Data X

Loss vs Risk

Data, Signals, and Systems

Ikhlaq Sidhu
Chief Scientist & Founding Director,
Sutardja Center for Entrepreneurship & Technology
IEOR Emerging Area Professor Award, UC Berkeley

Remember this Classification Example

Data

X	Y
1	0
2	1
5	0
4	0
7	1
10	1

Train

Logit function for Classification

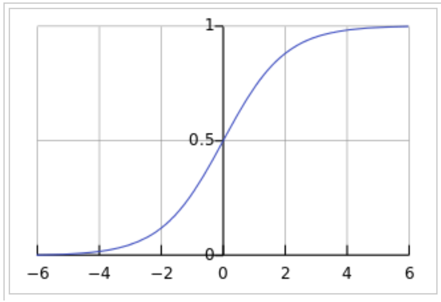
$$f(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

$t = w_0 + x_1 w_1$

Choose best parameter of w_0 , w_1

Model for Prediction

y



x

Illustration only
These numbers are not real

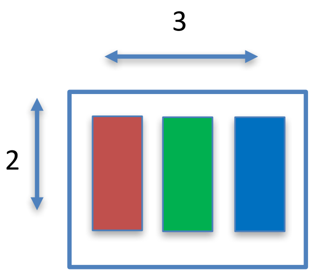
Classification Extended to Multiple Variables

Illustration only
These numbers are not real

Train
Data

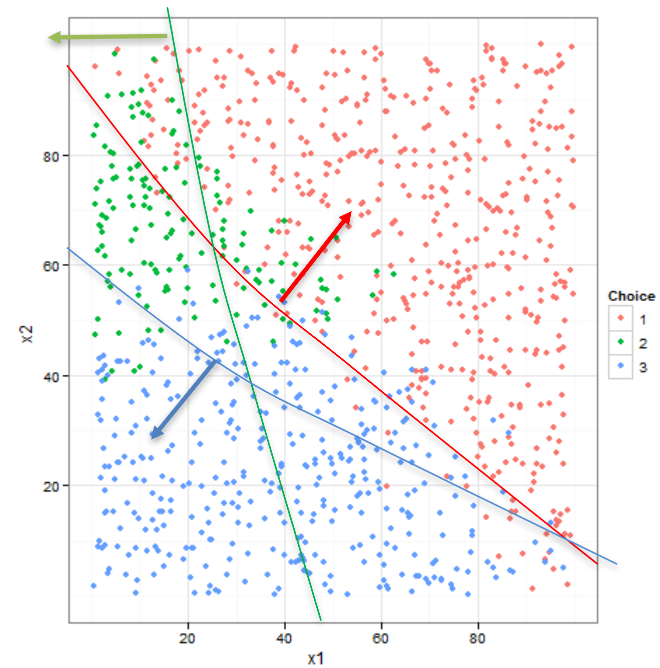
X
------
1 2
2 3
5 2
4 3
7 9
10 8

F(X,W)
W is (2 x 3)



Choose best
parameter of W

Y
-----
0 0 0
1 0 0
0 0 1
0 1 1
1 1 1
1 1 1

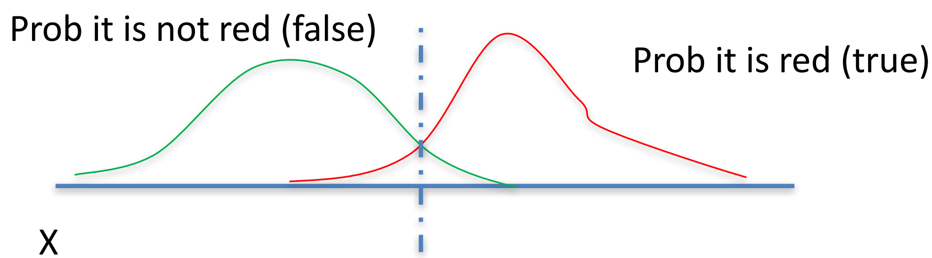


Model for Prediction

y1 boundary
Y2 boundary
Y3 boundary

$$f(x_i, W) = \frac{1}{1 + e^{-(w0\ + xi,1\ w1 + xi,2\ w2 \ldots)}}$$

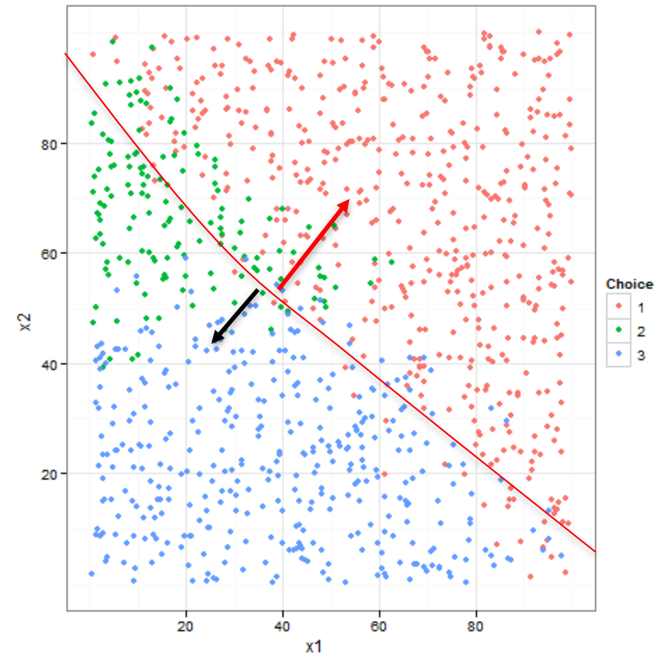
How Should We Decide a Classification Boundary?



One way is by likelihood that it is red vs green

P(green|x) > Prob (red|x) for all x
(this is maximum likelihood estimation)

Suppose we are detecting Red:
For now, lets look at Red = true
And Green as (not Red), ie False



For Reference: Maximum Likelihood Estimation

To use the method of maximum likelihood,[7] one first specifies the joint density function for all observations. For an independent and identically distributed sample, this joint density function is

$$f(x_1, x_2, \ldots, x_n \mid \theta) = f(x_1 \mid \theta) \times f(x_2 \mid \theta) \times \cdots \times f(x_n \mid \theta).$$

Now we look at this function from a different perspective by considering the observed values x_1, x_2, …, x_n to be fixed "parameters" of this function, whereas θ will be the function's variable and allowed to vary freely; this same function will be called the likelihood:

$$\mathcal{L}(\theta\,;\,x_1, \ldots, x_n) = f(x_1, x_2, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} f(x_i \mid \theta).$$

Note that " ; " denotes a separation between the two categories of input arguments: the parameters θ and the observations $x_1, \ldots, x_n$.

In practice it is often more convenient when working with the natural logarithm of the likelihood function, called the **log-likelihood**:

$$\ln \mathcal{L}(\theta\,;\,x_1, \ldots, x_n) = \sum_{i=1}^{n} \ln f(x_i \mid \theta),$$

or the average log-likelihood:

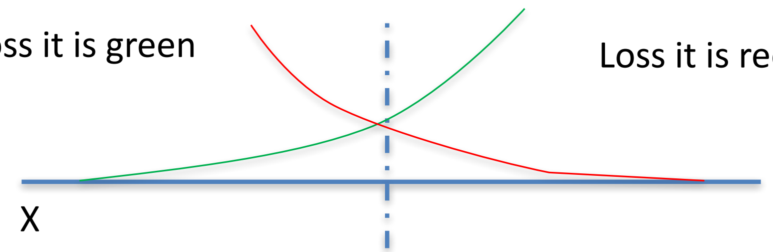
$$\hat{\ell} = \frac{1}{n} \ln \mathcal{L}.$$

The hat over ℓ indicates that it is akin to some estimator. Indeed, $\hat{\ell}$ estimates the expected log-likelihood of a single observation in the model.

The method of maximum likelihood estimates θ_0 by finding a value of θ that maximizes $\hat{\ell}(\theta; x)$. This method of estimation defines a **maximum likelihood estimator (MLE)** of θ_0:

How Should We Decide a Classification Boundary?

Another way would be to use a loss function to decide

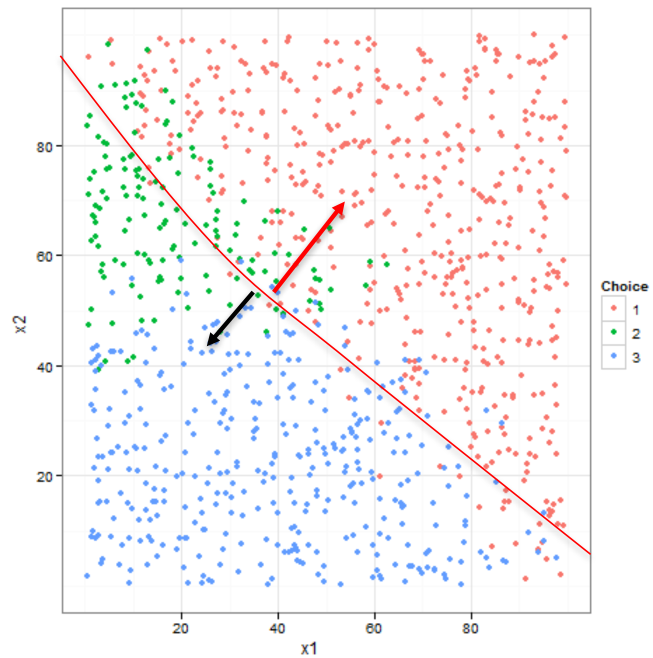


Lred(x): red loss as a function of vector X
(assume the actual outcome is red, then Lred(x) is larger when x is not likely predict to be red.
Lgreen(x): green loss as a function of vector X

Boundary could be where L(red)=L(green)

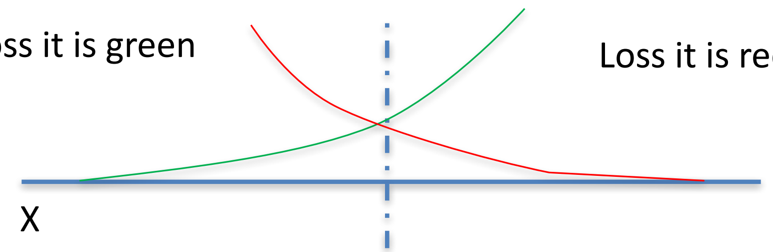
e.g. Cross Entropy $= -t\ln(f(\vec{x})) - (1-t)\ln(1-f(\vec{x}))$

Suppose we are detecting Red:
For now, lets look at Red = true
And Green as (not Red), ie False

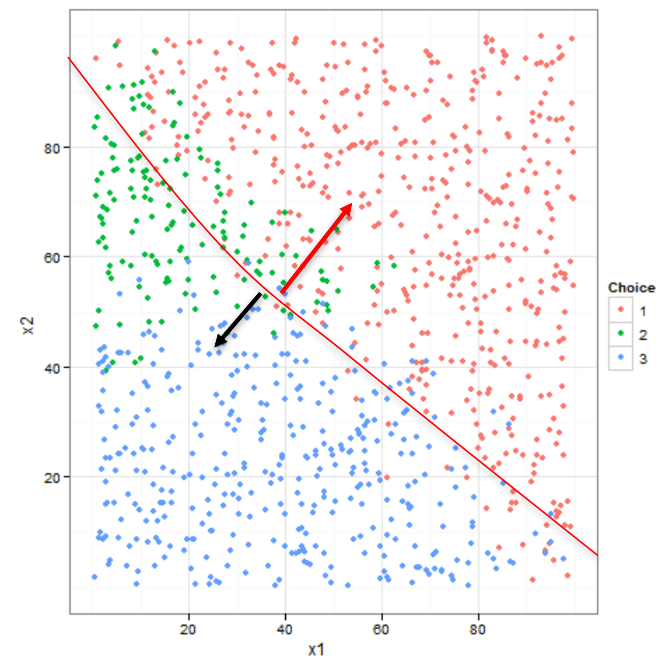


How Should We Decide a Classification Boundary?

- Loss function in math notation:



Suppose we are detecting Red:
For now, lets look at Red = true
And Green as (not Red), ie False



In math notation:

Loss Function: $L(\theta, \delta(X))$

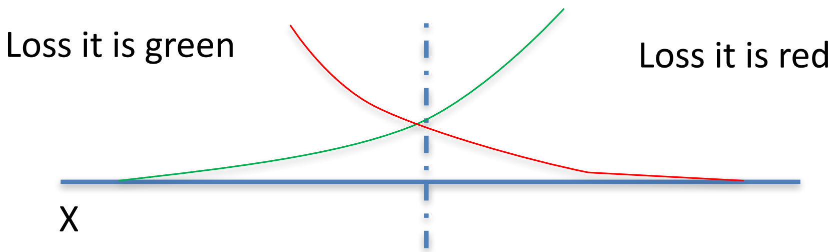
Expected Loss: $R(\theta, \delta) = \mathbb{E}(L(\theta, \delta(X)))$

e.g. Cross Entropy $= -t \ln(f(\vec{x})) - (1-t)\ln(1-f(\vec{x}))$

DataX

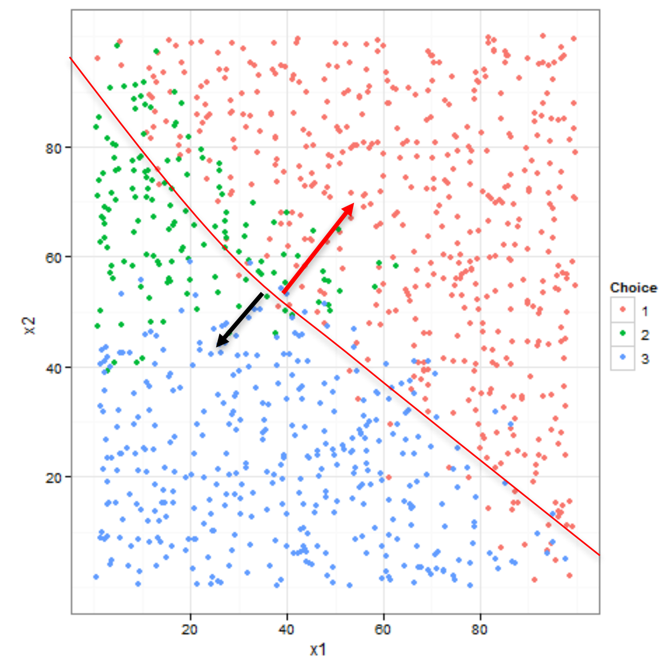
How Should We Decide a Classification Boundary?

- Another way would be to use a loss function to decide:



Boundary could be where L(red)=L(green)
BUT THERE IS A PROBLEM WITH THIS

Suppose we are detecting Red:
For now, lets look at Red = true
And Green as (not Red), ie False



Data X

How Should We Decide a Classification Boundary?

National Security

Consider the Application:

Red = there is a national security breach

Green = everything is OK

Reward or Risk is not same as loss

Predicted	Predicted No Breach	Predicted Breach
Actual: No Breach	Loss = 0 COST = 0	Loss = .5 COST = 10
Actual: Breach	Loss = 0.6 COST = 1000	Loss = 0 COST = 0

How Should We Decide a Classification Boundary?

Consider the Application:

Reward or Risk is not same as loss

National Security

Red = there is a national security breach

Green = everything is OK

Predicted	Predicted No Breach	Predicted Breach
Actual: No Breach	Loss = 0 COST = 0	Loss = .5 COST = 10
Actual: Breach	Loss = 0.6 COST = 1000	Loss = 0 COST = 0

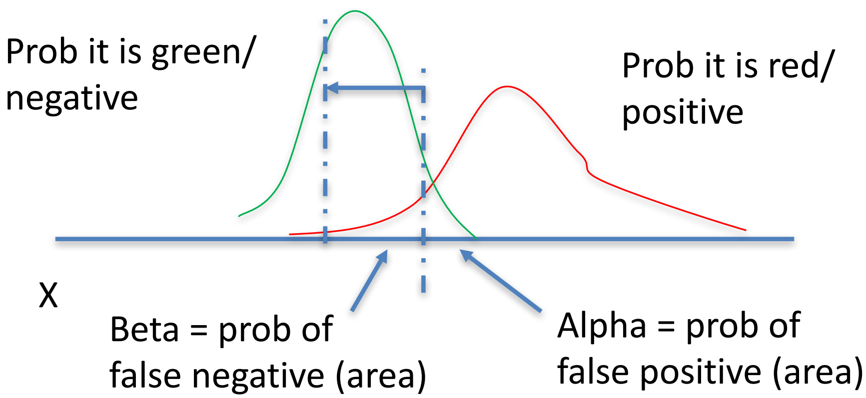
Supermarket Coupon by fingerprint

Red = fingerprint does not match for supermarket coupon

Green = fingerprint match

Predicted	Predicted Match	Predicted No Match
Actual: Match	Loss = 0 COST = 0	Loss = .5 COST = 1000
Actual: No match	Loss = 0.6 COST = 10	Loss = 0 COST = 0

A Risk Function will consider the application



Predicted	Predicted RED	Predicted GREEN
Actual: RED	#500 Risk = 0	#100 COST = 50 FN
Actual: GREEN	#300 COST = 10 FP	#300 Risk = 0

We want to set the boundary where the Risk of False Positive = Risk of False Negative

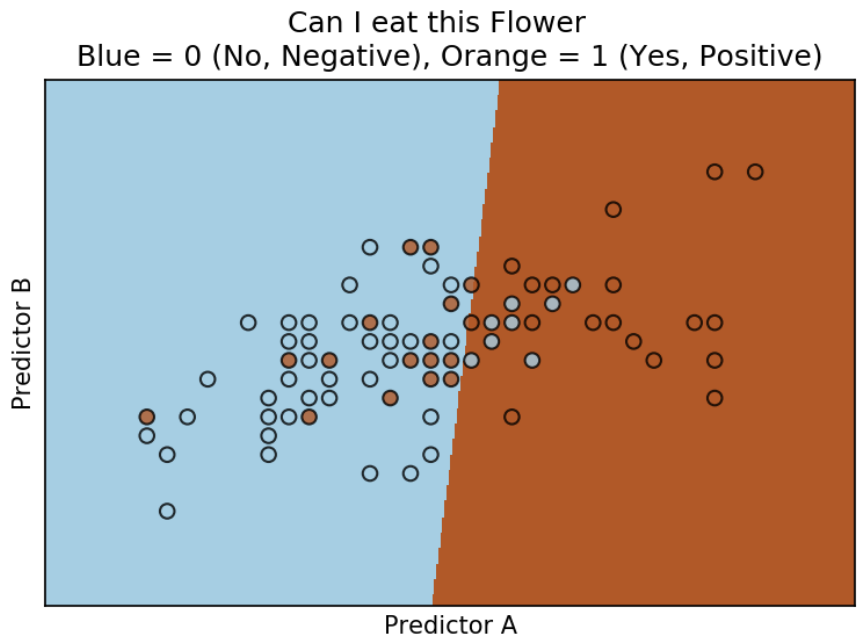
Alpha x FPcost = Beta x FNcost

Risk (Red = actual, Green = predicted)
= Prob (Red actual and Green predicted) x cost of false negative
= 100/600 x 50 = 8.3

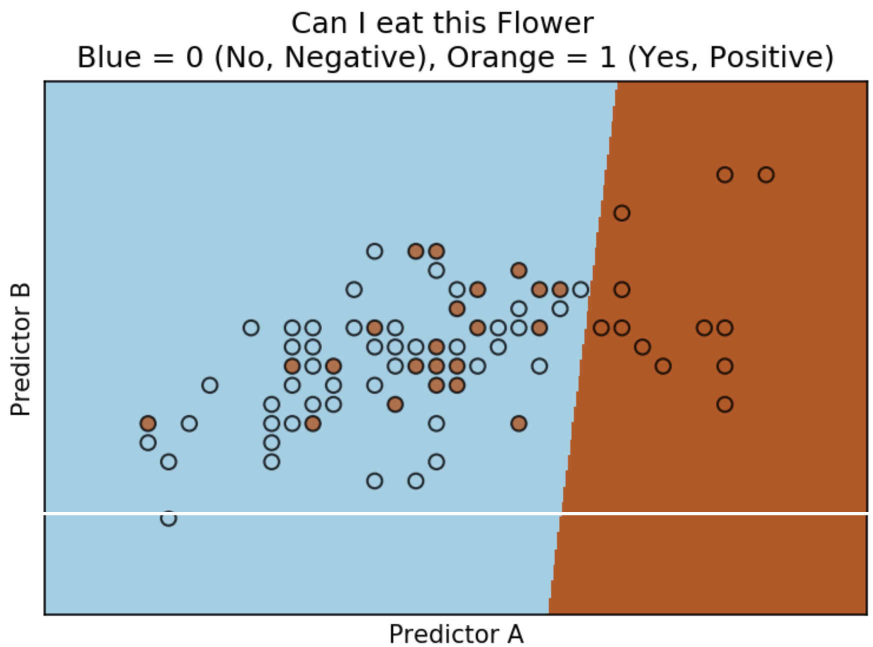
Risk (Green actual, Red predicted) = Prob (Green actual, red predicted) x cost of false positive)
=   300/600m x 10 = 5

Logistic Regression Example from Notebook

Example: 2 features/predictors, 1 output decision
Test if false positive or false negative rate per risk and cost. (in example too many false positives)
Option A: change boundary threshold from 0.5 to higher (0.66. in example)
Option B: Eliminate the most false negative data points from X until the boundary moves




```
Prediction Accuracy = 0.692307692308
Number of False Positives (misclassified blue points): 8
```




```
Prediction Accuracy = 0.717948717949
Number of False Positives (misclassified blue points): 0
```

End of Section