# Logistic Regression

CS 530
Chapman

Spring 2021

1

## Take 🏠 message for rest of course: Logistic Regression

- Logistic regression, lies at the intersection of regression and classification
- We will introduce the ubiquitous maximum-likelihood estimation
- We will introduce important notions in binary classification:
  - Confusion matrix
  - ROC curve
  - Pseudo-$R^2$ measures (McFadden's)
  - Does data trend according to logistic regression model?



2

## TABLE OF CONTENTS



3

## Intro to Logistic Regression

After covering regression, we move on to classification with Logistic Regression, at the intersection of regression and classification.

Regression models relate a continuous (or almost continuous), dependent variable and other, similarly continuous, variables.

But how do we model relations between continuous variable & categorical variables (e.g., gender, species, alma mater, hometown)?

4

## Intro to Logistic Regression

Example: Probability of defaulting on credit card payment given card balance

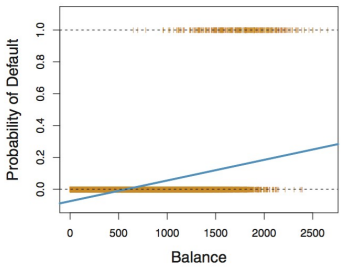



Image Source

5

## Intro to Logistic Regression

The $x$ axis is like simple linear-regression. But the $y$ axis is dichotomous (or, in this case, binary). So, we see that linear regression does not (and cannot) work well. The linear-regression model is

$$p(X) = \beta_0 + \beta_1 X$$

And, even if we insisted on fitting a linear regression, how can we interpret the resulting model? We cannot say that defaulting, as a binary condition, increases with the balance.

Logistic regression is a considerably better model for these kinds of data.

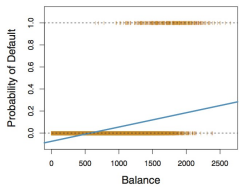



Image Source

6

## What is Logistic Regression?

To understand logistic regression, we need to understand the notion of *odds*.

The odds of an event, $x$, happening are defined as

$$\text{odds}(x) = \frac{p(x)}{1-p(x)}.$$

Here $p(x)$ is the probability that the event occurs.

For one fair die, the odds of the result 3 is

$$\text{odds}(x=3) = \frac{p(x=3)}{1-p(x=3)} = \frac{1/6}{5/6} = \frac{1}{5}.$$

This is often reported as 1:5.

7

## What is Logistic Regression?

Fitting a **linear regression** above, we use the model

$$p(X) = \beta_0 + \beta_1 X,$$

where $p(X)$ is the probability of an event occurring.

With **logistic regression**, we set the logarithm of the odds of $X$ (also known as log-odds) equal to a linear function of $X$:

$$\text{logit}(p) = \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

The logit function is the inverse of the logistic function.

The logit function.

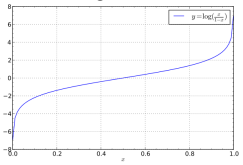


Image Source

8

## What is Logistic Regression?

We can simplify this to a more explicit expression for $p(X)$:

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

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

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

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

$$p(X)\left[1+e^{\beta_0+\beta_1 X}\right] = e^{\beta_0+\beta_1 X}$$

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

This is the logistic function.

The *sigmoid* curve

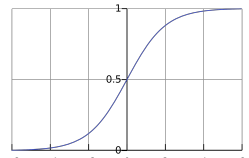


Image Source

9

## What is Logistic Regression?

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

Applying logistic regression to the defaulting data above, we get the bottom right figure. The logistic regression plot, which fits a sigmoid function to the data, makes much more sense and is more easily interpretable.

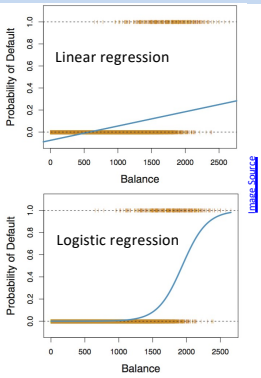



Image Source

10

## What is Logistic Regression?

For linear regression, a change of 1 unit in $x$ results in a change of $\beta_1$ units in $y$, regardless of $x$'s value:

$$p(X) = \beta_0 + \beta_1 X$$

For logistic regression, a change of 1 units in $x$ results in a change of $\beta_1$ units in the log-odds of $x$, or a change of $e^{\beta_1 X}$ in the odds of $x$. So the rate of change of $x$ changes with $x$:

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

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

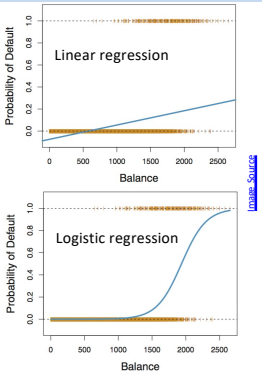



Image Source

11

## What is Logistic Regression?

How do we find optimal $\beta_0$ and $\beta_1$ for logistic regression?

For linear regression we used least squares on the RSS. We could use (non-linear) least squares here too.

But we will use a more general approach: *maximum (log) likelihood*

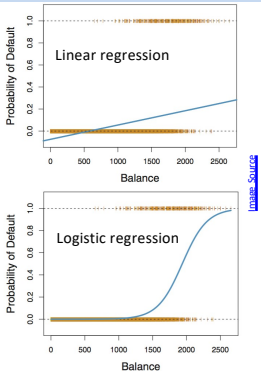



Image Source

12

## Probability and likelihood in statistics

In statistics, Probability is used to describe future events, before the data become available.

$$P(\text{die}_1 = 6 \wedge \text{die}_2 = 6)$$

Likelihood is used after the data are available to fit the parameters of a given model to those data.

E.g.: Likelihood of fair dice given (6,6)×3

13

## Maximum Likelihood: Fitting Logistic Regression

Given our available data, we estimate $\beta_0$ and $\beta_1$ such that $\hat{p}(x_i)$, for every trial $i$, will be as close as possible to the actual $p(x_i)$. In other words, we want $\hat{\beta}_0$ and $\hat{\beta}_1$ that result in $\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x}}$ that is as close to 0 for no defaulting and as close to 1 for defaulting as possible.

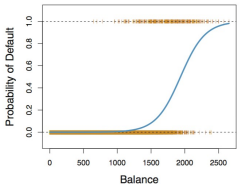



Image Source

14

## Maximum Likelihood: Fitting Logistic Regression

Given our available data, we estimate $\beta_0$ and $\beta_1$ such that $\hat{p}(x_i)$, for every trial $i$, will be as close as possible to the actual $p(x_i)$. In other words, we want $\hat{\beta}_0$ and $\hat{\beta}_1$ that result in $\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x}}$ that is as close to 0 for no defaulting and as close to 1 for defaulting as possible.

This intuition can be formalized by the likelihood function:

$$\ell(\beta_0, \beta_1; \vec{x}) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} \left(1 - p(x_j)\right),$$

which we want to maximize.

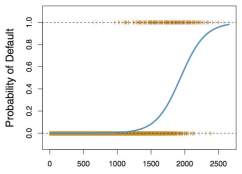



Image Source

15

## Maximum Likelihood: Fitting Logistic Regression

$$\ell(\beta_0, \beta_1; \vec{x}) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} \left(1 - p(x_j)\right)$$

$$= \prod_{i:y_i=1} \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_i}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_i}} \prod_{j:y_i=0} \left(1 - \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_j}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_j}}\right)$$

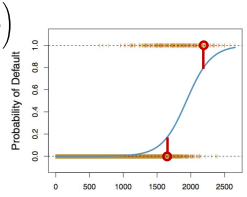



Image Source

16

## Maximum Likelihood: Fitting Logistic Regression

$$\ell(\beta_0, \beta_1; \vec{x}) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} \left(1 - p(x_j)\right)$$

If the logistic regression curve fits the data perfectly, every term in every product of the likelihood function will be 1, giving us

$$\ell(\beta_0, \beta_1; \vec{x}) = 1.$$

This would maximize the likelihood function in our case.

Maximum likelihood is a very general method for parameter estimation for non-linear models. In fact, the least-squares approach for linear regression is a special case of maximum likelihood estimation.

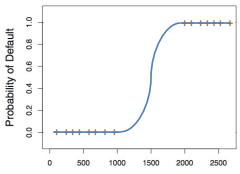



Image Source

17

## Maximum Likelihood: Fitting Logistic Regression

$$\ell(\beta_0, \beta_1; \vec{x}) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} \left(1 - p(x_j)\right)$$

Maximum likelihood is a very general method for parameter estimation for non-linear models. In fact, the least-squares approach for linear regression is a special case of maximum likelihood estimation.

18

## Maximum Likelihood: Fitting Logistic Regression

$$\ell(\beta_0, \beta_1; \vec{x}) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} \left(1 - p(x_j)\right)$$

It is typically simpler to work with the log-likelihood function (numerical stability & analytic derivation), which turns products into sums:

$$L(\beta_0, \beta_1; \vec{x}) = \ln[l(\beta_0, \beta_1; \vec{x})]$$

$$= \ln\left[\prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} \left(1 - p(x_j)\right)\right]$$

$$= \ln\left[\prod_{i=1}^{n} p(x_i)^{y_i} \left(1 - p(x_i)\right)^{1-y_i}\right]$$

$$= \sum_{i=1}^{n} \left[y_i \ln\left(p(x_i)\right) + (1 - y_i) \ln\left(1 - p(x_i)\right)\right]$$

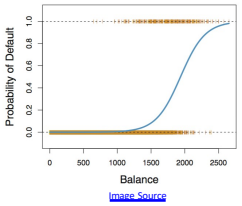


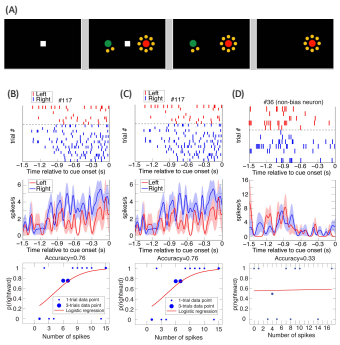
Image Source

19

## Using Logistic Regression

Example: Predicting answers before we ask the questions

Monkeys chose between small/large immediate/delayed reward (A). We fit a logistic regression model to the firing rate of DLPFC (B, D) & VS (C) neurons before leftward and rightward saccades. We then used those models to decide whether to predict future leftward/rightward saccades before the stimulus informing the monkey of the question came on.



Maoz et al., Front Neurosci., 2013

20

## Pros and Cons of Logistic Regression

What are the advantages of using logistic regression over linear regression for binary classification?

- Logistic function only gives values between 0 and 1
- Logistic function "transitions quickly" from 0 to 1
- Logistic regression does not require assumptions about normally distributed errors or homoscedasticity of errors

Disadvantages?

- Logistic regression is an iterative process without a closed-form solution. So, it may not converge due to
  - completely separable classes
  - multicollinearity
  - sparse data
- Logistic regression becomes cumbersome if we consider more than 2 categories in the dependent variable

21

## Prediction with Logistic Regression

Prediction based on a fitted logistic regression model—i.e., for given balance, how do we predict defaulting, a binary event?

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

The model returns the probability of default. We can then set a threshold (e.g., $p(X)=0.5$) and from that make a prediction (e.g., iff balance>1800 predict default).

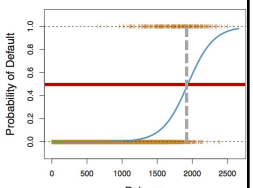


Image Source

22

## Prediction with Logistic Regression

Prediction based on a fitted logistic regression model—i.e., for given balance, how do we predict defaulting, a binary event?

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

The model returns the probability of default. We can then set a threshold (e.g., $p(X)=0.5$) and from that make a prediction (e.g., iff balance>1800 predict default).

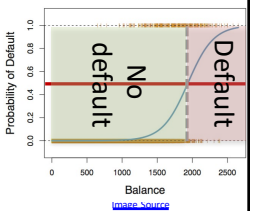


Image Source

23

## Multiple Logistic Regression

What if we wanted to find the probability of default based on more variables: balance, credit history, & income? We could use multiple logistic regression.

With multiple logistic regression, we just add more terms to the linear expression (on the right-hand side):

$$\ln\left(\frac{p(\vec{X})}{1 - p(\vec{X})}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n = \beta_0 + \vec{\beta} \cdot \vec{X},$$

for $\vec{X} = (X_1, X_2, \ldots, X_n)$ and $\vec{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)$. And similar algebra to before would give us:

$$p(\vec{X}) = \frac{e^{\beta_0 + \vec{\beta} \cdot \vec{X}}}{1 + e^{\beta_0 + \vec{\beta} \cdot \vec{X}}}.$$

And, again, finding good $\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_n$ can be achieved using the maxim (log) likelihood method.

24

## Multinomial Logistic Regression

What if we wanted to create a predictive model of mortgage payment from income, where our dependent variable, $y$, is categorical, but with more than two choices?

For example, say our subjects can pay (P), short sale (S), or foreclosure (F). And we had to predict which they will move.

This is termed multinomial logistic regression.


25

## Multinomial Logistic Regression

For multinomial logistic regression, we pick a "pivot" outcome for $y$. In our example, we can pick $y = \mathrm{P}$. Multinomial logistic regression then starts with equations for the non-pivot outcomes for $y$:

$$\ln\left[\frac{Pr(y=\mathrm{S})}{Pr(y=\mathrm{P})}\right] = \vec{\beta}_0^1 + \vec{\beta}^1 \cdot \vec{X},$$

$$\ln\left[\frac{Pr(y=\mathrm{F})}{Pr(y=\mathrm{P})}\right] = \vec{\beta}_0^2 + \vec{\beta}^2 \cdot \vec{X},$$

where the variables $x_1$,..., $x_q$ represent incomes of different people, ...

26

## Multinomial Logistic Regression

Modifying these equations, we get:

$$Pr(Y=\mathrm{S}) = Pr(Y=\mathrm{P})\, e^{\vec{\beta}_0^1 + \vec{\beta}^1 \cdot \vec{X}},$$

$$Pr(Y=\mathrm{F}) = Pr(Y=\mathrm{P})\, e^{\vec{\beta}_0^2 + \vec{\beta}^2 \cdot \vec{X}}$$

But,

$$Pr(Y=\mathrm{S}) + Pr(Y=\mathrm{F}) + Pr(Y=\mathrm{P}) = 1.$$

So

$$Pr(Y=\mathrm{P})\, e^{\vec{\beta}_0^1 + \vec{\beta}^1 \cdot \vec{X}} + Pr(Y=\mathrm{P})\, e^{\vec{\beta}_0^2 + \vec{\beta}^2 \cdot \vec{X}} + Pr(Y=\mathrm{P}) = 1.$$

And we get

$$Pr(Y=\mathrm{P}) = \frac{1}{e^{\vec{\beta}_0^1 + \vec{\beta}^1 \cdot \vec{X}} + e^{\vec{\beta}_0^2 + \vec{\beta}^2 \cdot \vec{X}} + 1}.$$

27

## Multinomial Logistic Regression

Multinomial logistic regression can similarly be extended beyond 3-class classification, to categorical variable $y$ with any finite number of values.

However, a more popular approach for multi-class classification is LDA. That said, LDA's assumption of Gaussian distributions for all classes is stronger than logistic regression's assumption that the log-odds of $\vec{X}$ (the variables) are a linear function of $\vec{X}$ (those variables).

28

## Questions from exit survey

- How are logistic and linear regression different?
- Multiple vs. multinomial logistic regression. How are they different? Why do we need both?

29

## Model Fit: Accuracy

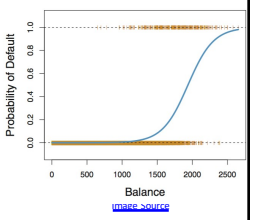
How do we test the goodness of fit for the logistic regression model?

The simplest and most intuitive way is to check the accuracy of the model. We set a threshold and check how often model predictions line up with our training data.

30

## Model Fit: Accuracy

For the original default data and a decision threshold of 0.5

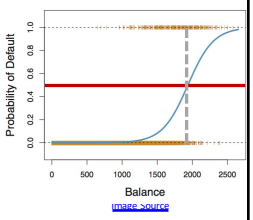



31

## Model Fit: Accuracy

For the original default data and a decision threshold of

$P(\text{Default}) = 0.5,$



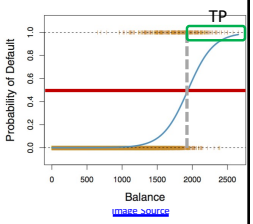

32

## Model Fit: Accuracy

For the original default data and a decision threshold of

$P(\text{Default}) = 0.5,$

True positive (TP): Predicted default and default occurred.




33

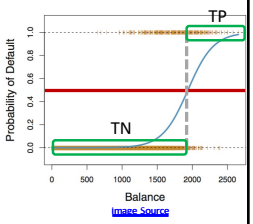
## Model Fit: Accuracy

For the original default data and a decision threshold of

$P(\text{Default}) = 0.5,$

True positive (TP): Predicted default and default occurred.

True negative (TN): Predicted no default and no default occurred.




34

## Model Fit: Accuracy

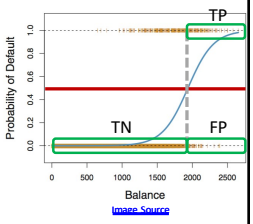
For the original default data and a decision threshold of

$P(\text{Default}) = 0.5,$

True positive (TP): Predicted default and default occurred.

True negative (TN): Predicted no default and no default occurred.

False positive (FP): Predicted default but no default occurred.




35

## Model Fit: Accuracy

For the original default data and a decision threshold of

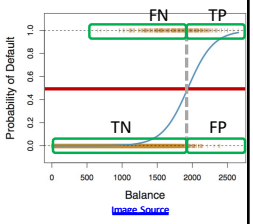
$P(\text{Default}) = 0.5,$

True positive (TP): Predicted default and default occurred.

True negative (TN): Predicted no default and no default occurred.

False positive (FP): Predicted default but no default occurred.

False negative (FN): Predicted no default but default occurred.




36

## Model Fit: Confusion Matrix

More generally, we can describe accuracy for *binary classification* with a confusion matrix.

False Positive = Type I Error

False Negative = Type II Error

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

| | Predict True | Predict False |
|---|---|---|
| Actual True | TP | FN |
| Actual False | FP | TN |

37

## Model Fit: Confusion Matrix

Other measures stemming from the confusion matrix include:

True Positive Rate (hit rate, recall, **sensitivity**):

$$\text{TPR} = \frac{TP}{TP + FN}.$$

True Negative Rate (**specificity**, selectivity):

$$\text{TNR} = \frac{TN}{TN + FP}.$$

False Positive Rate (fall out, false alarm):

$$\text{FPR} = \frac{FP}{TN + FP} = 1 - \text{TNR}.$$

TPR and TNR: higher values mean better model fit.

| | Predict True | Predict False |
|---|---|---|
| Actual True | TP | FN |
| Actual False | FP | TN |

38

## Model Fit: Confusion Matrix

TPR, TNR, & FPR all tradeoff and depend on the threshold we set.

$\text{TPR} = \frac{\text{TP}}{TP+FN}$ (sensitivity)

$\text{TNR} = \frac{\text{TN}}{FP+TN}$ (specificity)

$\text{FPR} = \frac{\text{FP}}{TN+FP} = 1 - \text{TNR}$

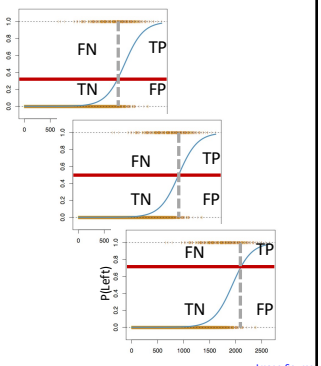
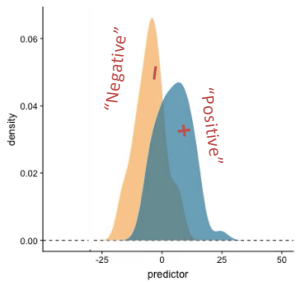



Image Source

39

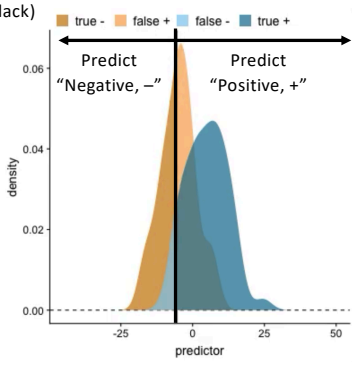
## Model Fit: ROC Curve




40

## Model Fit: ROC Curve

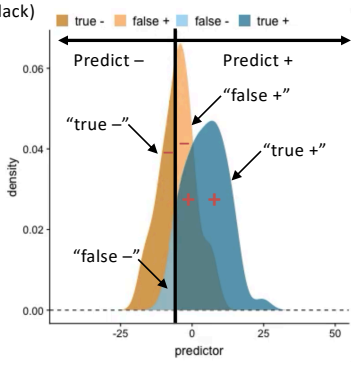
Predicted (in black)
Actual (in red)




41

## Model Fit: ROC Curve

Predicted (in black)
Actual (in red)




42

## Model Fit: ROC Curve

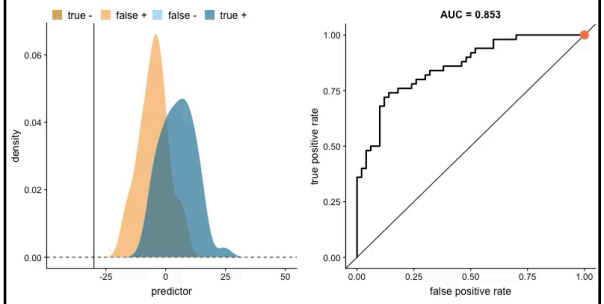
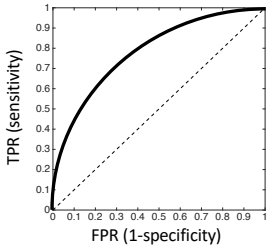



Image Source

43

## Model Fit: ROC Curve

The Receiver Operating Characteristic (ROC) curve for *binary classification* plots TPR (sensitivity) against FPR (fall-out, 1-specificity) for various possible thresholds.

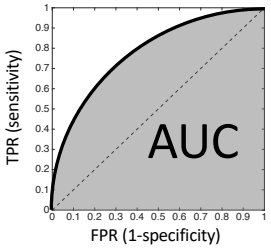



44

## Model Fit: ROC Curve

The Receiver Operating Characteristic (ROC) curve for *binary classification* plots TPR (sensitivity) against FPR (fall-out, 1-specificity) for various possible thresholds.

The higher the area under the curve (AUC) the better the model fit for the binary classification. AUC=1 for perfect classification. The dashed line, with AUC=0.5, designates chance-level performance.




45

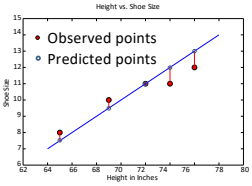
## Model Fit: McFadden Pseudo-$R^2$

For linear regression:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum(y_i - y)^2}.$$

This captures 3 notions about $R^2$:

1. R-squared as explained variability
2. R-squared as improvement from null model to fitted model
3. R-squared as the square of the correlation coefficient (r)




46

## Model Fit: McFadden Pseudo-$R^2$

The $R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - y)^2}$ we encountered for linear regression *does not work* for logistic regression. (For example, looking at differences from $y$ makes little sense for logistic regression.)

Logistic regression is calculated based on maximum-likelihood estimates, an iterative process. So, other $R^2$ measures are needed.

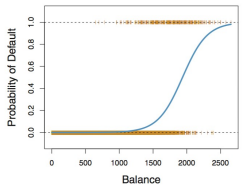



Image Source

47

## Model Fit: McFadden Pseudo-$R^2$

But there exist several pseudo-$R^2$ indicators, which capture some of the characteristics of the $R^2$. The most common of which is *McFadden's pseudo-$R^2$* ($R^2_{McF}$):

$$R^2_{McF} = 1 - \frac{\ln(L_M)}{\ln(L_0)}$$

$L_M$ is the likelihood of the full, fitted model, and $L_0$ is the likelihood of the null model—the model with no coefficients except for the intercept. Hence $0 \leq L_M, L_0 \leq 1 \Longrightarrow \ln(L_M), \ln(L_0) \leq 0$. Comparing $R^2_{McF}$ for 2 models on the same data, $R^2_{McF}$ would be higher for the model with the higher likelihood.

$R^2_{McF}$ captures the $R^2$ as a measure of explained variability and as an improvement from a null model to a fitted model. But it does not capture the $R^2$ as the square of the correlation.

48

## Model Fit: Wald Test

How statistically significant is each coefficient estimate $\hat{\beta}_j$ in our model? For linear regression we carried out t-tests on the variables. For logistic regression we can run a similar test based on the *z-statistic*, named the Wald test:

$$W_j = \frac{\hat{\beta}_j}{SE_{\hat{\beta}_j}},$$

where SE is the standard error of the mean. $|W_j| > 2$ (roughly) is significant at the 95% level.

49

## Python code

```
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
X, y = load_iris(return_X_y=True)
clf = LogisticRegression(random_state=0, solver='lbfgs',
                         multi_class='multinomial').fit(X, y)

                            # solver:Algorithm to use in the optimization problem.
                            # For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga'
                            # are faster for large ones.

                            # For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs'
                            # handle multinomial loss; 'liblinear' is limited to one-versus-rest schemes.



clf.predict(X[:2, :])
```

```
array([0, 0])
```

```
clf.predict_proba(X[:2, :]) # Probability estimates
                            # The returned estimates for all classes are ordered by the label of classes.


```

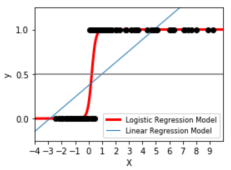
```
array([[  9.81805980e-01,   1.81940056e-02,   1.43354257e-08],
       [  9.71803214e-01,   2.81967563e-02,   2.97462927e-08]])
```

```
clf.score(X, y)    #Returns the mean accuracy on the given test data and labels.
```

```
0.9733333333333338
```

50

```
# General a toy dataset:s it's just a straight line with some Gaussian noise:
xmin, xmax = -5, 5
n_samples = 100
np.random.seed(0)
X = np.random.normal(size=n_samples)
y = (X > 0).astype(np.float)
X[X > 0] *= 4
X += .3 * np.random.normal(size=n_samples)

X = X[:, np.newaxis]

# Fit the classifier
clf = linear_model.LogisticRegression(C=1e5, solver='lbfgs')
clf.fit(X, y)

# and plot the result
plt.figure(1, figsize=(4, 3))
plt.clf()
plt.scatter(X.ravel(), y, color='black', zorder=20)
X_test = np.linspace(-5, 10, 300)


def model(x):
    return 1 / (1 + np.exp(-x))


loss = model(X_test * clf.coef_ + clf.intercept_).ravel()
plt.plot(X_test, loss, color='red', linewidth=3)

ols = linear_model.LinearRegression()
ols.fit(X, y)
plt.plot(X_test, ols.coef_ * X_test + ols.intercept_, linewidth=1)
plt.axhline(.5, color='.5')

plt.ylabel('y')
plt.xlabel('X')
plt.xticks(range(-5, 10))
plt.yticks([0, 0.5, 1])
plt.ylim(-.25, 1.25)
plt.xlim(-4, 10)
plt.legend(('Logistic Regression Model', 'Linear Regression Model'),
           loc='lower right', fontsize='small')
```




51

## Summary

- For Logistic Regression $\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$ or $p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$. It is therefore a classification method rather than strictly a regression.
- Logistic regression is fit iteratively using the maximum likelihood method.
- It has extensions for multiple variables (multiple logistic regression) and multiple classes (multinomial logistic regression).
- Discussing model accuracy for binary classification, we encountered
  - Confusion Matrix: Accuracy, Sensitivity, Specificity
  - ROC Curve
  - McFadden Pseudo-R2
  - Wald Test

52

## Take 🏠 message for rest of course: Logistic Regression

- Our first instance of a full-blown classification method, logistic regression, lies at the intersection of regression and classification
- We introduced the ubiquitous maximum-likelihood estimation
- We encountered important notions in binary classification :
  - Confusion matrix
  - ROC curve
  - Pseudo-$R^2$ measures (McFadden's)
  - Test whether data trends according to the logistic regression model?

53