Week 4: Logistic Regression and Classification

MATH-516 Applied Statistics

Linda Mhalla

2024-03-13

Section 1

Logistic Regression for Bernoulli and Binomial Data

Generalized Linear Model for Binary Variables

- In the case of a binary response variable, assume Y_n follows a Bernoulli distribution with parameter π_n, $Y_n \sim \text{Bin}(\pi_n)$, where

$$\pi_n = \Pr(Y_n = 1 \mid \mathbf{X}_n) = \mathbb{E}(Y_n \mid \mathbf{X}_n)$$

- An appropriate link function for binary responses is the **logit** function

$$g(z) := \text{logit}(z) = \log\left(\frac{z}{1-z}\right)$$

- The **logistic regression model** is

$$g(\pi_n) = \log\left(\frac{\pi_n}{1-\pi_n}\right) = \eta_n := \beta_0 + \beta_1 \mathrm{X}_{n1} + \cdots + \beta_p \mathrm{X}_{np}$$

- The logit function g is the **quantile function of the logistic distribution** and links $\mathbb{E}(Y_n \mid \mathbf{X}_n) = \pi_n(\mathbf{X}_n)$ and η_n

Logistic Regression: Logit Function

- The logistic model is

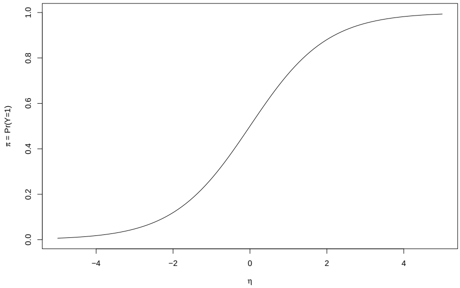
$$\eta_n = \log\left(\frac{\pi_n}{1-\pi_n}\right) = \beta_0 + \beta_1 \mathrm{X}_{n1} + \cdots + \beta_p \mathrm{X}_{np}$$

- This model can also be written on the mean scale by using the inverse-logit function,

$$\mathbb{E}(Y_n \mid \mathbf{X}_n) = \pi_n = \frac{\exp(\beta_0 + \beta_1 \mathrm{X}_{n1} + \cdots + \beta_p \mathrm{X}_{np})}{1 + \exp(\beta_0 + \beta_1 \mathrm{X}_{n1} + \cdots + \beta_p \mathrm{X}_{np})}$$

- We have an expression for the mean $\pi_n = \mathbb{E}(Y_n \mid \mathbf{X}_n)$ as a function of the explanatory variables $\mathbf{X}_n$, but ...
- what does this function look like?
- what does this tell us about the relationship between π_n and η_n (and thus $\mathbf{X}_n$)?

Logistic Distribution Function



- Notice that π is an increasing function of $\eta = \beta_0 + \sum_{j=1}^{p} \beta_j \mathrm{X}_j$
  - If β_j is positive and X_j increases, $\Pr(Y = 1)$ also increases
  - If β_j is negative and X_j increases, $\Pr(Y = 1)$ decreases
- We also see that the relationship between $\Pr(Y = 1)$ and η (and thus each X_j) is non-linear

Parameter interpretations in terms of odds

- Quantifying the effect sizes in logistic regression is not easy because it's a nonlinear model
- The coefficients can be interpreted in terms of **odds** and **odds ratios**
- Let $\pi = \Pr(Y = 1 \mid \mathrm{X}_1, \ldots, \mathrm{X}_p)$, the logistic regression model is

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 \mathrm{X}_1 + \cdots + \beta_p \mathrm{X}_p$$

- By exponentiating both sides, we obtain

$$\mathrm{odds}(Y \mid \mathbf{X}) = \frac{\pi(\mathbf{X})}{1-\pi(\mathbf{X})} = \exp(\beta_0 + \beta_1 \mathrm{X}_1 + \cdots + \beta_p \mathrm{X}_p),$$

where $\pi(\mathbf{X})/\{1-\pi(\mathbf{X})\}$ are the odds of $\Pr(Y = 1 \mid \mathbf{X})$ relative to $\Pr(Y = 0 \mid \mathbf{X})$

Odds

- The logit function corresponds to modelling the **log-odds**
- The odds for binary Y are the quotient

$$\text{odds}(\pi) = \frac{\pi}{1-\pi} = \frac{\Pr(Y=1)}{\Pr(Y=0)}$$

- For example, an odds of 4 means that the probability that $Y=1$ is four times higher than the probability that $Y=0$
- An odds of 0.25 means the probability that $Y=1$ is only a quarter times the probability that $Y=0$, or equivalently, the probability that $Y=0$ is four times higher than the probability that $Y=1$

$\Pr(Y=1)$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Odds	0.11	0.25	0.43	0.67	1	1.5	2.33	4	9
Odds (frac.)	$\frac{1}{9}$	$\frac{1}{4}$	$\frac{3}{7}$	$\frac{2}{3}$	1	$\frac{3}{2}$	$\frac{7}{3}$	4	9

Interpretation of the intercept in terms of the odds

- When $\mathrm{X}_1 = \cdots = \mathrm{X}_p = 0$, it is clear that

$$\mathrm{odds}(Y \mid \mathbf{X} = \mathbf{0}_p) = \exp(\beta_0)$$

  and

$$\Pr(Y = 1 \mid \mathrm{X}_1 = 0, \ldots \mathrm{X}_p = 0) = \frac{\exp(\beta_0)}{1 + \exp(\beta_0)}$$

  which represents the probability that $Y = 1$ when $\mathbf{X} = \mathbf{0}_p$
- As for linear regression, $\mathrm{X}_1 = \cdots = \mathrm{X}_p = 0$ might not be physically possible, in which case there is no sensible interpretation for β_0

Parameter interpretation in terms of the odds ratio

Consider for simplicity a logistic model of the form logit$(\pi) = \beta_0 + \beta_1 x$

The factor $\exp(\beta_1)$ is the change in odds when X increases by one unit,

$$\text{odds}(Y \mid \mathrm{X} = x + 1) = \exp(\beta_1) \times \text{odds}(Y \mid \mathrm{X} = x)$$

- If $\beta_1 = 0$ then the odds ratio is unity
  - meaning that the variable X is not associated with the odds of Y
- If β_1 is positive, then the odds ratio $\exp(\beta_1)$ is larger than one,
  - meaning that, as X increases, the odds of Y increases
- If β_1 is negative, the odds ratio $\exp(\beta_1)$ is smaller than one,
  - meaning that, as X increases, the odds of Y decreases

Interpretation of β_k in terms of odds ratio

For the logistic model, the odds ratio when $\mathrm{X}_k = x_k + 1$ versus $\mathrm{X}_k = x_k$ when $\mathrm{X}_j = x_j$ $(j = 1, \dots, p, j \neq k)$ is

$$\frac{\mathrm{odds}(Y \mid \mathrm{X}_k = x_k + 1, \mathrm{X}_j = x_j, j \neq k)}{\mathrm{odds}(Y \mid \mathrm{X}_k = x_k, \mathrm{X}_j = x_j, j \neq k)} = \frac{\exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j + \beta_k\right)}{\exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)}$$
$$= \exp(\beta_k)$$

When X_k increases by one unit **and all the other covariates are held constant**, the odds of Y changes by a factor $\exp(\beta_k)$

- The odds increase if $\exp(\beta_k) > 1$, i.e., if $\beta_k > 0$
- The odds decrease if $\exp(\beta_k) < 1$, i.e., if $\beta_k < 0$

Assessing Quality of Fit

The quality of fit of $\hat{\pi}_n$ to y_n (either 0 or 1) is measured by the **deviance**[1]

$$\begin{aligned}\mathrm{Dev}\left(\hat{\pi}_i, y_i\right) &= \begin{cases} -2\log\hat{\pi}_i & \text{if } y_i = 1 \\ -2\log\left(1-\hat{\pi}_i\right) & \text{if } y_i = 0 \end{cases} \\ &= y_i\left(-2\log\hat{\pi}_i\right) + \left(1-y_i\right)\left\{-2\log\left(1-\hat{\pi}_i\right)\right\}\end{aligned}$$

- The Residual Deviance

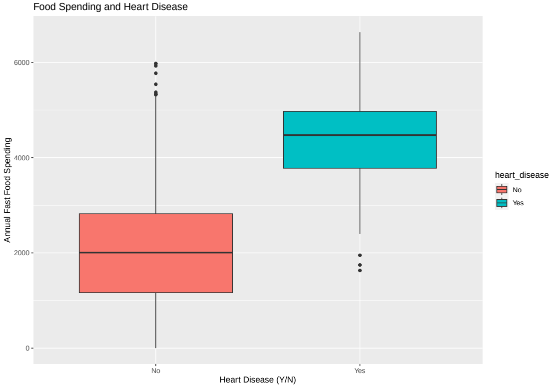
  $$D = \sum_{n=1}^{N} \mathrm{Dev}\left(\hat{\pi}_n, y_n\right)$$

  should behave like χ^2_{N-p-1} if the model is correct **and** n_i's (sample sizes per combination of covariates) are large. ΔD (equiv. LRT) can otherwise be used for model comparison (but not with saturated model)
- The deviance residuals $\epsilon_n^d = sign(y_n - \hat{\pi}_n)\sqrt{\mathrm{Dev}\left(\hat{\pi}_n, y_n\right)}$ have the same interpretation as for the ordinary linear model

[1] the likelihood of the saturated model is 1

Example: Heart Disease Data

Understand how drinking coffee, spending on fast food, and annual income are related to the likelihood of heart disease



Example: Heart Disease Data

```
Call:
glm(formula = factor(heart_disease) ~ factor(coffee_drinker) +
    fast_food_spend + income, family = binomial(link = "logit"),
    data = heart_data)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
factor(coffee_drinker)1 -6.468e-01  2.363e-01  -2.738  0.00619 **
fast_food_spend          2.295e-03  9.276e-05  24.738  < 2e-16 ***
income                   3.033e-06  8.203e-06   0.370  0.71152
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.5  on 9996  degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```

Example: Heart Disease Data

- All covariates except `income` are significant
  - coffee drinking is associated with a decrease in the odds of having a heart disease: a decrease of $\exp(-0.65) \approx 0.52$, ceteris paribus
  - spending in fast food is associated with an increase in the odds of having a heart disease: an increase of $\exp(2.3 * 10^{-3}) \approx 1$, ceteris paribus
- What about predictions?

```
head(predict(log_reg, type="link")) #linear combination of covariates
```

```
        1         2         3         4         5         6 
-6.549544 -6.791338 -4.614261 -7.724689 -6.245449 -6.217871 
```

```
head(predict(log_reg, type="response")) #predicted probabilities
```

```
           1            2            3            4            5            6 
0.0014287239 0.0011222039 0.0098122716 0.0004415893 0.0019355062 0.0019895182 
```

Example: Heart Disease Data

What about binary classification?

Once you have predicted probabilities, how large should a predicted probability be to predict a heart disease?

- a cutoff of 0.5 seems a fair choice, but why?
  - it estimates the Bayes Classifier

$$\mathcal{C}_{Bayes}(\mathbf{x}) = \underset{0 \leq k \leq J-1}{\arg\max} \Pr(Y = k | \mathbf{X} = \mathbf{x})$$

- would a cutoff of 0.55 be better?

Section 2

Model Evaluation

Confusion Matrix

Given any chosen cutoff c, we can form binary predictions for each observation by applying the cutoff to the fitted probabilities

$$\hat{y}_i = \begin{cases} 1 & \text{if } \hat{\pi}_i > c \\ 0 & \text{if } \hat{\pi}_i \leq c \end{cases}$$

The **confusion matrix**

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	# true negative (TN)	# false positive (FP)	N_0
$y = 1$	# false negative (FN)	# true positive (TP)	N_1

- the diagonal gives the count of the correctly predicted instances

$$accuracy = (\#TP + \#TN)/(N_0 + N_1)$$

$\Rightarrow$ an optimal cutoff can be chosen to minimize $\#FP + \#FN$ or (equivalently) maximize accuracy of the classifier. But not always ...

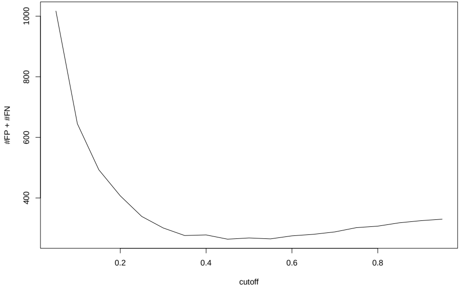
Heart Disease Data: Confusion Matrix

Table 2: cutoff 0.5 - accuracy=0.9732

	0	1
0	9627	40
1	228	105

Table 3: cutoff 0.35 - accuracy=0.9724

	0	1
0	9571	96
1	180	153



The smallest value corresponds to the cutoff 0.55. **Remember to check accuracy on a test set (out of sample)**

ROC curves [2]

Let's define two measures of performance

- Sensitivity = true positive rate = $\#TP/N_1$
  - sensitivity decreases as the cutoff increases
- Specificity = true negative rate = $\#TN/N_0$ = 1-FPR
  - specificity increases as the cutoff increases

Accuracy can be misleading if one class appears much more frequently than another, as in the Heart Disease dataset

- a model that just blindly predicts all patients to not develop heart disease would achieve an accuracy of 96.67%
- the accuracy would be even higher under more extreme imbalance (very rare disease)

⇒ To compare classifiers across all cutoffs, we look at the ROC (Receiver Operating Characteristics) curve

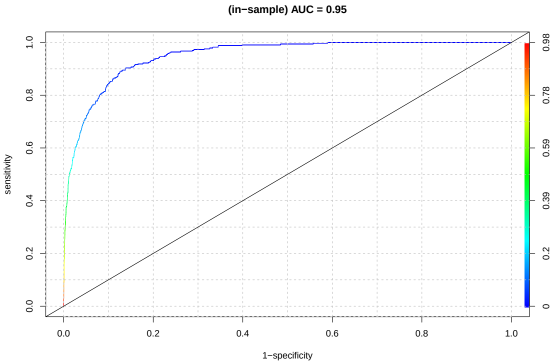
[2] Wojtek J. Krzanowski and David J. Hand, ROC Curves for Continuous Data (2009)

ROC curve

If the purpose of the logistic regression is to construct a predictive model, then a ROC curve is a useful graphical assessment of fit

- ROC curve plots the specificity against $1-$sensitivity for a range of cutoffs $\rightarrow$ takes the trade-off between FP and TP into account
- a coin-toss classifier $\equiv$ ROC curve is identity
- the area under the curve (AUC) is viewed as a measure of prediction accuracy
    - the larger the AUC, and hence the farther away the ROC curve is from the diagonal, the better the model performance
- computing AUC allows to quantitatively evaluate model performance
    - this could serve as a useful tool for model comparison as well
    - AUC$=1$ $\Rightarrow$ model able to perfectly distinguish between positive and negative
    - AUC$=0.5$ $\Rightarrow$ model is no better than a random classifier

Heart Disease: ROC curve



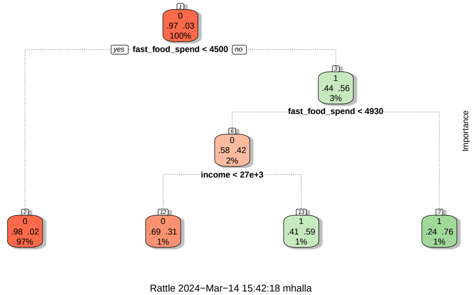
A good model has a high AUC, i.e., as often as possible a high sensitivity and specificity!

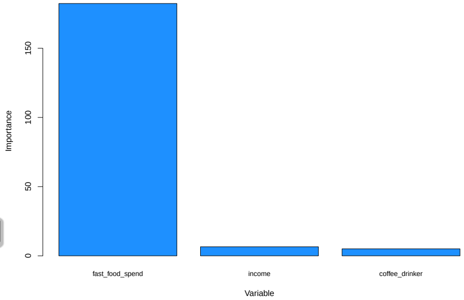
Note: AUC should be estimated out-of-sample or cross-validated (AUC= 0.9497 with 5 folds)

Heart Disease: Classification Tree [3]

```r
library(rpart)
library(rattle)

tree <- rpart(heart_disease ~., data=heart_data, method="class")
fancyRpartPlot(tree,palettes=c("Reds", "Greens"))
VI <- tree$variable.importance
barplot(VI, xlab="Variable", ylab="Importance", names.arg=names(VI),cex.names=0.8,
```





[3]See the MATH-517 lecture notes

Heart Disease: Classification Tree

What about prediction and accuracy?

```
ConfusionMatrix <- predict(tree, heart_data, type="class")
matrix          <- table(heart_data$heart_disease,ConfusionMatrix)
print(matrix)
```

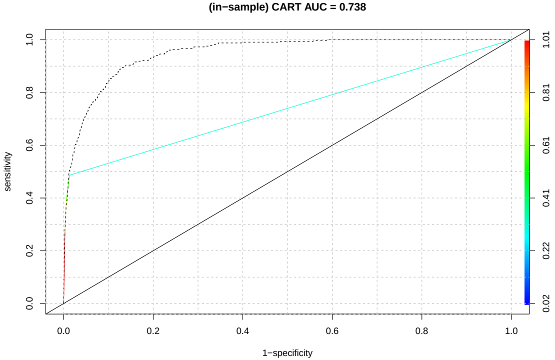
```
   ConfusionMatrix
       0    1
  0 9611   56
  1  203  130
```

```
accuracy <- sum(diag(matrix))/sum(matrix)
print(accuracy)
```

```
[1] 0.9741
```

Heart Disease: Classification Tree

Since classification is binary with decision trees, one can use predicted class probabilities to construct a ROC curve



Classification: Final Remarks

- A classifier assumes a model for the joint distribution of $(Y, \mathbf{X})$ and estimates it
  - Naive Bayes estimates a likelihood and a prior $(\Pr(\mathbf{X} \mid Y) \Pr(Y))$ based on assumptions of conditional independencies
  - Logistic regression estimates $\Pr(Y \mid \mathbf{X})$ parametrically
  - Classification trees estimate $\Pr(Y \mid \mathbf{X})$ non-parametrically
- Criteria for a good classifier
  - Accuracy (report AUC as it works under imbalance)
  - Runtime
  - Interpretability
  - Flexibility