is cancerous or malignant.

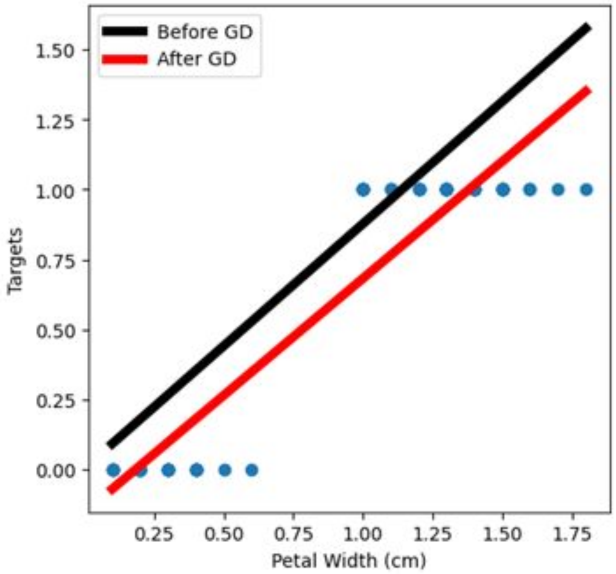
- Logistic Regression is more commonly used than Linear Regression

This week...

- Our focus will be logistic regression - an augmentation of what we learnt last week
- Logistic Regression is more commonly used than Linear Regression
- When you think about it, logistic regression provides direct outcomes.
- Most “Linear Regression” style questions can be reframed (better) in a way that transforms the problem into a classification one.

Logistic Regression

- Among the many things linear regression cannot do, it is especially bad at fitting data when the dependent variable is *discrete* and not *continuous*



Logistic Regression

- Examples
  1. Suppose we are a bank. We want to find which credit card transactions are legitimate and which ones are not.
  2. Suppose we are investors. We want to find out whether stock price of company X will go up or down tomorrow
  3. Suppose we are investors. We want to find out by how much will stock price of company X go up or down tomorrow
- Which among the three examples above is a good use case for logistic regression?
  - Example 1 can be a good use case
  - Example 2 can be a good use case
  - Example 3 is a terrible use case

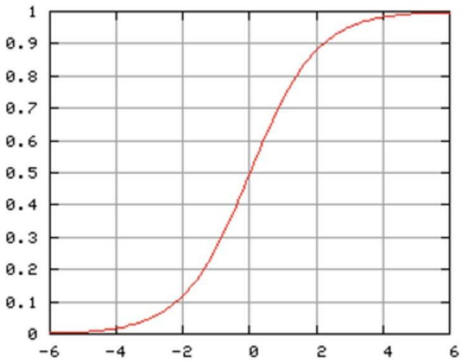
Logistic Regression

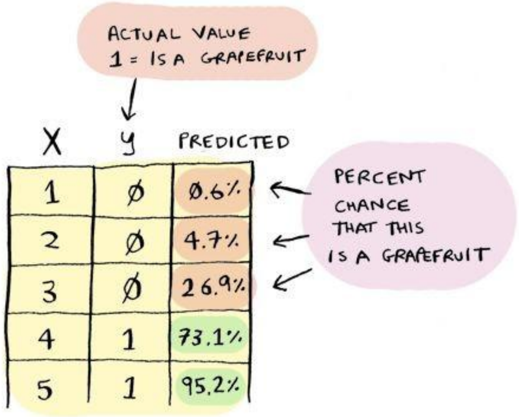
- How can classification be represented numerically?
- “Success” can be represented as 1, while “Failure” can be 0!
- So how do we go about this?

Logistic Regression

- A solution to this is to use a “squeezing” function that modulates the values to the range [0,1].
- Sigmoid function has output which lies between 0 and 1, so we get the added bonus of being able to think of it as *probabilities*

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





ACTUAL VALUE
1 = IS A GRAPEFRUIT

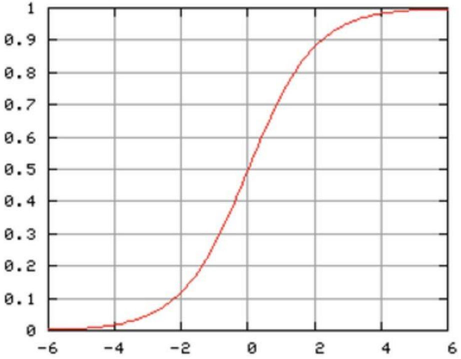
X	y	PREDICTED
1	Ø	Ø.6%
2	Ø	4.7%
3	Ø	26.9%
4	1	73.1%
5	1	95.2%

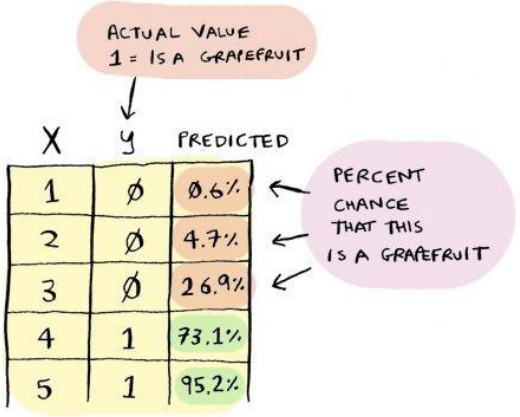
PERCENT CHANCE THAT THIS IS A GRAPEFRUIT

Logistic Regression

- When do we use this?
  - Binary outputs (Yes/No, Pass/Fail, etc.)
  - When probability information might be useful

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





ACTUAL VALUE
1 = IS A GRAPEFRUIT

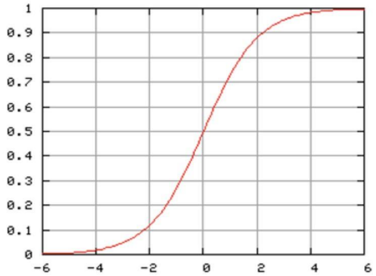
X	y	PREDICTED
1	Ø	Ø.6%
2	Ø	4.7%
3	Ø	26.9%
4	1	73.1%
5	1	95.2%

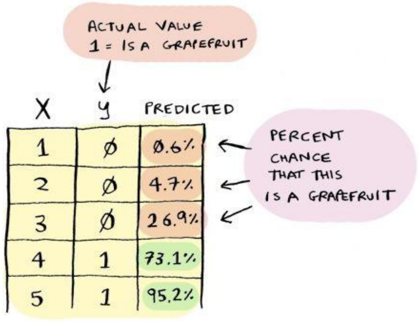
PERCENT CHANCE THAT THIS IS A GRAPEFRUIT

Logistic Regression

- Recall the Linear Regression function we defined earlier
  - $h_\Theta(x^i) = \Theta_0 + \Theta_1 \cdot x_1^i + \Theta_2 \cdot x_2^i + \Theta_3 \cdot x_3^i$
- In this case, we adapt this to become a probability
  - $h_\Theta(x^i) = \dfrac{1}{1 + e^{-(\Theta_0 + \Theta_1 \cdot x_1^i + \Theta_2 \cdot x_2^i + \Theta_3 \cdot x_3^i)}}$
  - $\Theta^T x^i = \Theta_0 + \Theta_1 \cdot x_1^i + \Theta_2 \cdot x_2^i + \Theta_3 \cdot x_3^i$
  - $h_\Theta(x^i) = \dfrac{1}{1 + e^{-\Theta^T x^i}}$

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





ACTUAL VALUE
1 = IS A GRAPEFRUIT

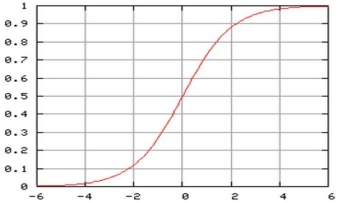
X	y	PREDICTED
1	Ø	Ø.6%
2	Ø	4.7%
3	Ø	26.9%
4	1	73.1%
5	1	95.2%

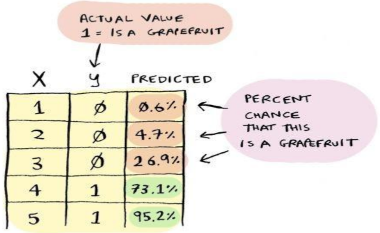
PERCENT CHANCE THAT THIS IS A GRAPEFRUIT

Logistic Regression

- Recall the cost function we defined earlier
  - $$J(\Theta) = \frac{1}{2n}\sum_{i=1}^{n}(y^i - (h_\Theta(x^i))^2$$
- In this case, we adapt this cost function to something that more accurately depicts probabilities rather than values.
- $$J(\Theta) = -\frac{1}{n}\sum_{i=1}^{n} y^i \cdot log(h_\Theta(x^i)) - (1-y) \cdot log(1 - h_\Theta(x^i))$$

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




ACTUAL VALUE
1 = IS A GRAPEFRUIT

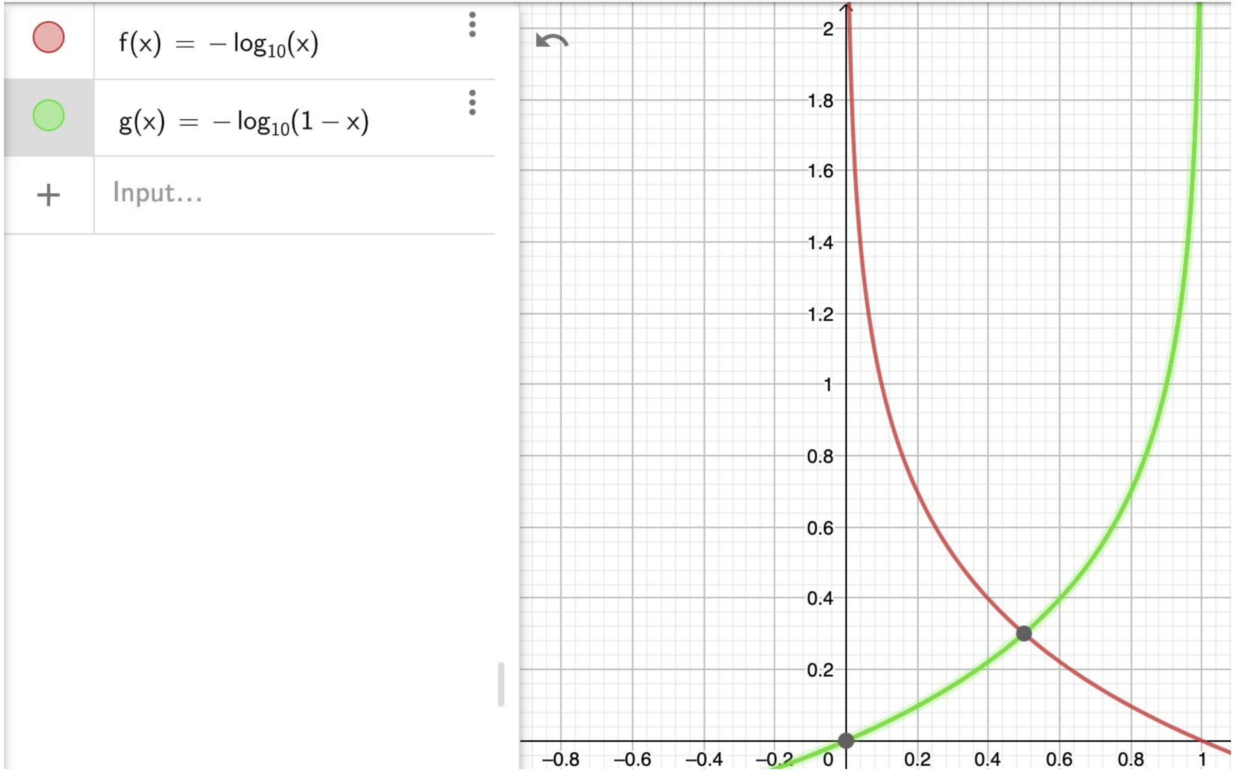
X	y	PREDICTED
1	Ø	Ø.6%
2	Ø	4.7%
3	Ø	26.9%
4	1	73.1%
5	1	95.2%

PERCENT CHANCE THAT THIS IS A GRAPEFRUIT

The Intuition

- Recall the cost function we defined earlier
  - $J(\Theta) = \frac{1}{2n}\sum_{i=1}^{n}(y^i - (h_\Theta(x^i))^2$
- In this case, we adapt this cost function to something that more accurately depicts probabilities rather than values.
- $J(\Theta) = -\frac{1}{n}\sum_{i=1}^{n} y^i \cdot log(h_\Theta(x^i)) - (1-y) \cdot log(1-h_\Theta(x^i))$
- So why y and 1-y? This to cover the two possible outcomes and the error that comes with them.

The Intuition



$f(x) = -\log_{10}(x)$

$g(x) = -\log_{10}(1 - x)$

An alternative...

- Instead of using a complicated function, just total up the number of incorrect classifications!

- If the value < 0.5, assume that the classification result is 0, and that it is 1 if the value > 0.5.

- Total up the number of misclassifications to get your error function!

Gradient Descent

- So how do we get to the correct parameters?
- We follow the same principle of gradient descent!
- Minimise the error with respect to each theta value.
- $\Theta_j = \Theta_j - \alpha \cdot \frac{1}{n}\sum_{i=1}^{n}((y^i - h_\Theta(x^i)) \cdot x_j^i)$
- Remember how Θ_0 is the bias term, and effectively x_0 is equal to 1 so the extra term does not really matter.

Pros and Cons

Advantages	Disadvantages
Output is in probabilities → might be useful for further statistical analysis	Possibility of over / under confidence
Probabilities can be interpreted as “confidence” of prediction, which can be understood easily	Can be tedious when using multiclass regression for a discrete, but a large quantity of outputs

What about cases with multiple possible outcomes?

- So far we’ve discussed situations with two possible outcomes.
- But what of situations with 3 or more possible outcomes?
- In problems like this (multiclass regression), we can make models for each of the outcomes and make a hypothesis based on this.

What about cases with multiple possible outcomes?

- Let’s say we are trying to identify a food item as either one of 4 foods
  - Hotdog
  - Pasta
  - Pizza
  - Baked Rice

- We can build 4 separate models. For instance, one of the models would be “Hotdog or NOT hotdog”, while the second would be “Baked Rice or NOT Baked Rice”, and so on and so forth.

- Find the model which has the greatest value, that’s your prediction!

Bias vs Variance

- In the previous lesson we discussed bias and variance.
  - High Bias: Under-fitted graph
  - High Variance: Over-fitted graph

- So what does that look like in logistic regression?

Diagnostics

Revisiting types of data

- Training data (60%++), Cross Validation data (20%), Testing Data (20%).
- What do we use cross validation data for?
  - For model selection

$$h_\Theta(x) = \Theta_0 + \Theta_1(x)$$

$$h_\Theta(x) = \Theta_0 + \Theta_1(x) + \Theta_2(x)^2$$

$$h_\Theta(x) = \Theta_0 + \Theta_1(x) + \Theta_2(x)^2 + \Theta_3(x)^3$$

$$\ldots$$

$$h_\Theta(x) = \Theta_0 + \Theta_1(x) + \Theta_2(x)^2 \ldots + \Theta_8(x)^8$$

- Imagine we have the following possible models - (Logistic regression has an extra sigmoid/other activation function wrapping around the terms).

Revisiting types of data

- Training data (60%++), Cross Validation data (20%), Testing Data (20%).

- Why can’t we use Cross-Validation data as a measure for a model’s accuracy?

- Thinking about a little deeper, you realise
  - There is an extra parameter for the degree of the model
  - This parameter has been fitted to the Cross-Validation data, and therefore renders it “tainted”.

- Therefore, we need the final step of using the testing data to represent what we believe the error of the model will be.

C.V. data for Regularization

- Training data (60%++), Cross Validation data (20%), Testing Data (20%).

- Use the same technique to identify which value of lambda you would like to have in your regularization parameter.

- Remember: $\sum_{j=1}^{n} \lambda \Theta_j^2$ term is added to the cost function to help minimise the coefficients and ensure the model is not overfitted to the training data.

- The greater the value of lambda, the more the minimisation of the parameters.

Scenario Time

- You are tasked with building a model for a certain task.

- In theory, it should work...but you’re getting bad results

- What do you do?
  - Get a larger training data set so that you can train your model better?
  - Improve the feature set of your model?
    - Reduce features
    - Increase features
  - Reduce or increase lambda in the regularization parameter?

Scenario Time

- Solutions: -
  - Getting more training examples: Fixes high variance
  - Trying smaller sets of features: Fixes high variance
  - Adding features: Fixes high bias
  - Adding polynomial features: Fixes high bias
  - Decreasing λ: Fixes high bias
  - Increasing λ: Fixes high variance.
- Identify whether a model has low/high bias/variance by measuring its performance using a cost function versus training set size graph: -

Summary

Summary

- Almost anything can be transformed into a classification problem.
- Logistic Regression wraps the Linear Regression model in a squeezing function that provides a value between 0 and 1.
- Multiclass regression is possible.
- Bias and Variance have to be taken into account.
- How bias and variance manifest themselves in models.

Summary

- Diagnosing issues in models.
- Understanding what cross-validation data is useful for.
- Better understanding of the regularisation parameter.

Food for thought

What if negative class was -1

- $Probability\ of\ an\ output\ given\ particular\ inputs = p(y|x)$

- Assume that the data is not time-series data.
  Therefore, (x,y) combinations in training data are independent.

- Our goal is to make sure that p(y|x) is high for all the training data instances.

- $p(y^1y^2...y^m|x^1x^2...x^m) = p(y^1|x^1) \cdot p(y^2|x^2) \cdot ... \cdot p(y^m|x^m)$

- $p(y^i|x^i) = g(y^i \cdot \Theta^T x^i)$

Thank You!