2019
怪兽
学堂

Logistic Regression

虾米

时间：2019-03

Regression

- A form of statistical modeling that attempts to evaluate the relationship between one variable (termed the dependent variable) and one or more other variables (termed the independent variables).
- A model for predicting one variable from another.

Linear Regression

- Regression used to fit a linear model to data where the dependent variable is continuous:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \quad + \beta_n X_n + \varepsilon$$

- Given a set of points (Xi,Yi), we wish to find a linear function (or line in 2 dimensions) that "goes through" these points.
- In general, the points are not exactly aligned:
  - Find line that best fits the points

What is Best Fit?

- The smaller the SSE, the better the fit
- Hence,
  - Linear regression attempts to minimize SSE
- Assume 2 dimensions

$$Y = \beta_0 + \beta_1 X$$

Logistic Regression

- Regression used to fit a curve to data in which the dependent variable is binary, or dichotomous
- Typical application: Medicine
  - We might want to predict response to treatment, where we might code survivors as 1 and those who don't survive as 0

Example

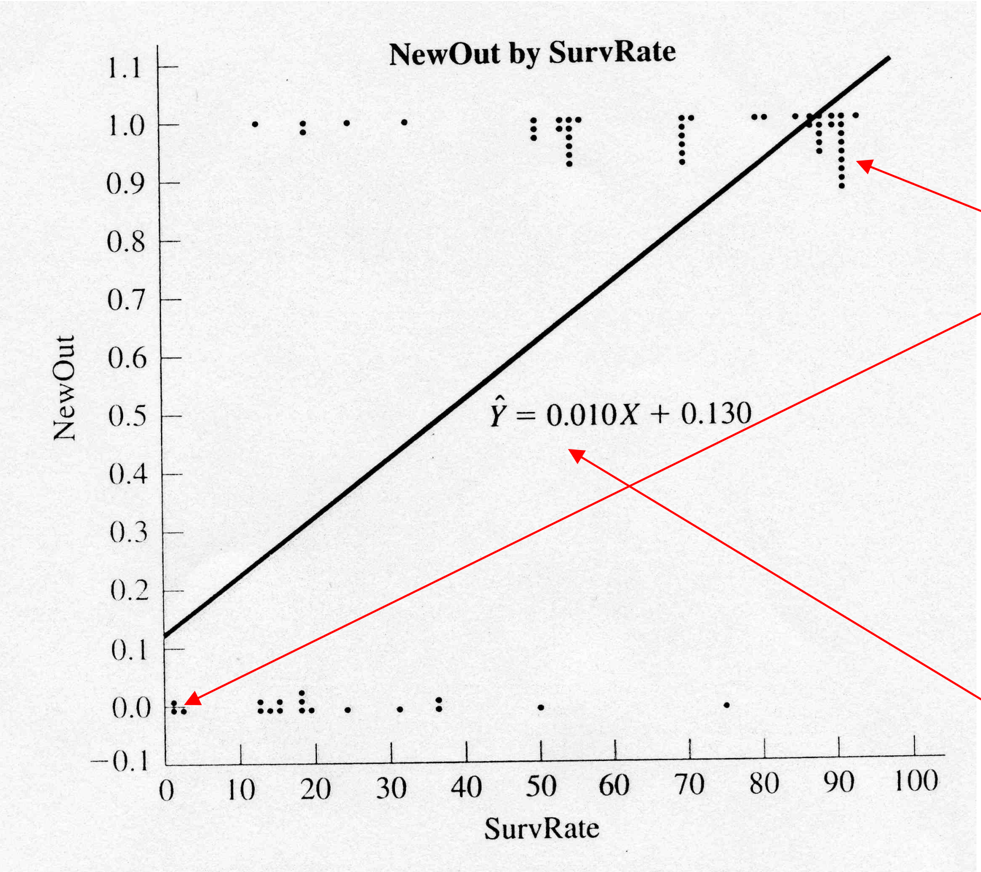


FIGURE 15·7 *Outcome as a function of SurvRate*

Observations:
For each value of SurvRate, the number of dots is the number of patients with that value of NewOut

Regression:
Standard linear regression

Problem: extending the regression line a few units left or right along the X axis produces predicted probabilities that fall outside of [0,1]

A Better Solution

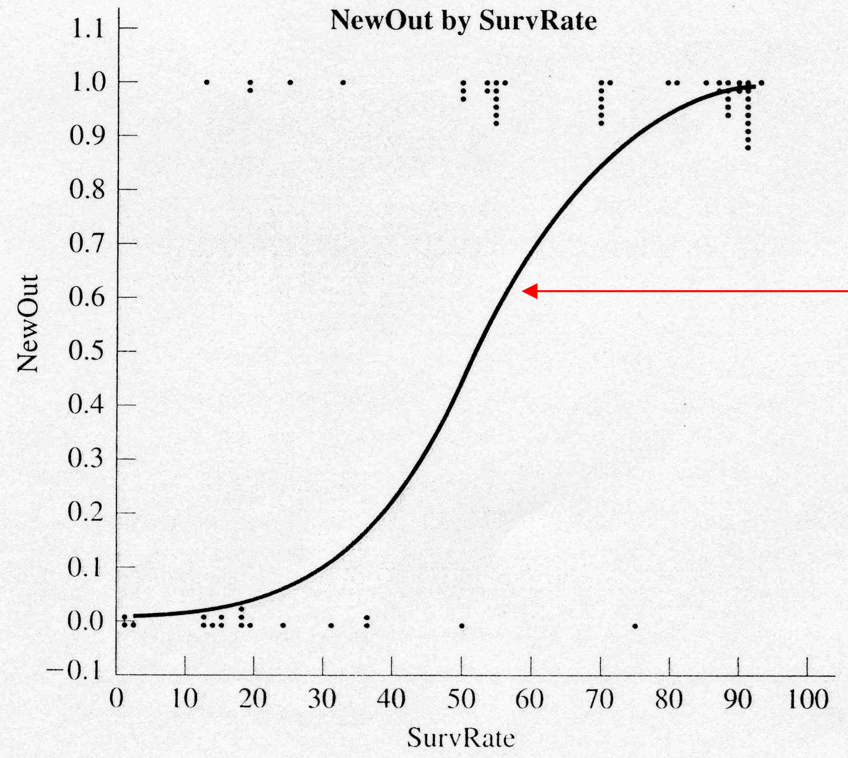


FIGURE 15.8 *More appropriate regression line for predicting outcome*

<u>Regression Curve</u>:
Sigmoid function!

(bounded by asymptotes *y*=0 and *y*=1)

Odds

- Given some event with probability p of being 1, the odds of that event are given by:

$$\text{odds} = p \,/\, (1-p)$$

- Consider the following data

Delinquent

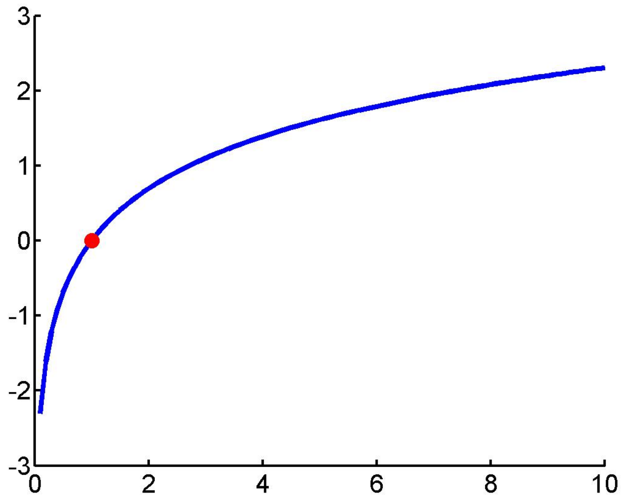
A		**Yes**	**No**	**Total**
	Normal	402	3614	4016
	High	101	345	446
		503	3959	4462

- The odds of being delinquent if you are in the Normal group are:

pdelinquent/(1–pdelinquent) = (402/4016) / (1 - (402/4016)) = 0.1001 / 0.8889 = 0.111

Logit Transform

- The logit is the natural log of the odds



- logit(p) = ln(odds) = ln (p/(1-p))

Logistic Regression

- In logistic regression, we seek a model:

$$\operatorname{logit}(p) = \beta_0 + \beta_1 X$$

- That is, the log odds (logit) is assumed to be linearly related to the independent variable X
- So, now we can focus on solving an ordinary (linear) regression!

Recovering Probabilities

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

$$\Leftrightarrow \frac{p}{1-p} = e^{\beta_0 + \beta_1 X}$$

$$\Leftrightarrow p = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

which gives *p* as a sigmoid function!

Figure out the loss function

- **Naive idea**

$$L_0 = \frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{1+e^{-w^T x^{(i)}}} - y^{(i)}\right)^2$$

However this loss function **is not a convex function** because of sigmoid function used here, which will make it very difficult to find the w to optimize the loss.

Figure out the loss function

- **Can we do better?**
  - Because of this is a binary classification problems, we can compute the loss for the two classes respectively

  - When target y = 1, the loss had better be very large when h(x) is close to 0, and the loss should be very small when h(x) is close to 1;

  - when target y = 0, the loss had better be very small when h(x) is close to 0, and the loss should be very large when h(x) is close to 1

$$L(h(x), y) = \begin{cases} -log(h(x)) & y = 1 \\ -log(1 - h(x)) & y = 0 \end{cases} = L(h(x), y) = -ylog(h(x)) - (1 - y)log(1 - h(x))$$

Find the best **w** to minimize the loss

$$
\begin{aligned}
\frac{\partial}{\partial w_j} L(w) &= -\left(y \frac{1}{g(w^T x)} - (1-y) \frac{1}{1-g(w^T x)}\right) \frac{\partial}{\partial w_j} g(w^T x) \\
&= -\left(y \frac{1}{g(w^T x)} - (1-y) \frac{1}{1-g(w^T x)}\right) g(w^T x)(1-g(w^T x)) \frac{\partial}{\partial w_j} w^T x \\
&= -(y(1-g(w^T x)) - (1-y) g(w^T x)) x_j \\
&= (h(x) - y) x_j
\end{aligned}
$$

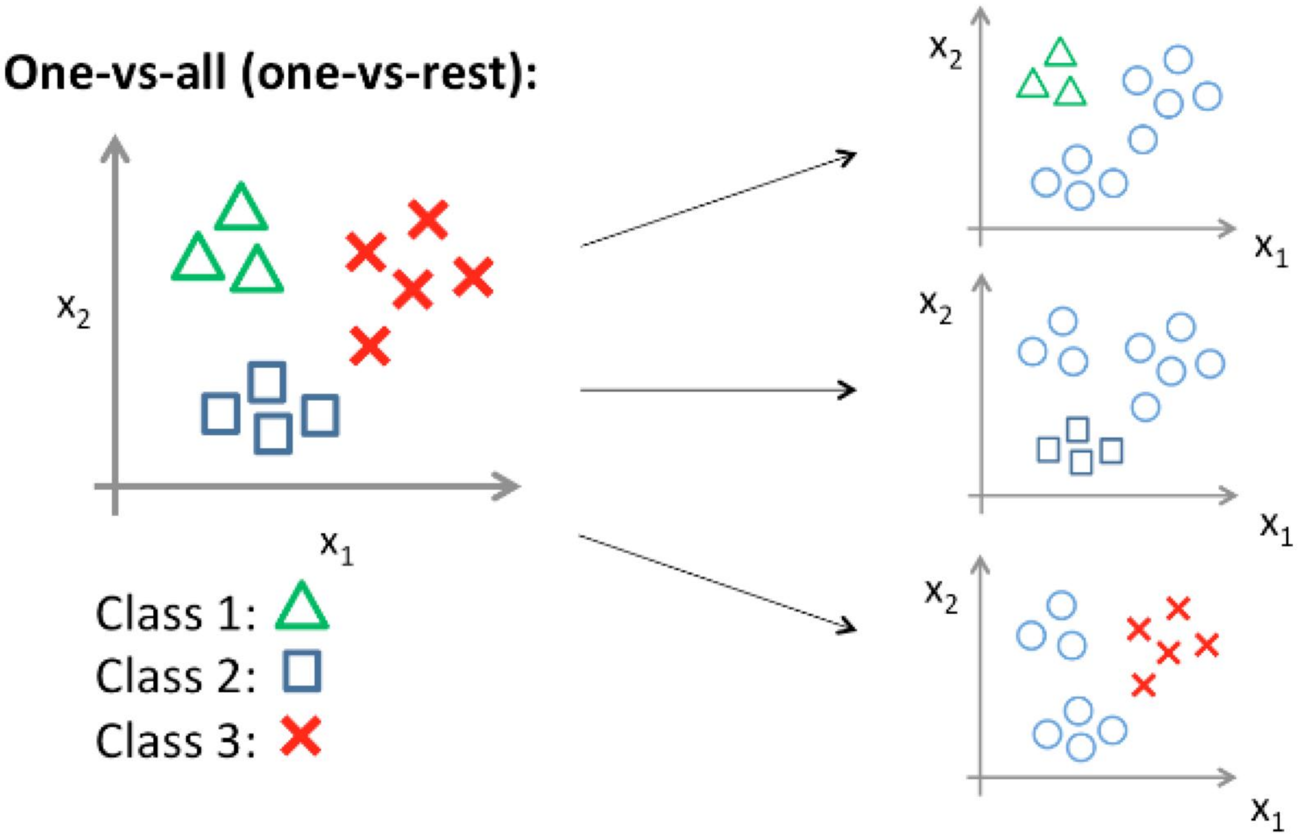
So the gradients are as following when considering all the samples:

$$
\frac{\partial}{\partial w_j} L(w) = \frac{1}{m} \sum_{i=1}^{m} (h(x) - y) x_j
$$

Softmax regression

multinomial logistic regression

Basic idea – Transfer multi-class classification into binary classification problem



One vs all → One vs one

Can we do better?

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

$$h(x^{(i)}) = \frac{e^{w_{y_j}^T x^{(i)}}}{\sum_{j=1}^{k} e^{w_j^T x^{(i)}}}$$

So why exponential function?

- It is a very simple and widely used non-linear function
- This function is strictly increasing
- This function is a convex function and its derivative is strictly increasing. That's to say, when the score is large, then make it even more larger.

Find the loss function

- we can use −log(h(x)) to compute the loss,

$$L_i = -log\left(h(x^{(i)})\right) = -log\left(\frac{e^{f_{y_j}^{(i)}}}{\sum_{j=1}^{k} e^{f_j^{(i)}}}\right) = -log\left(\frac{e^{w_{y_j}^T x^{(i)}}}{\sum_{j=1}^{k} e^{w_j^T x^{(i)}}}\right)$$

- Total loss for all sample is:

$$L = \frac{1}{m}\sum_{i=1}^{m} L_i = -\frac{1}{m}\sum_{i=1}^{m} log\left(h(x^{(i)})\right) = -\frac{1}{m}\sum_{i=1}^{m} log\left(\frac{e^{f_{y_j}^{(i)}}}{\sum_{j=1}^{k} e^{f_j^{(i)}}}\right) = -\frac{1}{m}\sum_{i=1}^{m} log\left(\frac{e^{w_{y_j}^T x^{(i)}}}{\sum_{j=1}^{k} e^{w_j^T x^{(i)}}}\right)$$

Is there any problem with the loss function

- Here is the trick by multiply the numerator and denominator by a constant C

$$\frac{e^{f_{y_j}^{(i)}}}{\sum_{j=1}^{k} e^{f_j^{(i)}}} = \frac{Ce^{f_{y_j}^{(i)}}}{C\sum_{j=1}^{k} e^{f_j^{(i)}}} = \frac{e^{f_{y_j}^{(i)}+logC}}{\sum_{j=1}^{k} e^{f_j^{(i)}+logC}}$$

set $logC = -max_j f_j^{(i)}$

Code example

Using softmax for multi-classification

2019
怪兽
学堂

THANKS