Machine Learning for Business

Module 6: Logistic regression
Day 3, 13.00 – 16.00

Asst. Prof. Dr. Santitham Prom-on

Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

Module 6 Overview

- Background
- Generalized linear model
- Logistic regression

** Hands-on workshop: bank-data

Regression so far…

- At this point we have covered:
- Simple linear regression
  - Relationship between numerical response and a numerical or categorical predictor
- Multiple regression
  - Relationship between numerical response and multiple numerical and/or categorical predictors

What we haven't seen is what to do when the predictors are weird (nonlinear, complicated dependence structure, etc.) or when the response is weird (categorical, count data, etc.)

Categorical target

- Categorical target variable has the values in class
- This can be
  - Success/Fail
  - Yes/No
  - Churn/Not Churn
  - Normal/Default
  - Downward/Normal/Upward
    - Upward vs Not-Upward
    - Downward vs Not-Downward

Odds

- Odds are another way of quantifying the probability of classes or events, commonly used in gambling, medical (and logistic regression).

$$odds(E) = \frac{P(E)}{P(E^c)} = \frac{P(E)}{1 - P(E)}$$
$$= \frac{x/(x+y)}{y/(x+y)}$$

- The latter is if we are told that the odds of E is *x* to *y*.

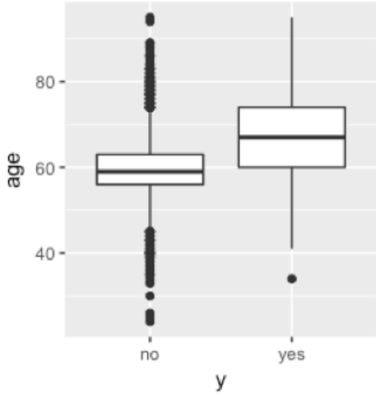
R: Bank data EDA, job vs y

```
bankData <- read.csv("bank-data.csv",sep=";")
table(bankData$job,bankData$y)
```

	no	yes
admin.	4540	631
blue-collar	9024	708
entrepreneur	1364	123
housemaid	1131	109
management	8157	1301
retired	1748	516
self-employed	1392	187
services	3785	369
student	669	269
technician	6757	840
unemployed	1101	202
unknown	254	34

R: Bank data
EDA, age vs y

```
bankData %>% filter(job == 'retired') %>%
   ggplot(mapping = aes(x = y, y = age)) + geom_boxplot()
```



Assess relationships of categorical variables

- It seems that there are relationships between categorical variables.
- How do we assess them?

Chi-square statistics

- Measures of association provide a means of summarizing the size of the association between two variables.
- One way to determine whether there is a statistical relationship between two variables is to use the chi square test for independence
- A cross classification table is used to obtain the expected number of cases under the assumption of no relationship between the two variables
- The value of the chi square statistic provides a test whether or not there is a statistical relationship between the variables in the cross classification table.

Chi-square: expectation

$$\chi^2 = \sum_{i=1}^{n} \frac{(\text{observed - expected})^2}{\text{expected}}$$

Weak relationship

Opinion	Male	Female	Total
Agree	65 (60.0)	25 (30.0)	90
Disagree	35 (40.0)	25 (20.0)	60
Total	100	50	150

$$\chi^2 = 0.417 + 0.833 + 0.625 + 1.250 = 3.125$$

$$df = 1$$

$$0.075 < \alpha < 0.10$$

Strong relationship

Opinion	Male	Female	Total
Agree	75 (66.7)	25 (33.3)	100
Disagree	25 (33.3)	25 (16.7)	50
Total	100	50	150

$$\chi^2 = 1.042 + 2.083 + 2.083 + 4.167 = 9.375$$

$$df = 1$$

$$0.001 < \alpha < 0.005$$

R: Chi-square (Cramer's V)

$$V = \sqrt{\frac{\chi^2}{nt}}$$

$$t = \min\left(r-1, c-1\right)$$

r	number of rows
c	number of columns
n	number of samples

R: Chi-square

```
library(tidyr)
bankData %>%
   group_by(marital,y) %>%
   summarise(n = n()) %>%
   spread(y, n) -> d
chisq.test(as.matrix(d[,-1]))
```

```
        Pearson's Chi-squared test

data:  as.matrix(d[, -1])
X-squared = 196.5, df = 2, p-value < 2.2e-16
```

Binomial distribution

- It seems clear that both age and job have an effect on the subscription, how do we come up with a model that will let us explore this relationship?
- Even if we set no to 0 and yes to 1, this isn't something we can transform our way out of - we need something more.
- One way to think about the problem - we can treat yes and no as successes and failures arising from a binomial distribution where the probability of a success is given by a transformation of a linear model of the predictors.

Generalized linear model

- It turns out that this is a very general way of addressing this type of problem in regression, and the resulting models are called generalized linear models (GLMs).
- Logistic regression is just one example of this type of model.

Generalized linear models

All generalized linear models have the following three characteristics:

1. A probability distribution describing the outcome variable
2. A linear model
   - $\eta = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$
3. A link function that relates the linear model to the parameter of the outcome distribution
   - $g(p) = \eta$ or $p = g^{-1}(\eta)$

Logistic regression

- Logistic regression is a GLM used to model a binary categorical variable using numerical and categorical predictors.
- We assume a binomial distribution produced the outcome variable and we therefore want to model p the probability of success for a given set of predictors.
- To finish specifying the Logistic model we just need to establish a reasonable link function that connects η to p.
- There are a variety of options but the most commonly used is the logit function.

Logit function

$$logit(p) = \log\left(\frac{p}{1-p}\right), \text{ for } 0 \le p \le 1$$

Logistic regression

- When the ordinal/numeric input associating with categorical dependent variable, logistic regression can be used
- Suitable when
  - Number of features is large, or
  - Number of observations is large
- Used in many models, including churn prediction

R: logistic regression

- In R we fit a GLM in the same was as a linear model except using *glm* instead of *lm* and we must also specify the type of GLM to fit using the family argument.

```
model <- glm(y ~ job + age, data = bankData, family = binomial)
summary(model)
```

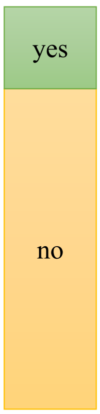
R: prediction with logistic regression

```
res <- predict(model, bankData, type="response")
```

```
         1          2          3          4          5          6          7          8 
0.14112047 0.11134643 0.08153215 0.07354416 0.11548506 0.13644982 0.13505438 0.08268467 
         9         10         11         12         13         14         15         16 
0.22682010 0.11117841 0.12232805 0.12015573 0.11286859 0.11372207 0.09136544 0.22474099 
        17         18         19         20         21         22         23         24 
0.12305963 0.07471030 0.22741662 0.08803596 0.07137455 0.14070908 0.07182634 0.08695072 
        25         26         27         28         29         30         31         32 
0.22150124 0.12287639 0.13725265 0.08398261 0.13866716 0.11000849 0.11355092 0.13927711 
        33         34         35         36         37         38         39         40 
0.12583664 0.07494556 0.13968499 0.11355092 0.07103743 0.11286859 0.12141884 0.12160022 
```

R: get predicted class

- Because in *y*, no is the first level. Thus the model will use no as a baseline and predict yes
- But the class is imbalanced, thus the probability of the first class will be for no, then for yes.
- We can convert the probability to class simply using ifelse




```
res_c <- factor(ifelse(res > 0.2, "yes","no"))
```

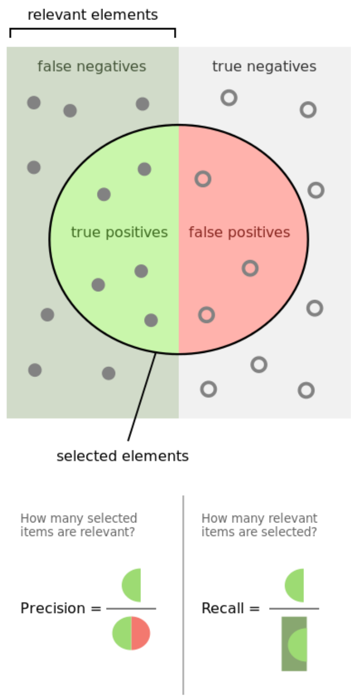
Precision and Recall

	Actual Positive (p)	Actual Negative (n)
The model says “Yes” = positive (y)	True positives	False positives
The model says “No” = not positive (n)	False negatives	True negatives

- Precision (Exactness) = the accuracy over the cases predicted to be positive, TP/(TP + FP)
- Recall (Completeness) = true positive rate = TP/(TP + FN)
- F-measure = the harmonic mean of precision and recall

  = the balance between recall and precision

  $= 2 . \frac{precision * recall}{precision + recall}$



R: confusion matrix

```
confusionMatrix(res_c,
  bankData$y,
  mode="prec_recall",
  positive="yes")
```

```
          Reference
Prediction    no   yes
       no  37505  4504
       yes  2417   785

               Accuracy : 0.8469
                 95% CI : (0.8436, 0.8502)
    No Information Rate : 0.883
    P-Value [Acc > NIR] : 1

                  Kappa : 0.106
 Mcnemar's Test P-Value : <2e-16

              Precision : 0.24516
                 Recall : 0.14842
                     F1 : 0.18490
```

R: logistic regression with all parameters

```
model_1 <- glm(y ~ ., bankData, family=binomial)
res_1 <- predict(model_1,bankData)
res_1c <- factor(ifelse(res_1 > 0.2, "yes","no"))
confusionMatrix(res_1c, bankData$y,
mode="prec_recall", positive="yes")
```

```
          Reference
Prediction    no   yes
       no  39054  3601
       yes   868  1688

               Accuracy : 0.9012
                 95% CI : (0.8984, 0.9039)
    No Information Rate : 0.883
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3833
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.66041
                 Recall : 0.31915
                     F1 : 0.43034
```

Issues

- Up until now, we use the training to test the model
- This will eventually lead to an overfitting model
- To avoid this we need to separate the data into training and testing sets
- Or we can sampling the training out of the dataset and train the model and use the whole set to test

R: Sampling and building model

```
bankData %>% sample_frac(0.1) -> bankData_train
model_2 <- glm(y ~ ., bankData_train, family=binomial)
res_2 <- predict(model_2,bankData)
res_1c <- factor(ifelse(res_1 > 0.2, "yes","no"))
confusionMatrix(res_1c, bankData$y, mode="prec_recall",
positive="yes")
```

```
            Reference
Prediction    no   yes
       no  39091  3657
       yes   831  1632

               Accuracy : 0.9007
                 95% CI : (0.8979, 0.9035)
    No Information Rate : 0.883
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3746
 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.66261
                 Recall : 0.30856
                     F1 : 0.42105
```

Activity

- Using the credit approval dataset
- Prepare the data for modeling
- Build a logistic regression model to predict the class (A16)

Thank you

Question?