Introduction to Machine Learning (ML)
Closer look into Supervised Learning

SWINBURNE UNIVERSITY OF TECHNOLOGY

1

Acknowledgement

- These slides are based on the following page:
- https://savan77.github.io/2017-04-19-ml-part1/

- **Why?**
  - They provide the code that you can try out. Yaay!!

2

Supervised Learning

- Supervised learning problems can be divided into two types:
  - Regression, and
  - Classification

Linear Regression

- **Example**: Property prices have become a hot topic and we happen to have a dataset about houses:

area (square feet)	price (1k$s)
3456	600
2089	395
1416	232
...	...

(the data happen to be for houses in San Francisco – measuring the house area in square feet and house price in USD) – the example code uses the a similar dataset for the Boston area in the US

- Say, we try to predict house prices based on their areas. Then, **price** becomes our target variable (y), and **area** is our only input variable (x)

Linear Regression

- The Regression Problem is the problem of learning the target function f(x) that outputs the target value y for a given input x. In our example, price = f(area)

- If we believe that linear regression can be a good model for the target function f, f would take the form of a linear function: y = f(x) = a * x + b

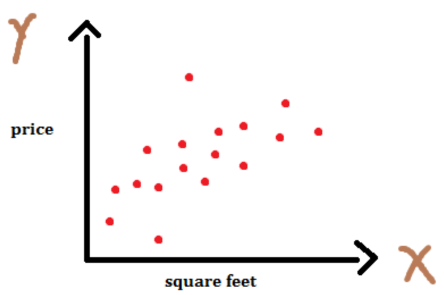
5

Linear Regression

- For our dataset:

area (square feet)	price (1k$s)
3456	600
2089	395
1416	232
…	…

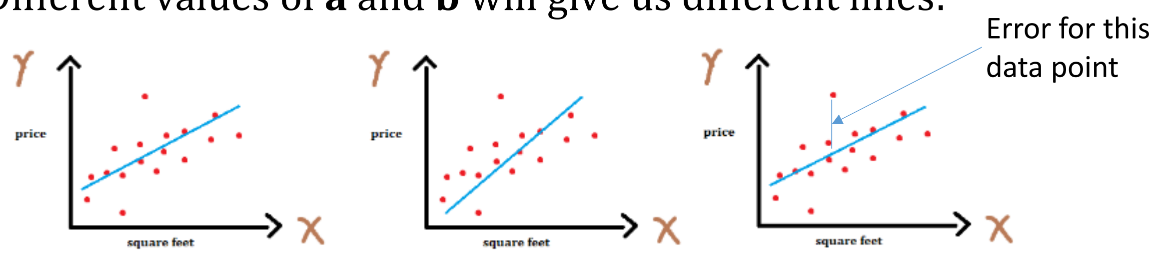
- If we visualise it:



6

Linear Regression

- If we choose a linear function to model the target function price = f(area) = **a** * area + **b**; the question is: What are the "good" values for **a** and **b**??
- Different values of **a** and **b** will give us different lines:



- The aim of ML algorithms is to search for the good values for the parameters **a** and **b** (**optimization** = minimizing the error)

7

Linear Regression

- Instead of writing $y = f(x) = \mathbf{a} * x + \mathbf{b}$, we can adopt the following standardized form: $y = f(x) = \mathbf{w_0} + \mathbf{w_1} * x$ (replacing the parameters **a** by $\mathbf{w_1}$ and **b** by $\mathbf{w_0}$)

  (this is why parameters are also called "**weights**")

- More generally, if instead of having just one input variable x, we have *m* input variables $x_1,\ldots, x_m$ then the linear regression target function will be written as:

  $$y = f(x_1,\ldots, x_m) = \mathbf{w_0} + \mathbf{w_1} * x_1 + \ldots + \mathbf{w_m} * x_m$$

8

Linear Regression

- Then, given a model represented by the parameters W = ($\mathbf{w_0}$, $\mathbf{w_1}$,… , $\mathbf{w}_{\boldsymbol{m}}$), we can calculate the error of this model using the cost function $\boldsymbol{J}$(W):

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

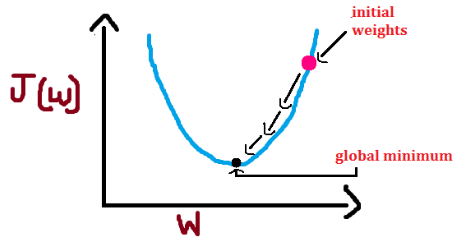
where:

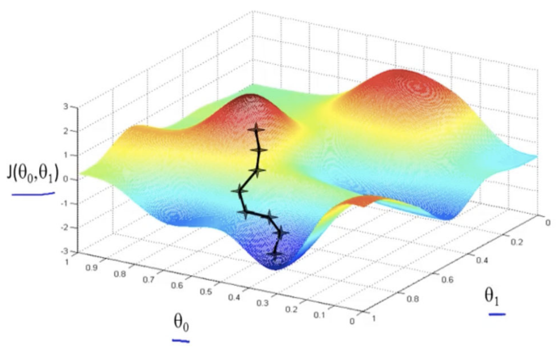
- n is the number of data points in the training dataset.
- For the i^{th} data point (x^i_1,…, x^i_m, y_i): $pred_i$ = f(x^i_1,…, x^i_m) is the model prediction.
- The above cost function $\boldsymbol{J}$ is the mean squared error.

9

Gradient descent

- Now that we have a way to measure the model's error in the cost function $\boldsymbol{J}$(W), we want to find the value of weights for which error is smallest (**minimizing** cost function).
- **Gradient Descent** is one of the most popular and widely used optimization algorithm.





10

More terminologies

- Cost function is also called "loss function"

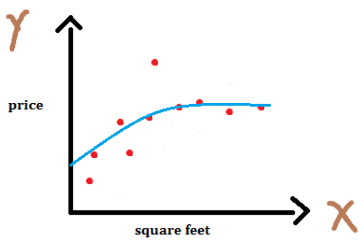
  Model loss ~ model error
- There are many variants of Gradient Descent (GD):
  - **Batch GD**: cost function is calculated on the entire training data (n = size of dataset) → slow
  - **Stochastic Gradient Descent (SGD)**: cost function is calculated on each training example ($n = 1$)
  - **Mini-batch gradient descent**: cost function is calculated on every mini-batch of k training examples (k < size of dataset)
- For more info: https://ruder.io/optimizing-gradient-descent/index.html#gradientdescentvariants

11

More terminologies

- What if a straight line (or hyperplane in the case of multiple input variables) does not fit the data well?
- Linear regression will still do the job by introducing a new feature $x_{new} = x^2$
- The target function will then look like this:

$$y = f(x) = \mathbf{w_0} + \mathbf{w_1} * x + \mathbf{w_2} * x^2$$

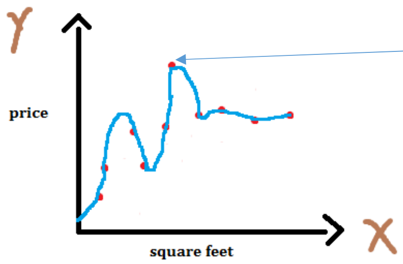


- And of course, we can have other features for x^3, x^4, etc.

12

Overfitting

- If we are overzealous in fitting the training data (e.g. by introducing x^3, x^4, etc.) we may end up overfitting the data:



This data point is perhaps anomaly
→ NOISE !

- The above model is unlikely to perform well on unseen data, especially around the anomaly data points.
- But, ***the aim of ML models is to perform well on unseen data***.

13

Underfitting

- The opposite of **overfitting** is **underfitting:**
  - Typically that is when the model is unable to capture the relationship between the input variables and the target variables accurately, generating a high error rate on both the training set and unseen data.
  - Perhaps we chose the wrong representation?
    - E.g. we should use decision tree (or SVM, or kNN, or Naïve Bayes) instead of linear regression??
- Perhaps there are no patterns that can generalise for the data at hand??

14

Overfitting – How to deal with noise?

- Instead of

$$J(W) = \frac{1}{n}\sum_{i=1}^{n}\left(w_0 + w_1 x_1^i + \cdots + w_m x_m^i - y_i\right)^2$$

- We will add the regularization term:

$$J(W) = \frac{1}{n}\left(\sum_{i=1}^{n}\left(w_0 + w_1 x_1^i + \cdots + w_m x_m^i - y_i\right)^2 + \lambda \sum_{j=1}^{m} w_j^2\right)$$

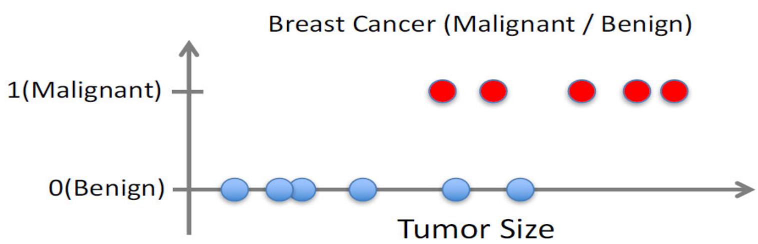
Regularization term L_2

When λ is sufficiently large, it will diminish the impact of the training examples

15

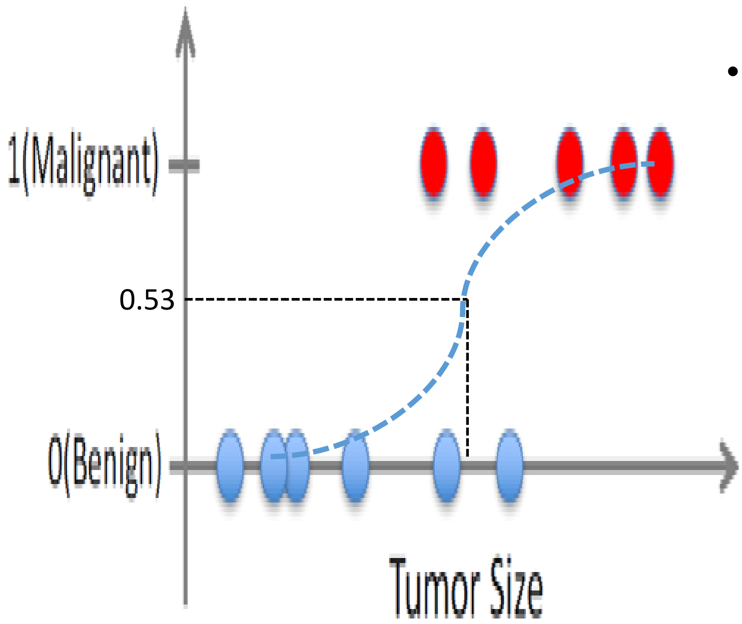
What about logistic regression?

- It is actually a classification technique! How??
- Remember this classification problem?



Based on example by Andrew Ng

16

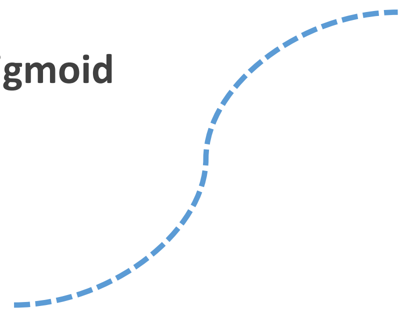


- One way to get the desired classification is by fitting a symmetric curve (dashed blue one). Then a **tumor size** can map to that curve and give us a value between 0 and 1 → the **probability that the tumor of that size is malignant**!

17

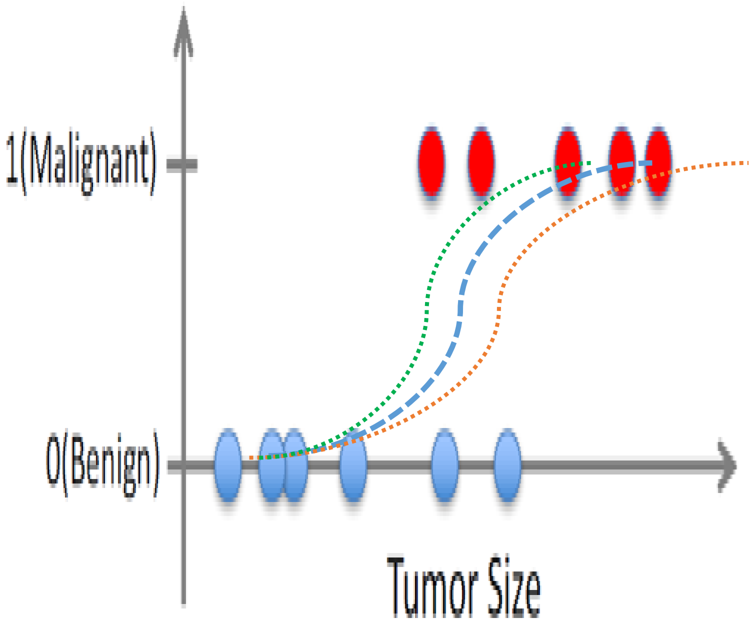
- A function with this shape is called **logistic function** or **sigmoid function:**

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



- As we don't want to return the probability (requiring some interpretation), what we can do is to map this probability outcome to the classification:
  - when the **prob ≥ 0.5**, the tumor will be *classified by our model* as **Malignant**; **else** it will be classified as **Benign**.

18



- Note that the blue model is not the only one. There are many. We still face the problem of which logistic regression model is the "right" one → **minimize the cost function**

19

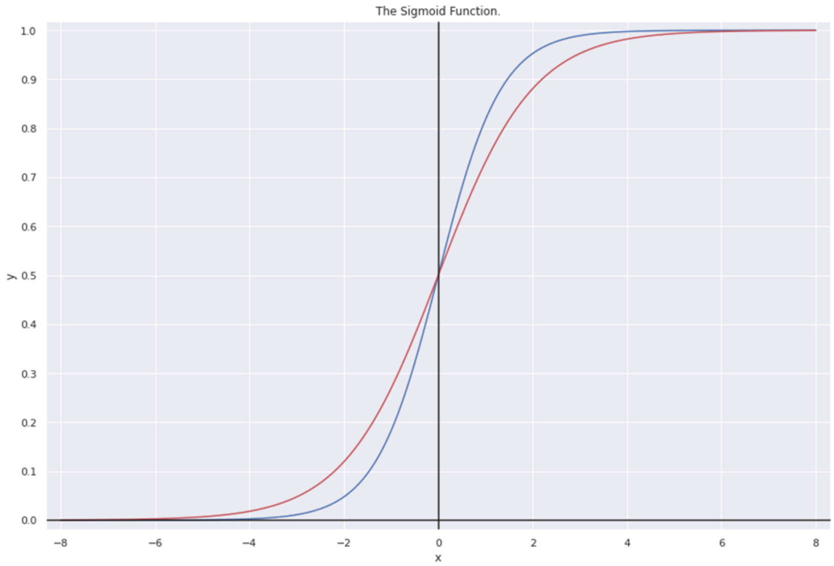
Logistic regression - mathematically

- $y \sim f(x) = \frac{1}{1+e^{-g(x)}} = \sigma(g(x))$
- (Well, more precisely $\mathrm{Prob}(y=1 \mid x) = \frac{1}{1+e^{-g(x)}}$)
- E.g., $\mathrm{Prob}(y = \mathrm{Malignant} \mid \mathit{size}) = \frac{1}{1+e^{-g(\mathit{size})}}$
- and $g(x) = \mathbf{w_0} + \mathbf{w_1} * x$
- Of course, for multivariate problem:

$$g_W(x_1,\ldots, x_m) = \mathbf{w_0} + \mathbf{w_1} * x_1 + \ldots + \mathbf{w}_m * x_m$$

20

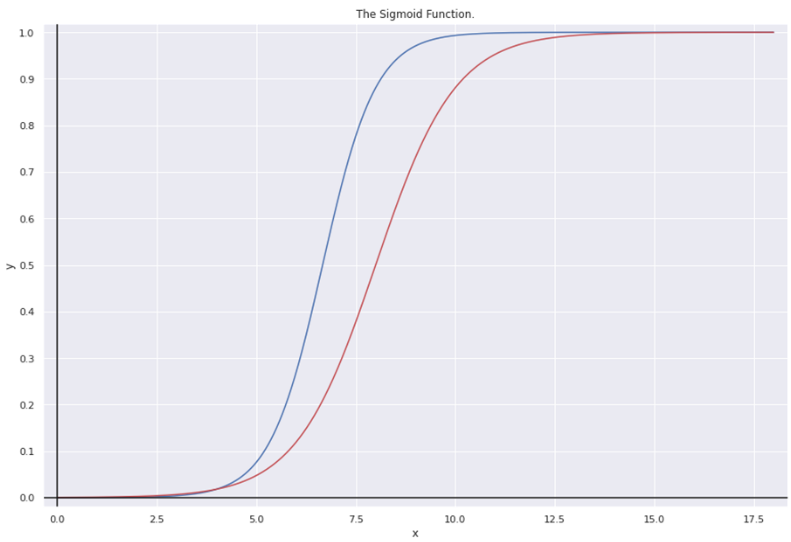
Logistic regression - examples



- The blue model is the function sigmoid(1.5*x), while the red line is the function sigmoid(x)

21

Logistic regression - examples



- The blue model is the function sigmoid(-10+1.5*x), while the red line is the function sigmoid(-8 + x)

22

Logistic regression – more maths

- So, what would a cost function look like for Log.Reg?

$$J(W) = -\frac{1}{n}\sum_{i=1}^{n}(y_i log\sigma(g(x^i)) + (1-y_i)\log(1-\sigma(g(x^i)))))$$

  - Note the minus (-) sign at the beginning
  - The expression after the sum is called the (log) likelihood of the parameters W (i.e., how likely does the model (parameterized by W) correctly classify the target variable y based on the input vars x)
  - Method: **Maximum Likelihood Estimation (MLE).** The mathematical intuition of the above can be found in Andrew Ng's video: https://www.youtube.com/watch?v=SHEPb1JHw5o

23

Summary

- In this mini lecture, we looked at the following:
  - The regression problem with the **linear regression** method (even though it's "linear", the model does NOT have to be a straight line; the **feature selection** problem)
  - The **cost** (or, loss) **function**
  - **Gradient Descent** (several variants)
  - Overfitting, Underfitting, Regularization
  - The classification problem with **logistic regression**
    - Sigmoid function, Maximum Likelihood Estimation

24