1

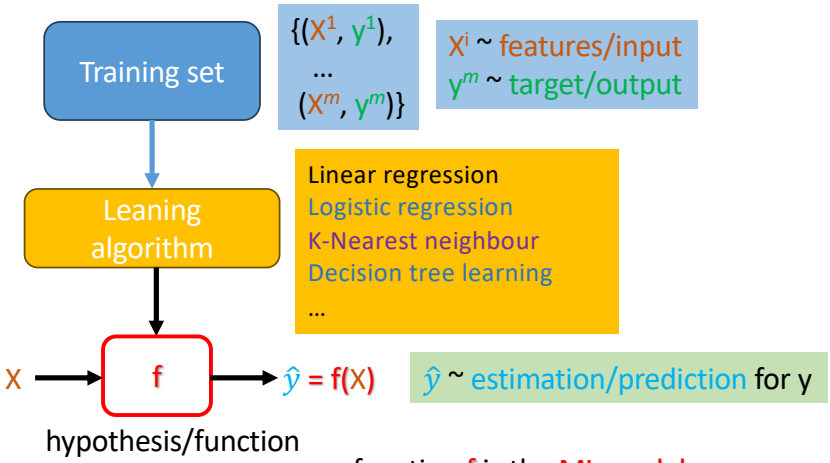
Overview

- Linear Regression
  - Cost function
- Logistic Regression
- Gaussian Discriminant Analysis (GDA)
- Naïve Bayes

2

Recap – From last week

Supervised learning:

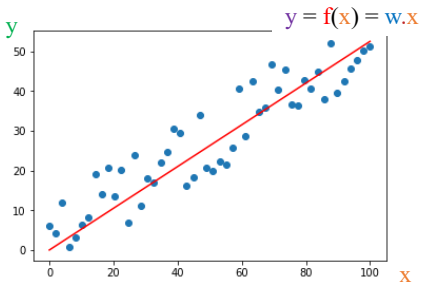


function f is the ML model produced by the learning algorithm

3

Cost function

Simple supervised learning algorithm: Linear regression (**one variable**)



w ~ the slope of function f
b ~ the bias (y-intercept of function f)

Question: What w and b should we have to "best fit" the training set?

"best" ~ optimization. More specifically, we'll **minimize** the "errors"

4

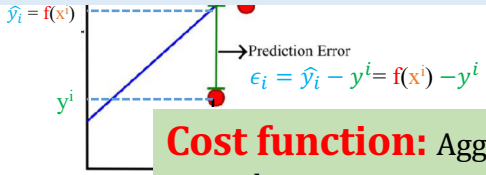
Cost function

Simple supervised learning algorithm: Linear regression (one variable)

Question: What w and b should we have to "best fit" the training set?

"best" ~ optimization. More specifically, we'll **minimize** the "errors"

That is, finding w and b so that $J_{w,b}$ is minimized.



example (x^i, y^i)

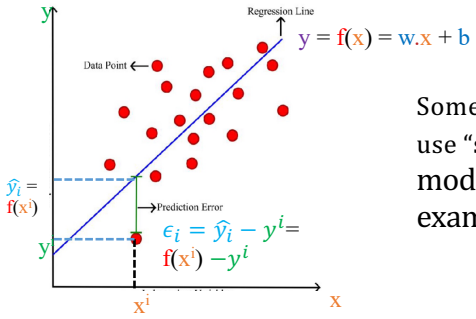
Cost function: Aggregating losses from all examples, e.g., **mean squared error**, **MSE**

$$J_{w,b} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y^i)^2$$

5

Cost function

Simple supervised learning algorithm: Linear regression (one variable)



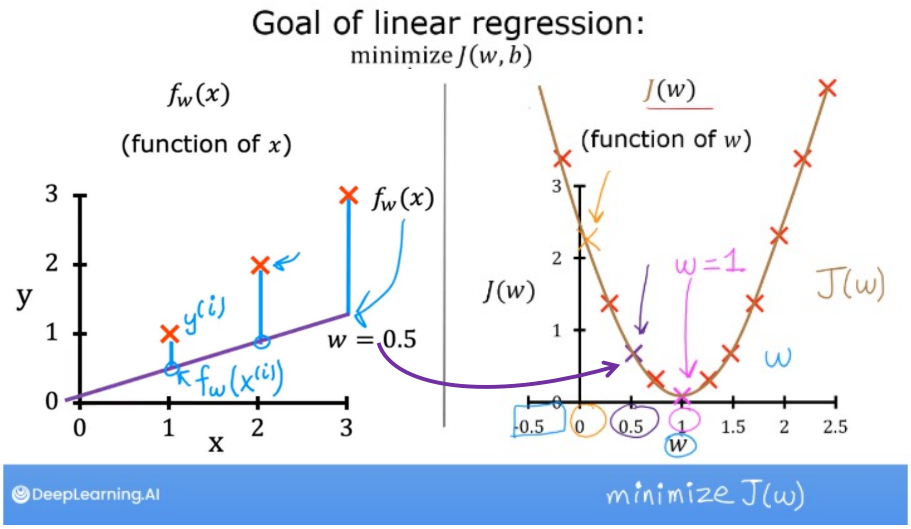
Sometimes, ϵ_i can be negative, use "**squared error**" ϵ_i^2 model f's loss from the example (x^i, y^i)

When we "**square**" the **error**, we may risk skewing the model f toward an outlier. Thus, another cost function is **Mean Absolute Error** (**MAE**). Yet, another one is **Root Mean Squared Error** (**RMSE**).

6

Minimizing cost function

Let's forget about b for a moment; i.e., assume b=0.

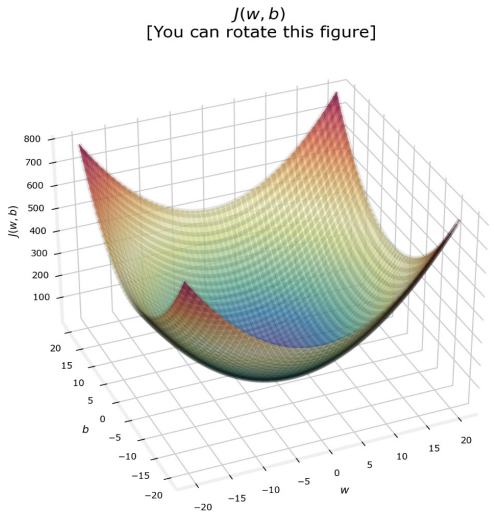


Source: **Andrew Ng/Stanford Online**

7

Minimizing cost function

More generally, this image shows a visualization of the cost function $J_{w,b}$ for a model f that predicts the *house price* y based on one single input feature *house size* x.



Source: **Andrew Ng/Stanford Online**

8

Minimizing cost function – Gradient descent

Cost function $J(w, b)$

Want: $\min_{w,b} J(w, b)$

Source: **Andrew Ng/Stanford Online**

Outline:

- Start with some w, b
- Repeatedly change w, b to reduce $J(w, b)$
- Until we settle at (or near) **a minimum** of $J(w, b)$

9

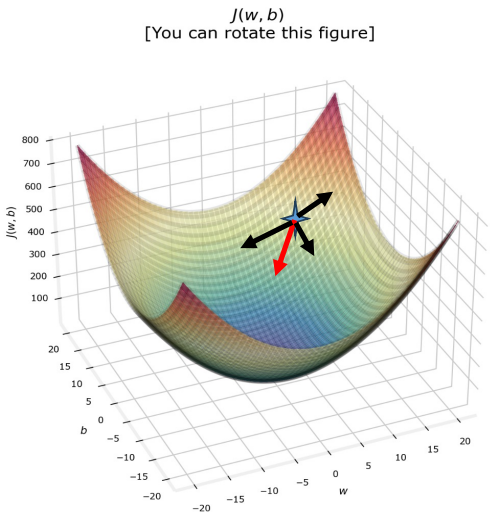
Minimizing cost function – Gradient descent

Cost function $J(w, b)$

Want: $\min_{w,b} J(w, b)$

Source: **Andrew Ng/Stanford Online**

Those "best" parameters can be computed using the gradient:
$(\frac{\partial}{\partial w} J(w, b), \frac{\partial}{\partial b} J(w, b))$



ce $J(w, b)$

nimum

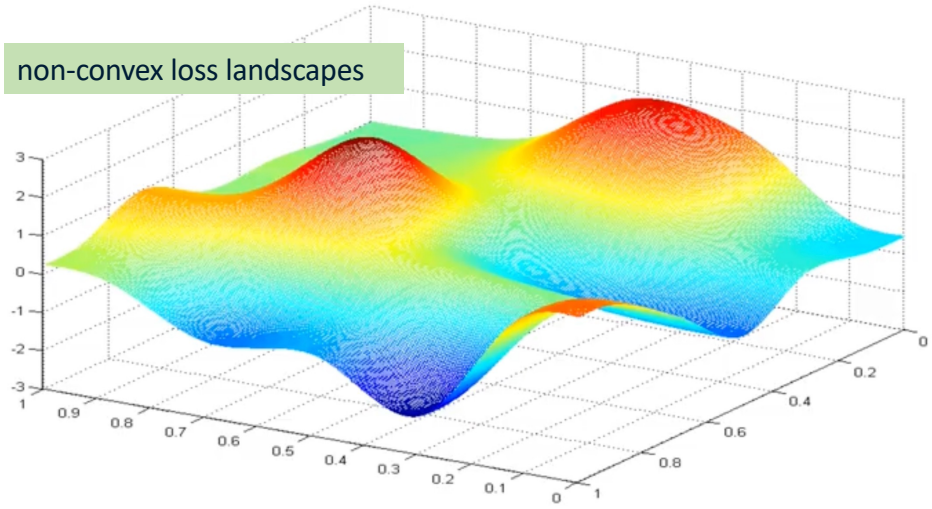
10

Minimizing cost function – Gradient descent

C

W

non-convex loss landscapes

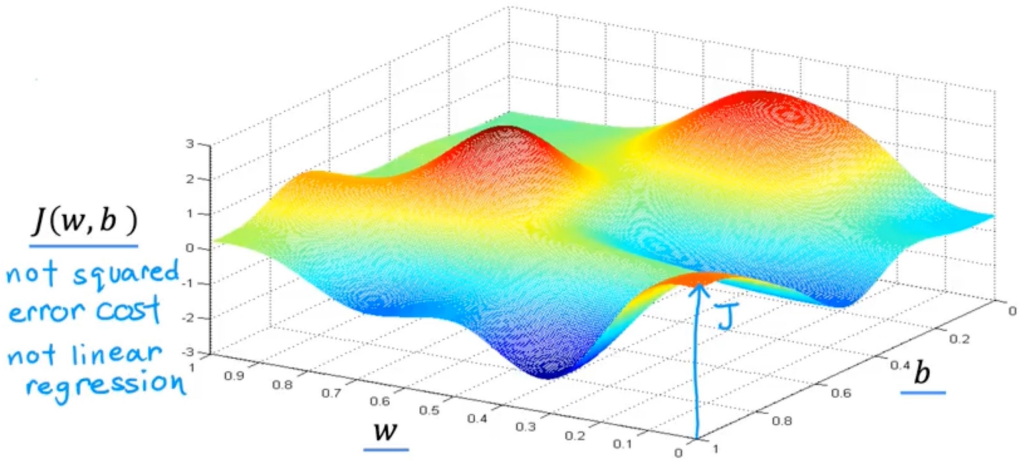


Source: **Andrew Ng/Stanford Online**

11

Minimizing cost function – Gradient descent

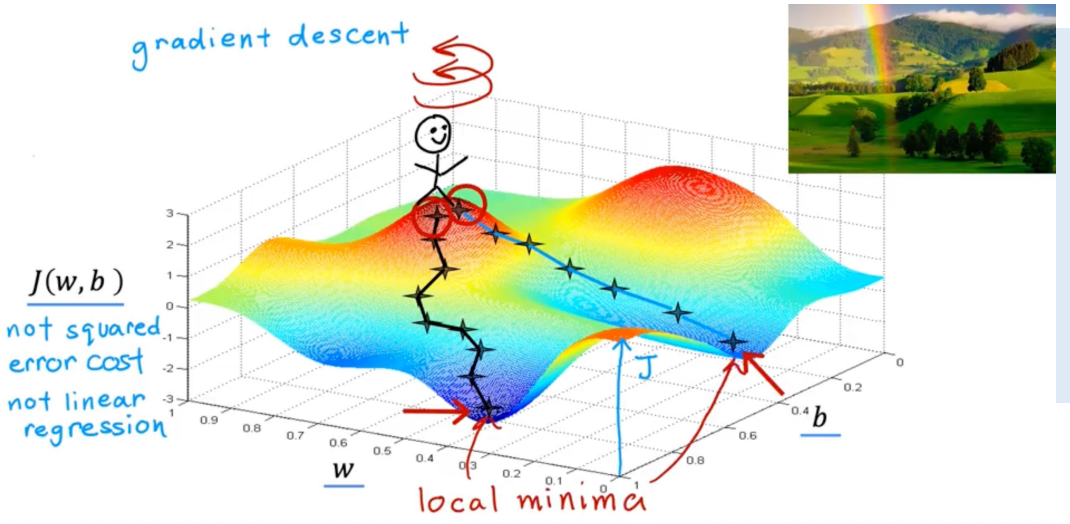
gradient descent



Source: **Andrew Ng/Stanford Online**

12

Minimizing cost function – Gradient descent



Source: **Andrew Ng/Stanford Online**

13

Gradient descent algorithm

Randomly assign some values to w and b

Repeat until convergence:

$$\{ w := w - \alpha \frac{\partial}{\partial w} J(w,b)$$

$$b := b - \alpha \frac{\partial}{\partial b} J(w,b) \}$$

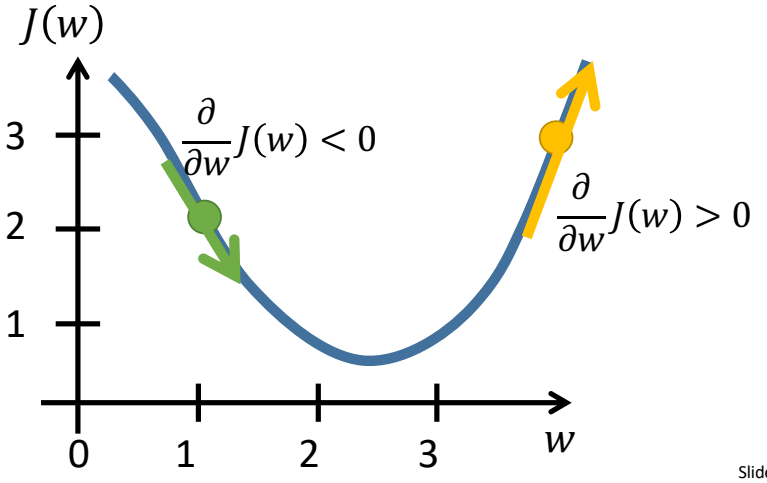
Updates to w and b (using a **learning rate** α) should be done **simultaneously**

The learning rate α controls how big of a step you take when updating the model's parameters, w and b.

14

Again, assuming b=0.

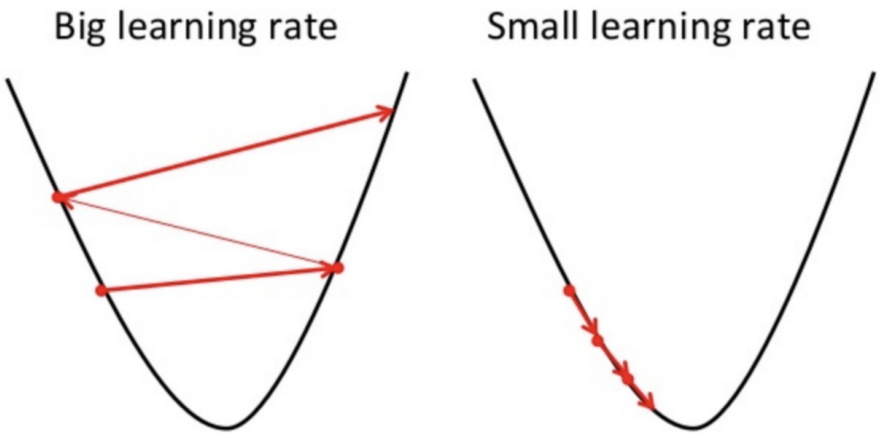
$$w := w - \alpha \frac{\partial}{\partial w} J(w)$$



Slide credit: Andrew Ng

15

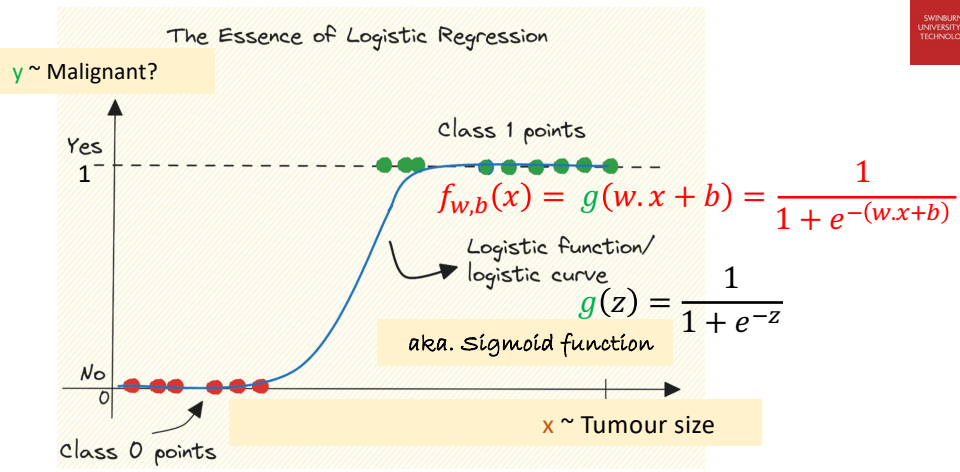
Learning rate α



16

What about classification?

Simple supervised learning algorithm: Logistic regression **(one variable)**

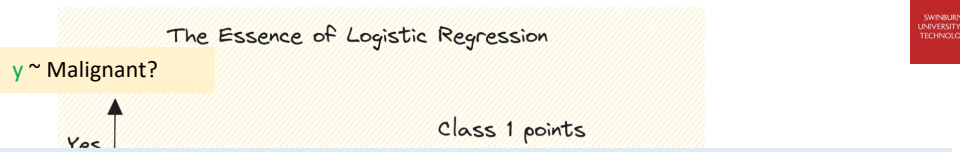


Interpretation: Given a tumour of size x, $f_{w,b}(x)$ is the **probability** that "it is malignant": $f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$

17

What about classification?

Simple supervised learning algorithm: Logistic regression **(one variable)**



But, if $f_{w,b}(x)$ returns a value (say, 0.6) for an input x, what should we conclude about the classification for y?

Use a threshold τ, say 0.5: if $f_{w,b}(x) \geq \tau$ then we return $y = 1$, otherwise we return $y = 0$.

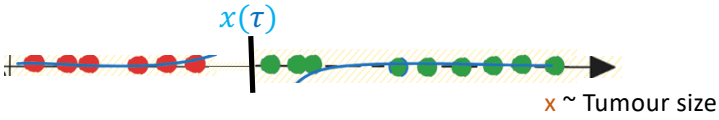
Interpretation: Given a tumour of size x, $f_{w,b}(x)$ is the **probability** that "it is malignant": $f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$

18

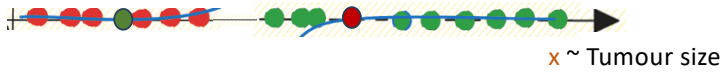
Decision boundary

Simple supervised learning algorithm: Logistic regression (**one variable**)

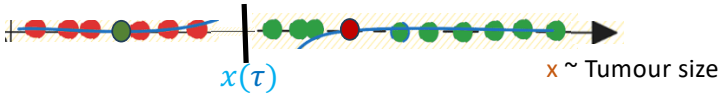
Let's assume that the classification of whether the prediction $\hat{y} = 1$ is based on the threshold $\tau = 0.5$ then it would form a decision boundary $x(\tau)$ that separate the tumour sizes:



Now, assume that the training dataset looks like this:



Then, what can we say about the following decision boundary $x(\tau)$?

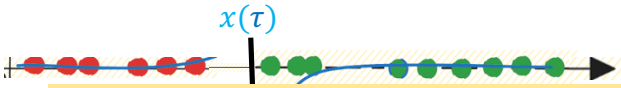


19

Decision boundary

Simple supervised learning algorithm: Logistic regression (**one variable**)

Let's assume that the classification of whether the prediction $\hat{y} = 1$ is based on the threshold $\tau = 0.5$ then it would form a decision boundary $x(\tau)$ that separate the tumour sizes:



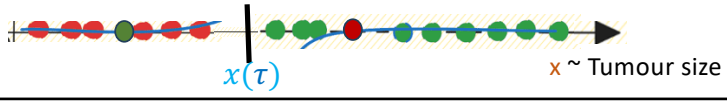
Note: By default, most people will choose the threshold 0.5 for **decision boundary**:

Now

Ther

Decision boundary

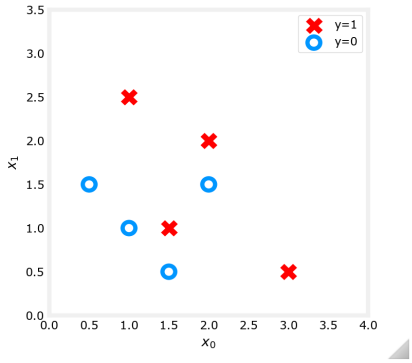
The decision boundary is the values of x for which $f(\mathbf{x}; \mathbf{w}) = \sigma(\mathbf{w}^\top \mathbf{x}) = 0.5$, i.e. $\mathbf{w}^\top \mathbf{x} = 0$.

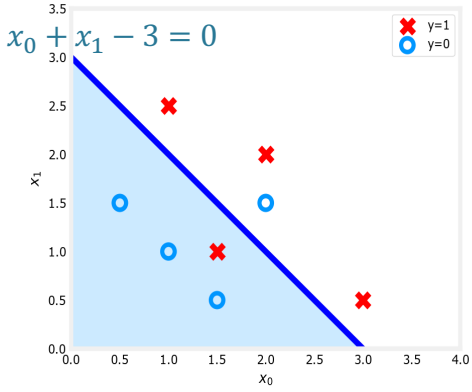


20

Decision boundary

Simple supervised learning algorithm: Logistic regression (**two variables**)

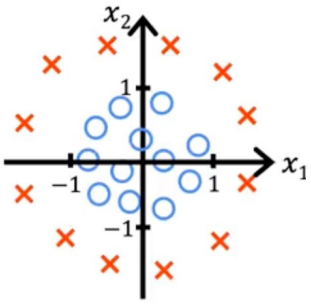




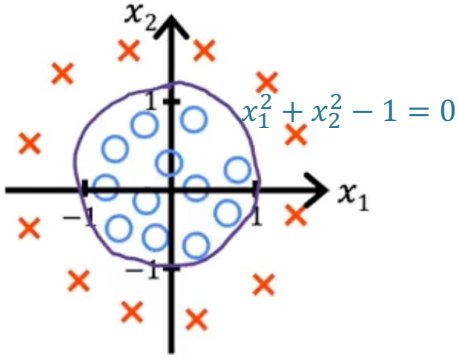
21

Non-linear decision boundary

Simple supervised learning algorithm: Logistic regression (**two variables**)



$$f_{\vec{w},b}(\vec{x}) = g(z) = g(\overbrace{w_1 x_1^2 + w_2 x_2^2 + b}^{z})$$



Source: Andrew Ng

22

Cost function: Squared error for logistic regression?

Squared Error Cost

cost

$$J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{\frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2}_{\text{loss}} \quad \text{where } \hat{y}^{(i)} = f_{\vec{w},b}(\vec{x}^{(i)})$$

average of training set

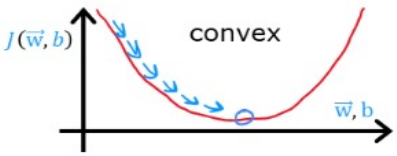
loss

$$L(\hat{y}^{(i)}, y^{(i)}) = \frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2$$

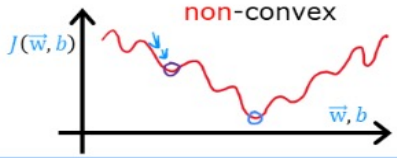
single training example

$$J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$$

linear regression $f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$



logistic regression $f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$



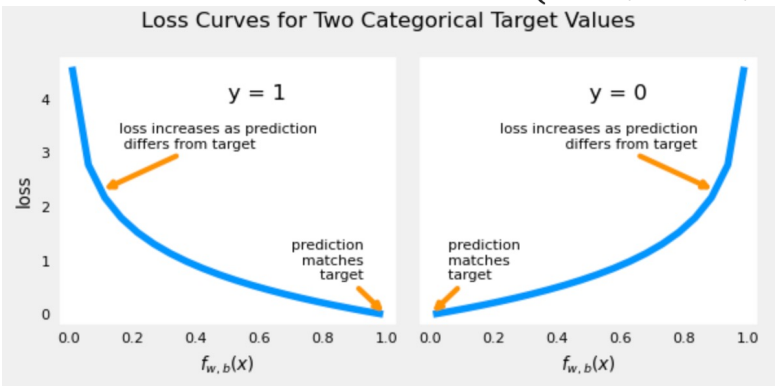
DeepLearning.AI Stanford ONLINE

Source: Andrew Ng

23

Cost function

$$loss(f_{\mathbf{w},b}(\mathbf{x}^{(i)}), y^{(i)}) = \begin{cases} -\log\left(f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$



Binary Cross Entropy loss

$$loss(f_{\mathbf{w},b}(\mathbf{x}^{(i)}), y^{(i)}) = (-y^{(i)} \log\left(f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right) \log\left(1 - f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right)$$

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} loss(f_{w,b}(x^{(i)}), y^{(i)})$$

Source: Andrew Ng

24

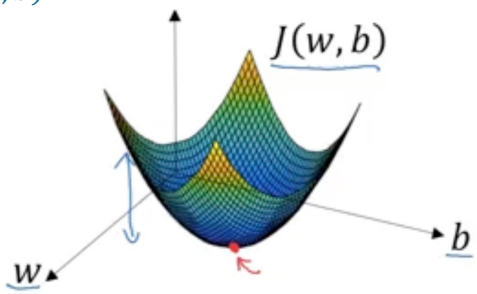
Minimizing cost function – Gradient descent

$$\hat{y} = f_{w,b}(x) = g(w.x + b) = \frac{1}{1 + e^{-(w.x+b)}}$$

Cost function:

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} loss(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} log\hat{y}^{(i)} + (1 - y^{(i)}) log(1 - \hat{y}^{(i)})]$$

We want: $\min_{w,b} J(w,b)$



Those "best" parameters can be computed using the gradient:
$(\frac{\partial}{\partial w} J(w,b), \frac{\partial}{\partial b} J(w,b))$

25

Gradient descent algorithm

Randomly assign some values to w and b

Repeat until convergence:

$$\{ w := w - \alpha \frac{\partial}{\partial w} J(w,b)$$

$$b := b - \alpha \frac{\partial}{\partial b} J(w,b) \}$$

Updates to w and b (using a **learning rate** α) should be done **simultaneously**

The learning rate α controls how big of a step you take when updating the model's parameters, w and b.

26

Maximum Likelihood Estimation (MLE)

MLE: A method regularly used by statisticians.

- vs cost function minimization (regularly used by ML practitioners) with **GD**

27

Maximum Likelihood Estimation (MLE)

MLE: A method regularly used by statisticians.

- vs cost function minimization (regularly used by ML practitioners) with **GD**

GD: Fits a model (i.e., parameters) to the data
MLE: Measures how well data can be explained by a model to determine the model that best explains the data (i.e., the model that maximizes the likelihood of the dataset)

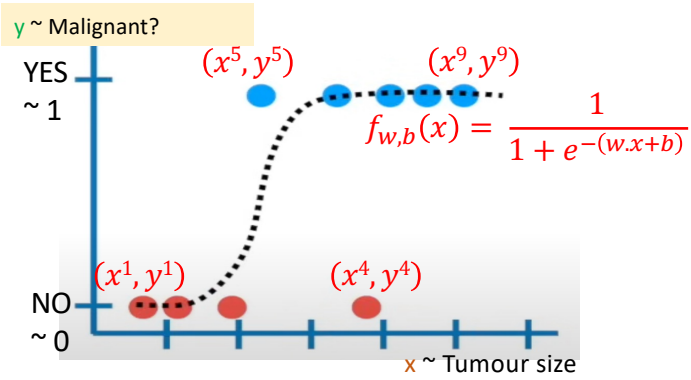
28

Maximum Likelihood Estimation (MLE)

MLE: A method regularly used by statisticians.

- vs cost function minimization (regularly used by ML practitioners) with **GD**

Let's consider the below dataset (9 data instances) and a model $f_{w,b}$



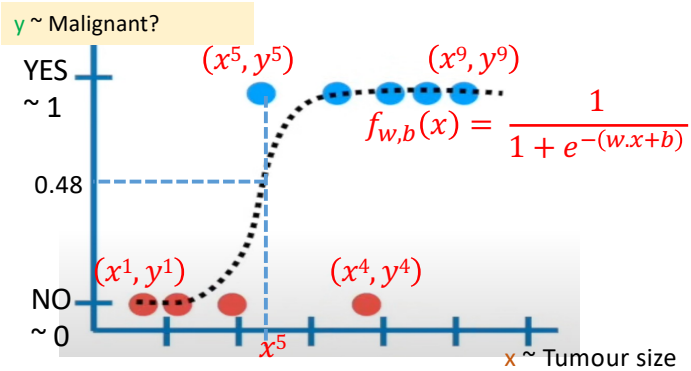
$$f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$$

29

Maximum Likelihood Estimation (MLE)

MLE: A method regularly used by statisticians.

The likelihood of five examples $\{(x^5, y^5), \ldots, (x^9, y^9)\}$ according to the model $f_{w,b}$ are the same as the probability $\boldsymbol{Prob}(y = 1|x; w, b)$.
For example, since $f_{w,b}(x^5)$ = 0.48, it is also the likelihood that the parameters w and b explain the data point (x^5, y^5).



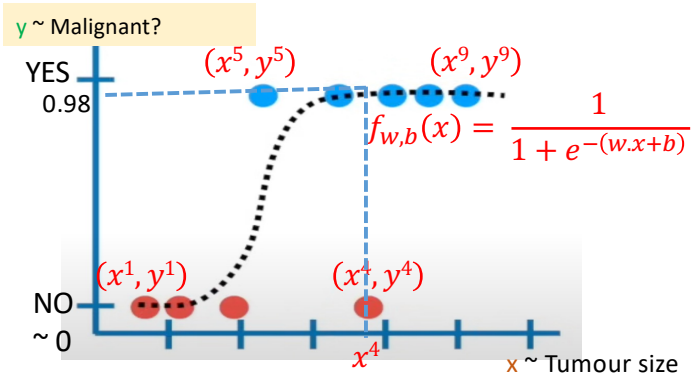
$$f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$$

30

Maximum Likelihood Estimation (MLE)

MLE: A method regularly used by statisticians.

The likelihood of four examples $\{(x^1, y^1), \ldots, (x^4, y^4)\}$ according to the model $f_{w,b}$ are the same as the probability
$\boldsymbol{Prob}(y = 0|x; w, b) = 1 - \boldsymbol{Prob}(y = 1|x; w, b) = 1 - f_{w,b}(x)$.
For example, since $f_{w,b}(x^4)$ = 0.98, the likelihood that the parameters w and b explain the data point $(x^4, y^4) = 1 - 0.98 = 0.02$.



$$f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$$

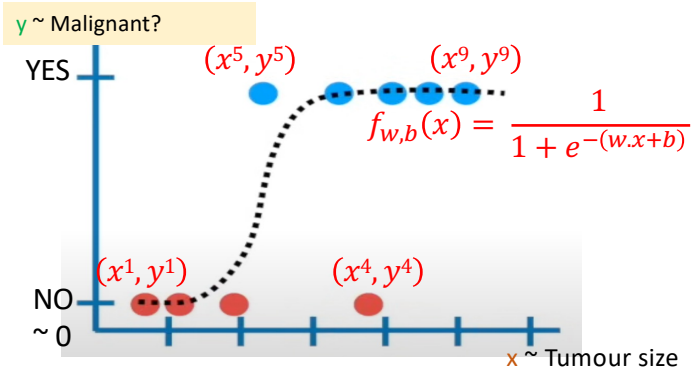
31

Maximum Likelihood Estimation (MLE)

MLE: A method regularly used by statisticians.

To conclude:
When $y^i = 1$, the likelihood $\mathcal{L}^i(w, b) = f_{w,b}(x^i)$;
When $y^i = 0$, the likelihood $\mathcal{L}^i(w, b) = 1 - f_{w,b}(x^i)$.

We can simplify it by:
$\mathcal{L}^i(w, b) = f_{w,b}(x^i)^{y^i}(1 - f_{w,b}(x^i))^{1-y^i}$



$$f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$$

32

Maximum Likelihood Estimation (MLE)

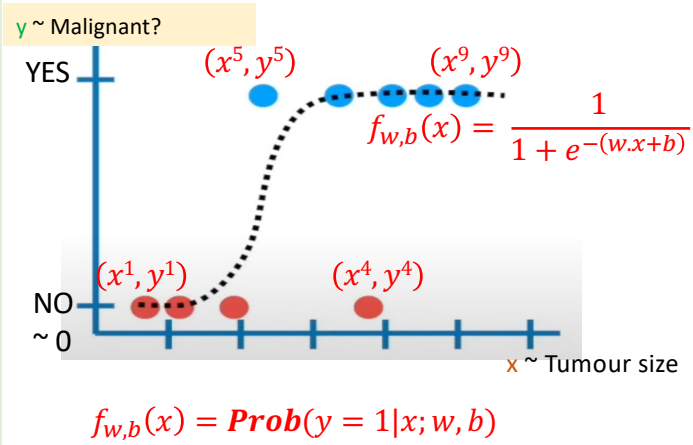
MLE: A method regularly used by statisticians.

We can now calculate the likelihood of our model across the entire dataset:

We can simplify it by:

$$\mathcal{L}(w,b) = \prod_{i=1}^{m} \mathcal{L}^i\,(w,b)$$

$$= \prod_{i=1}^{m} f_{w,b}\left(x^i\right)^{y^i}\left(1 - f_{w,b}\left(x^i\right)\right)^{1-y^i}$$



$$f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$$

33

Maximum Likelihood Estimation (MLE)

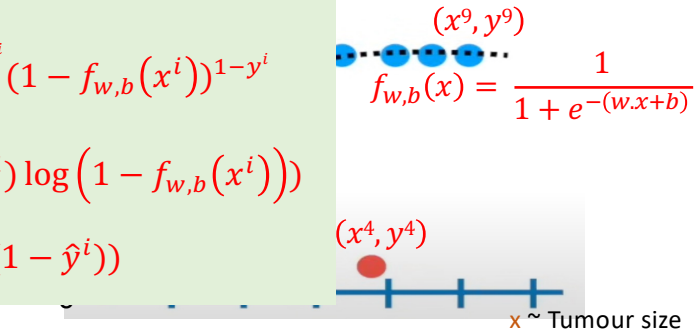
MLE: A method regularly used by statisticians.

We can maximize the Likelihood by maximizing the Log-Likelihood:

$$\log \mathcal{L}(w,b) = \log \prod_{i=1}^{m} f_{w,b}\left(x^i\right)^{y^i}\left(1 - f_{w,b}\left(x^i\right)\right)^{1-y^i}$$

$$= \sum_{i=1}^{m} \left(y^i \log f_{w,b}\left(x^i\right) + (1-y^i)\log\left(1 - f_{w,b}\left(x^i\right)\right)\right)$$

$$= \sum_{i=1}^{m} \left(\left(y^i \log \hat{y}^i + \left(1-y^i\right)\log(1-\hat{y}^i\right)\right)$$



$$f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$$

34

Maximum Likelihood Estimation (MLE)

Minimizing cost function – Gradient descent

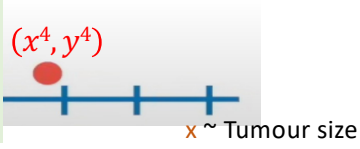
$$\hat{y} = f_{w,b}(x) = g(w.x + b) = \frac{1}{1 + e^{-(w.x+b)}}$$

Cost function:

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} loss(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} log\hat{y}^{(i)} + (1 - y^{(i)}) log(1 - \hat{y}^{(i)})]$$

$$= \sum_{i=1}^{m} (y^i \log f_{w,b}(x^i) + (1 - y^i) \log\left(1 - f_{w,b}(x^i)\right))$$

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



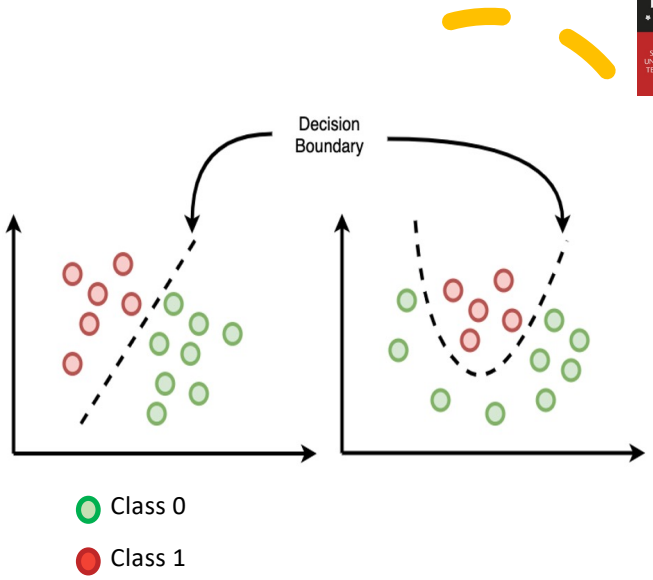
$$f_{w,b}(x) = \boldsymbol{Prob}(y = 1|x; w, b)$$

35

Discriminative vs. Generative Learning

- Logistic regression can be used to identify such decision boundaries using gradient descent/MLE

→ **Discriminative learning**



36

Generative learning?

- What about we look at each class (0 & 1) separately
- Can we generate a model of the input variables x when y = 0 (**Class 0**)?
- E.g., For a Benign tumor (y=0), what would be the typical tumor_size and clump_density?
- Similarly, model of x when y = 1??

37

But why would we want to do that?

- Isn't the purpose of ML is to get the input data of an unseen instance and try to classify this new instance??
- This is when Bayes' rule will be used:

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

- And of course:

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

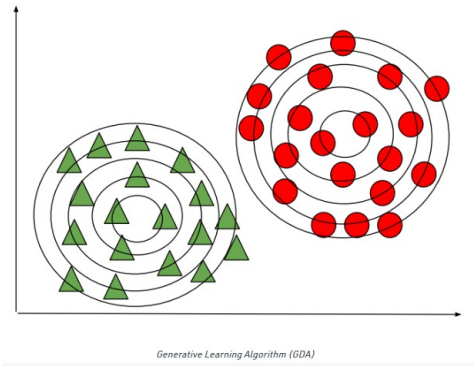
38

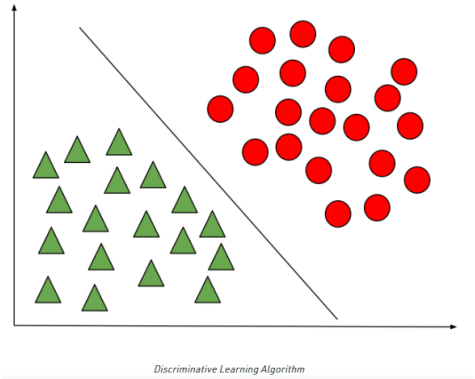
And, how?

- $P(y = 1)$ and $P(y = 0)$ are "innocent" enough: just the ratios of instances with a classification of 1 (and 0, respectively) *from the training set*!
  - You'll see later that it is another parameter that we might want to train!
- $P(x|y = 1)$ and $P(x|y = 0)$??
- Enter **Gaussian Discriminant Analysis (GDA)**, a Generative Learning Algorithm.

39

Let's visualise it: (source: www.geeksforgeeks.org/)

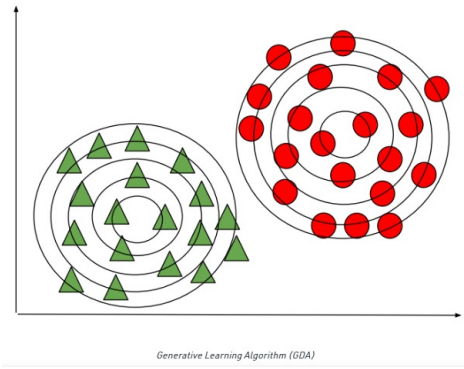




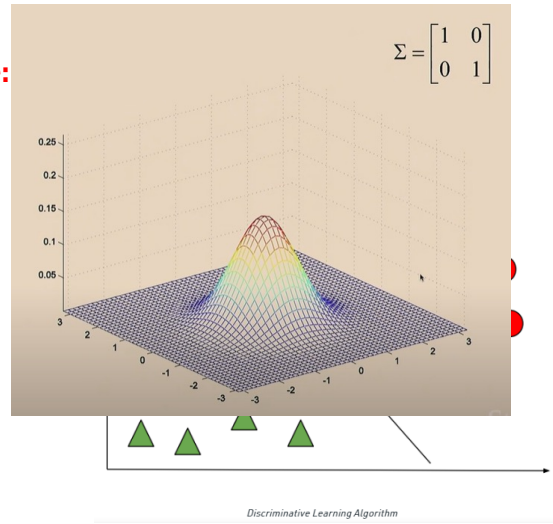
- The two sets of contours represent a generative learning model (e.g. GDA): the circle 1-contours and the triangles 0-contours.

40

Let's visualise it: (source:



Generative Learning Algorithm (GDA)

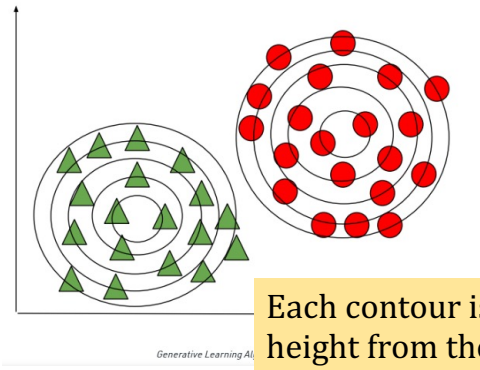


Discriminative Learning Algorithm

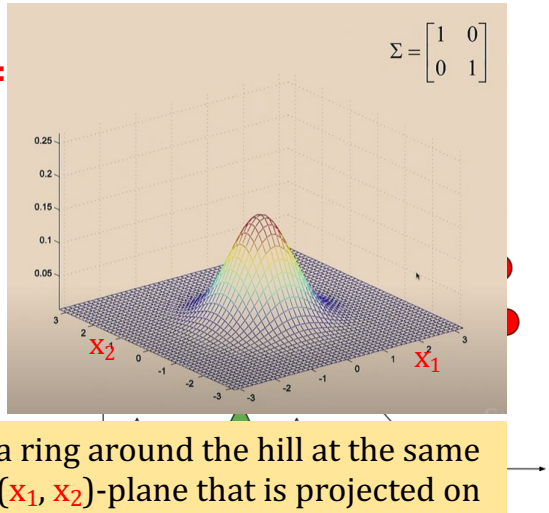
- The two sets of contours represent a generative learning model (e.g. GDA): the circle 1-contours and the triangles 0-contours.

41

Let's visualise it: (source:



Generative Learning Al

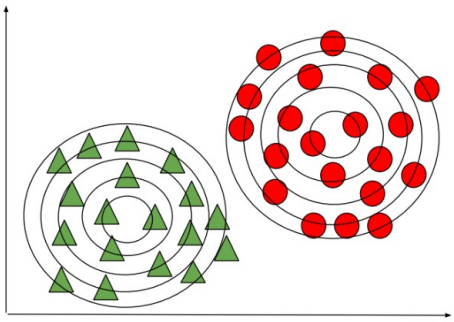


Each contour is a ring around the hill at the same height from the (x_1, x_2)-plane that is projected on the plane of the x_1 and x_2 axes.

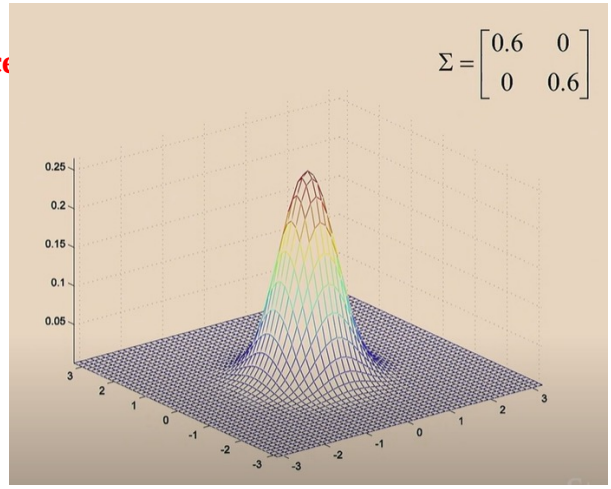
- The two sets of contours represent a generative learning model (e.g. GDA): the circle 1-contours and the triangles 0-contours.

42

Let's visualise it: (source



Generative Learning Algorithm (GDA)

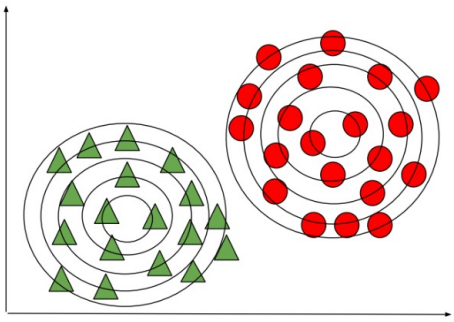


Discriminative Learning Algorithm

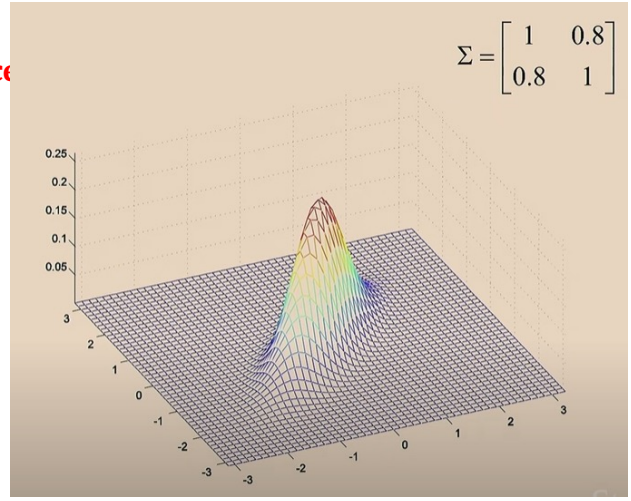
- The two sets of contours represent a generative learning model (e.g. GDA): the circle 1-contours and the triangles 0-contours.

43

Let's visualise it: (source



Generative Learning Algorithm (GDA)



Discriminative Learning Algorithm

- The two sets of contours represent a generative learning model (e.g. GDA): the circle 1-contours and the triangles 0-contours.

44

Want some maths? (source: Andrew Ng, Stanford)

$$p(y) = \phi^y(1-\phi)^{(1-y)}$$

$$p(x\,|\,y=0) = \frac{1}{(2\pi)^{n/2}|\sum|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu_0)^T \sum^{-1} (x-\mu_0))$$

$$p(x\,|\,y=1) = \frac{1}{(2\pi)^{n/2}|\sum|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu_1)^T \sum^{-1} (x-\mu_1))$$

- Simply speaking:
  - μ_0 is the centre of the 0-contours (mean of the triangle samples)
  - μ_1 is the centre of the 1-contours (mean of the circles samples)
  - Σ (capital sigma) is the co-variance matrix of the contours
  - ϕ = P(y=1)
    - Remember we promised that we would also determine P(y=1)?

45

Want some maths? (source: Andrew Ng, Stanford)

- Now, given a training set $\{x^{(i)}, y^{(i)}\}_{\text{i=1..m}}$
- The **joint likelihood** of the parameters:

$$\mathcal{L}(\phi, \mu_0, \mu_1, \Sigma) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) =$$
$$\prod_{i=1}^{m} p\left(x^{(i)}\middle|y^{(i)}; \phi, \mu_0, \mu_1, \Sigma\right) * \prod_{i=1}^{m} p(y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$

46

Want some maths? (source: Andrew Ng, Stanford)

- Now, given a training set $\{x^{(i)}, y^{(i)}\}_{\text{i=1..m}}$
- The **joint likelihood** of the parameters:

$$\mathcal{L}(\phi, \mu_0, \mu_1, \Sigma) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) = \prod_{i=1}^{m} p(x^{(i)}|y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) * \prod_{i=1}^{m} p(y^{(i)}; \phi, \mu_0, \mu_1, \Sigma)$$

- According to the principle of MLE we have to choose $\phi, \mu_0, \mu_1, \Sigma$ to maximize function $\mathcal{L}(\phi, \mu_0, \mu_1, \Sigma)$ above:
  - Instead of maximizing the Likelihood function we can maximize the Log-Likelihood Function:

$$\log(\mathcal{L}(\phi, \mu_0, \mu_1, \Sigma))$$

47

Want some maths? (source: Andrew Ng, Stanford)

- And the result is:

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\sum = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T \quad \text{where } k = 1\{y^{(i)} = 1\}$$

- (well, the Stanford students are actually required to prove mathematically that these equations are the results of maximizing the log likelihood function!!)

where 1{} is the indicator function 1{true} = 1 and 1{false} = 0.

48

Let's go back to our original question

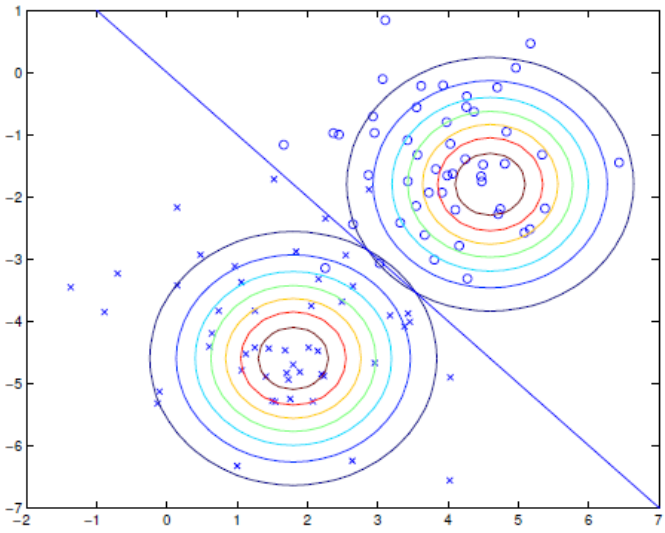
- Isn't the purpose of ML is to get the input data of an unseen instance and try to classify this new instance??

$$y_{new} = \underset{y}{\operatorname{argmax}}\, P(y|x_{new}) = \underset{y}{\operatorname{argmax}}\, \frac{P(x_{new}|y)P(y)}{P(x_{new})} = \underset{y}{\operatorname{argmax}}(P(x_{new}|y)P(y))$$

and, why do we ignore P(x_{new}) in the last expression?

49

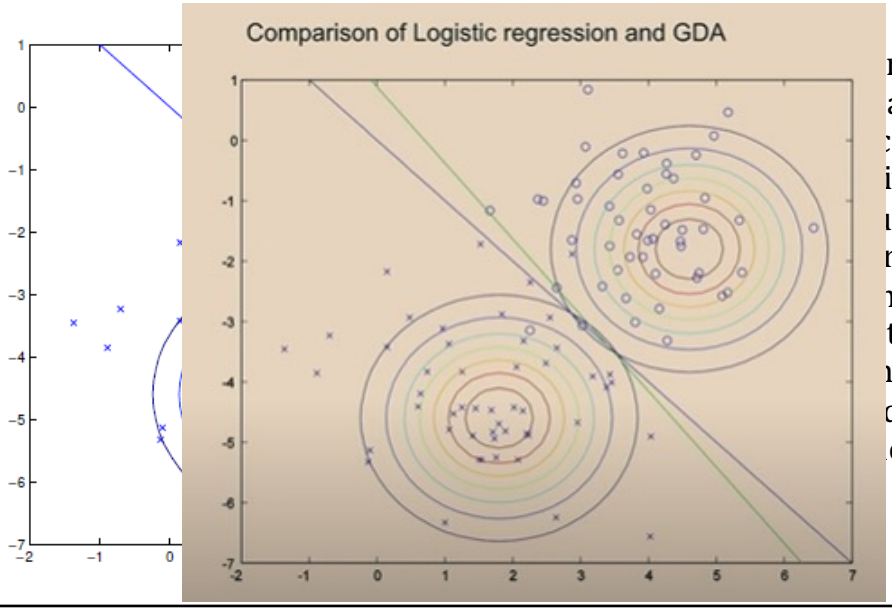
Let's visualise it: (source: Andrew Ng, Stanford)



- Based on the classification derived in the preceding slide, this generative model also give you a decision boundary (the blue line) which may be slightly different from the decision boundary obtained by logistic regression.

50

Let's visualise it: (source: Andrew Ng, Stanford)

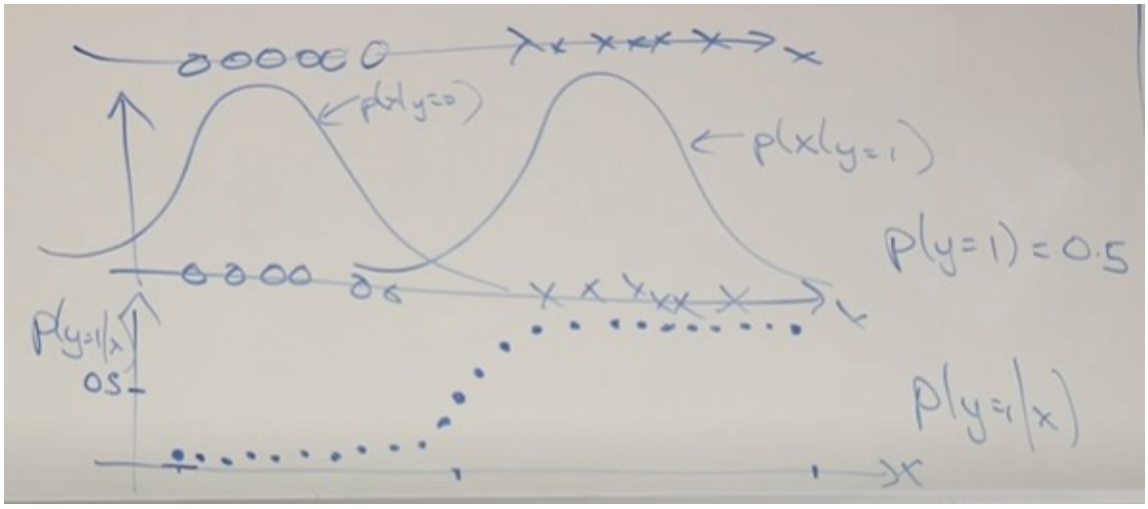


n the
ation derived in
eding slide, this
ive model also
a decision
ry (the blue line)
ay be slightly
t from the
boundary
d by logistic
on.

51

How do we compare GDA & Logistic Regression?

(source: Andrew Ng, Stanford)



52

Naïve Bayes algorithm

- It is also a **generative** learning algorithm
- Then, why can't we just stick with GDA???
- If instead of 3, 4 input variables x_1, x_2, x_3, x_4, we actually have 10,000 input variables!!
- A very common situation in natural language processing (NLP) tasks
  - E.g., How can email service providers (such as Gmail) classify whether an email is spam vs non-spam?
- Question 1: How do we represent our input data (emails) as numerical input variables to use the ML algorithms we saw??

53

Naïve Bayes algorithm

- Answer: Use *feature vector*.
- Given a 10,000-word English dictionary (e.g., the top-10,000 English words seen in emails)
- The feature vector:
- $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{10,000} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = 1\{\begin{bmatrix} a \\ aardvark \\ aardwolf \\ \vdots \\ zymurgy \end{bmatrix}\}$

where $1\{\}$ is again the indicator function.

54

Naïve Bayes algorithm

- This is when Naïve Bayes can be a useful algorithm by assuming that $x_1, x_2, x_3, \dots, x_{10,000}$ are **conditionally independent** given y
- That is,

$P(x_1, x_2, x_3, \dots, x_{10,000} | y)$

$= P(x_1|y)\, P(x_2 | y, x_1)\, P(x_3|y, x_1, x_2) \dots P(x_{10,000} | y, x_1, x_2, x_3, \dots, x_{9,999})$

$= (\text{C.I.}) = P(x_1|y)\, P(x_2 | y)\, P(x_3|y) \dots P(x_{10,000} | y) =$

$= \prod_{i=1}^{10,000} P(x_i|y)$

55

Naïve Bayes algorithm

- Now, let's talk about the parameters of this model:
- $\phi_{j|y=1} = P(x_j = 1|y = 1)$
- $\phi_{j|y=0} = P(x_j = 1|y = 0)$
- $\phi_y = P(y = 1)$

- Then, joint likelihood of the parameters:

$$\mathcal{L}(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}) = \prod_{i=1}^{n} p(x^{(i)}, y^{(i)}).$$

56

Naïve Bayes algorithm

- Maximizing this with respect to ϕ_y, $\phi_{j|y=0}$ and $\phi_{j|y=1}$ gives the maximum likelihood estimates:

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{n} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}}$$

$$\phi_y = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}{n}$$

57

Summary

- Linear regression is one of the most popular machine learning algorithms for the regression problem
  - A commonly used cost function is Mean Squared Error (MSE)
- For classification problems, there are two types of ML models:
  - Discriminative models (e.g., logistic regression, nearest neighbor, etc.)
  - Generative models (e.g., GDA, Naïve Bayes)
- Generative models are known to make stronger assumptions but are also more efficient to compute

58