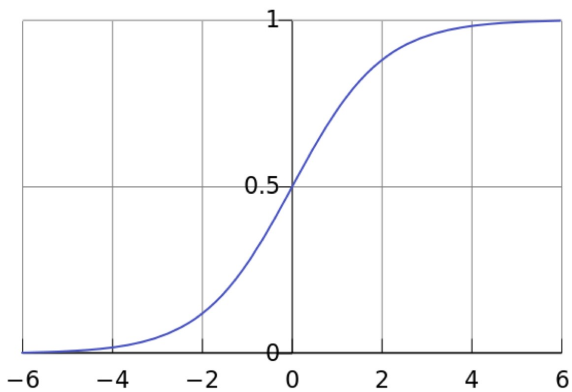
Artificial Intelligence

Machine Learning
Logistic Regression



Classification

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n) / x_i \in \mathbb{R}^d$ and y_i is discrete (categorical/qualitative), $y_i \in \mathbb{Y}$.

Example $\mathbb{Y} = \{-1, +1\}, \mathbb{Y} = \{0, 1\}$.

Task: Learn a classification function:

$$f : \mathbb{R}^d \longrightarrow \mathbb{Y}$$

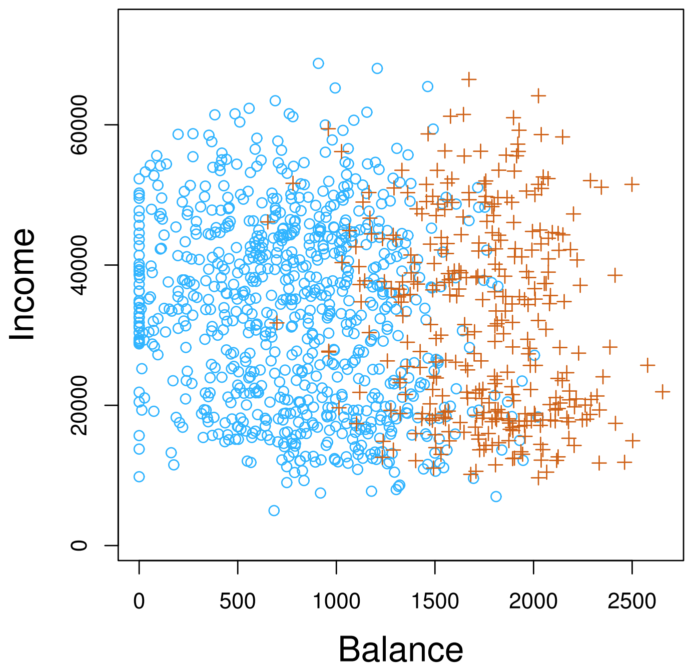
Linear Classification: A classification model is said to be linear if it is represented by a linear function f (linear hyperplane)

Classification: examples

1. Email Spam/Ham $\rightarrow$ Which email is junk?
2. Tumor benign/malignant $\rightarrow$ Which patient has cancer?
3. Credit default/not default $\rightarrow$ Which customers will default on their credit card debt?

Balance	**Income**	**Default**
300	\$20,000.00	no
2000	\$60,000.00	no
5000	\$45,000.00	yes
.	.	.
.	.	.
.	.	.

Classification: example



Credit: Introduction to Statistical Learning.

Classification

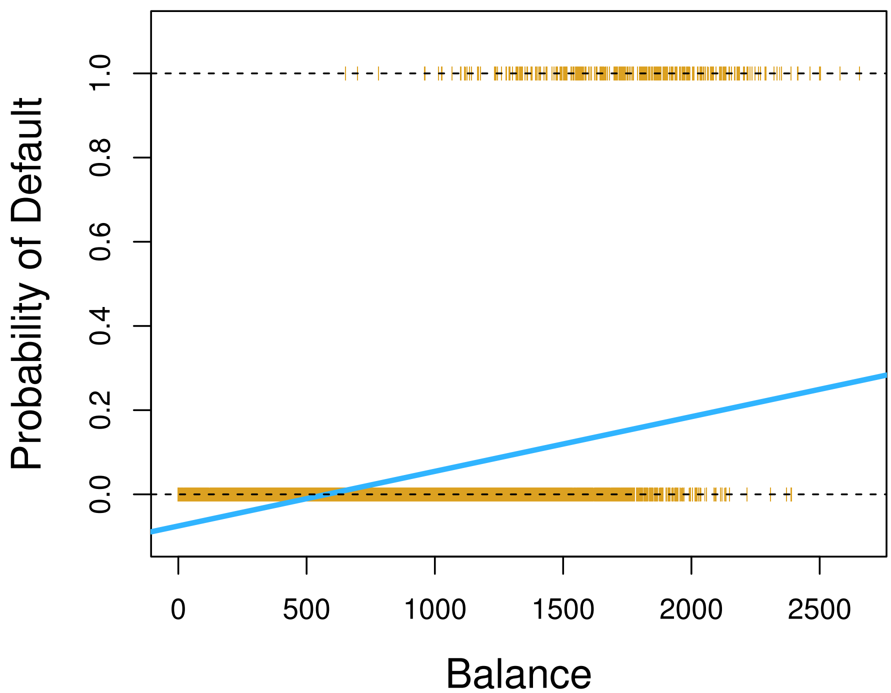
- We can't predict Credit Card Default with any certainty. Suppose we want to predict how likely is a customer to default. That is output a probability between 0 and 1 that a customer will default.
- It makes sense and would be suitable and practical.
- In this case, the output is real (regression) but is bounded (classification).

$$P(y|x) = P(\text{default} = \text{yes }|\text{balance})$$

Classification

- Can we use linear regression?
- Yes. However...
  - Works only for *Binary* classification (2 classes).
    Won't work for *Multiclass* classification e.g.,
    $\mathbb{Y} = \{$ green, blue, brown$\}$
    $\mathbb{Y} = \{$ stroke, heart attack, drug overdose$\}$
  - If we use linear regression, some of the predictions will be outside of [0,1].
  - Model can be poor. Example.

Classification: example



Credit: Introduction to Statistical Learning.

Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

We want $0 \leq f(x) \leq 1$; $f(x) = P(y = 1|x)$

Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

We want $0 \leq f(x) \leq 1$; $f(x) = P(y = 1|x)$
We use the sigmoid function:

$$g(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$

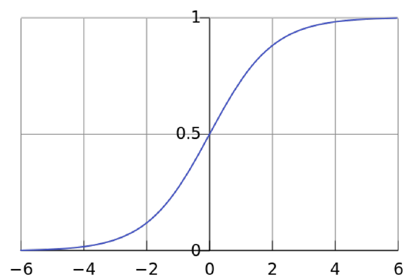
Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

We want $0 \leq f(x) \leq 1$; $f(x) = P(y = 1|x)$
We use the sigmoid function:

$$g(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$



$g(z) \to 1$ when $z \to +\infty$ $\qquad$ $g(z) \to 0$ when $z \to -\infty$

Logistic Regression

$$g(\beta_0 + \beta_1 x) = \frac{e^{(\beta_0+\beta_1 x)}}{1 + e^{(\beta_0+\beta_1 x)}}$$

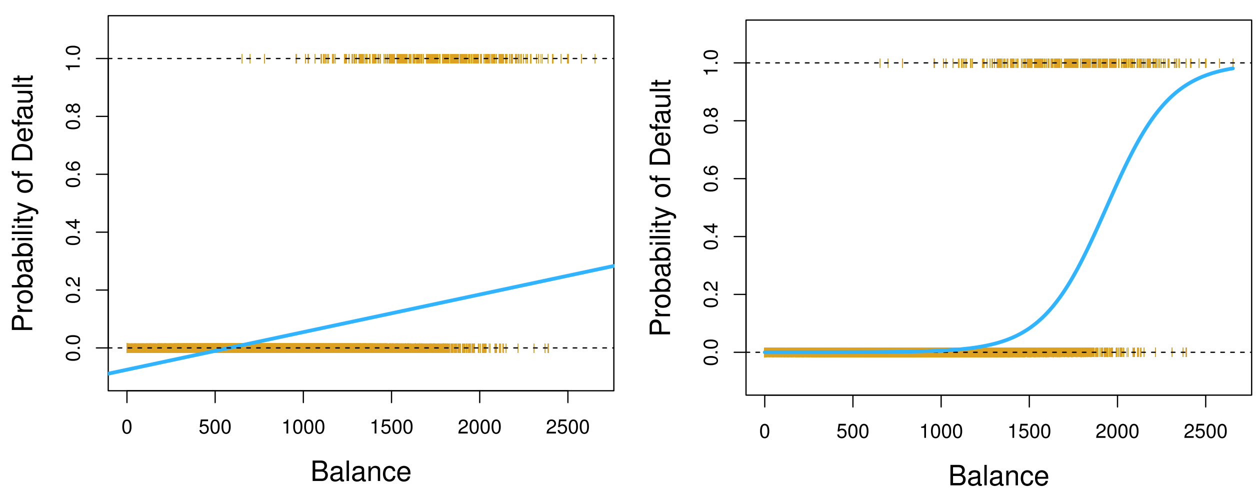
$$\boxed{\text{New } f(x) = g(\beta_0 + \beta_1 x)}$$

In general:

$$f(x) = g(\sum_{j=1}^{d} \beta_j x_j)$$

In other words, cast the output to bring the linear function quantity between 0 and 1.
Note: One can use other S-shaped functions.

Logistic Regression



Credit: Introduction to Statistical Learning.

Logistic regression is not a regression method but a classification method!

Logistic Regression

How to make a prediction?

- Suppose $\beta_0 = -10.65$ and $\beta_1 = 0.0055$. What is the probability of default for a customer with \$1,000 balance?

Logistic Regression

How to make a prediction?

- Suppose $\beta_0 = -10.65$ and $\beta_1 = 0.0055$. What is the probability of default for a customer with \$1,000 balance?

$$P(default = yes|balance = 1000) = \frac{1}{1 + e^{10.65 - 0.0055 * 1000}}$$

$$P(default = yes|balance = 1000) = 0.00576$$

Logistic Regression

How to make a prediction?

- Suppose $\beta_0 = -10.65$ and $\beta_1 = 0.0055$. What is the probability of default for a customer with $1,000 balance?

$$P(default = yes|balance = 1000) = \frac{1}{1 + e^{10.65 - 0.0055 * 1000}}$$

$$P(default = yes|balance = 1000) = 0.00576$$

- To predict the class:

$$\text{If} g(z) \geq 0.5 \text{ predict } y = 1 \quad (z \geq 0)$$

$$\text{If} g(z) < 0.5 \text{ predict } y = 0 \quad (z < 0)$$

Logistic Regression

How to find the β's?

$$R(\beta) = \frac{1}{n}\sum_{i=1}^{m}\frac{1}{2}(f(x)-y)^2$$

$$Loss = \frac{1}{2}(f(x)-y)^2$$

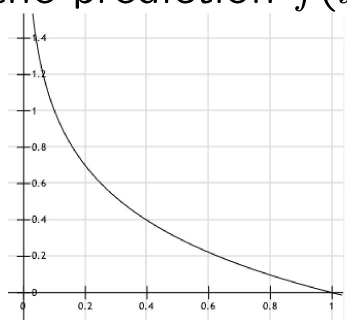
- Remember, $f(x)$ is now the logistic function so the $(f(x)-y)^2$ is not the quadratic function we had when f was linear.
- Cost is a complicated non-linear function!
- Many local optima, hence Gradient Descent will not find the global optimum!
- We need a different function that is convex.

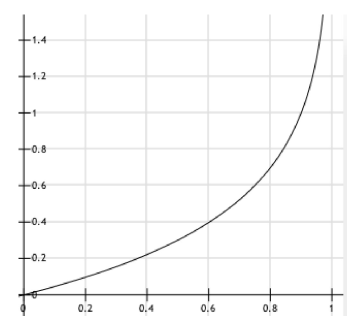
Logistic Regression

New Convex function:

$$Cost(f(x), y) = \begin{cases} -log(f(x)) & \text{if } y = 1 \\ -log(1 - f(x)) & \text{if } y = 0 \end{cases}$$

1. If $y = 1$ if the prediction $f(x) = 1$ then cost $= 0$
   If $y = 1$ if the prediction $f(x) = 0$ then cost $\to \infty$
2. If $y = 0$ if the prediction $f(x) = 0$ then cost $\to 0$
   If $y = 0$ if the prediction $f(x) = 1$ then cost $= \infty$





Case 1 Case 2

Logistic Regression

Nice convex functions!

Let's combine them in a compact function (because $y = 0$ or $y = 1$!):

$$Loss(f(x), y) = -ylogf(x) - (1-y)log(1-f(x))$$

$$R(\beta) = -\frac{1}{m}[\sum_{i=1}^{m} ylogf(x) + (1-y)log(1-f(x))]$$

Gradient Descent

Repeat {

Simultaneously update for all β's

$$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} R(\beta)$$

}

After some calculus:

Repeat {

Simultaneously update for all β's

$$\beta_j := \beta_j - \alpha \sum_{i=1}^{m} (f(x) - y) x_j$$

}

Note: Same as linear regression BUT with the new function f.

Credit

- When mentioned, some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013)" with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.