ISLR

Chapter 2 - Statistical Learning:

2.1 - What is statistical Learning?

Imagine a client wants to increase sales - they cant increase sales, but they can control advertising expenditure.

2.1.1 - Why estimate F?

2.1.2 - How to estimate F?

2.1.3 - Trade off between prediction accuracy and model interpretability

2.1.4 - supervised vs unsupervised learning

2.1.5 - regression vs classification

2.2 - Assessing Model Accuracy

2.2.1 - measuring quality of fit

2.2.2 - Bias Variance trade off

2.2.3 - The classification setting

2.3 - Lab: introduction to R

2.3.1 - basic commands

2.3.1 - graphics

2.3.3 - indexing data

2.3.4 - loading data

2.3.5 - Additional Graphical and Numerical Summaries

2.4 - Exercises

Chapter 4 - Classification:

4.1 - An overview of classification

linear regression will have a quantitative y value. sometimes the y is qualitative (like yes/no etc). logistic regression, linear discriminant analysis and k nearest neighbours

4.2 - why not linear regression?

dont use linear regression cause for a 1/0 response, you get weird results (above 1 and below zero). so you can say "imagine you have a function that is related to probability, the regression equation will reflect that". There are many functions that relate a regression to a PD (probit, logit etc.).

4.3 - Logistic regression

In logistic regression use the logit functio

so y = b0 + b1x1, pd = exp(y)/(1+exp(y)),

so the regression equation becomes b0 + b1x1 = ln(pd/(1-pd))

4.3.1 - The logistic regression

use Maximum likelihood to estimate, not least squares

so if you get 

the probability of default if you have a balance of 1000 is:

-10.6513 + (.0055 \* 1000) = -5.1513 = y

this is the log odds, so exp(y)/(1+exp(y)) = probability =

4.3.2 - estimating the regression coefficients

Unlike linear regression where the sum of squares is minimized, in logistic regression, the MLE imaximised.

4.3.3 - making predictions

you have a regression equation - this will give you your LOG ODDS, so use the inverse logistic function (exp(y)/(1+exp(y)) to get the probability.

4.3.4 - multiple logistic regression

similar to multiple linear regression. still takes the same form y = b0 + b1x1 + b2x2 ...

but still falls in the form exp(y)/(1+exp(y)) = probability

learned abot confounding variables

how can you run a 1 variable logistic regression and get a negative coefficient for a variable, but when you run a multiple logistic regression, you end up with a positive (or vice versa)

suppose you are trying to model default and you have "Student" as a flag, along with balance and income.

in a 1 variable regression you end up with a positive coefficient for student (implying students are more risky) but in multiple you get a negative, its cause on average students default more, but at a given balance they default less than normal (cause students on average have a higher balance).

4.3.5 - logistic regression for >2 response classes

logistic regression has a way to expand the number of response classes - not important right now. probably caue it's outclased by linear discriminant analysis

4.4 - Linear discriminant analysis

logistic regression becomes unstable with multiple well defined classes for some reason

4.4.1 - usng bayes theorem for classification

linear discriminant analysis goes from finding P(Y|X) instead they go P(X|Y). i.e. they flip it for some reason then use bayes theorem to flip and and get Y

4.4.2 - linear discriminant analysis for p = 1

p = 1 means theres only one category.

1. assume that x is normally distributed

2. do some algebra to figure out decision boundary

4.4.3 - Linear discriminant analysis

false positive false negative, can change threshold to change error rate of false positive and false negative. different industries might want different rates of false positives and false negatives. e.g. defaults in credit cards are risky, so you can set it so that you decrease the rates of a false negative (i.e. you are ok with false positives, you wouldnt lend to those people and thats ok - you're trying to avoid defaults).

5 - Resampling methods

cross validation and bootstrap.

5.1 cross validation

5.1.1 validation set approach randomly divide dataset into 2 groups training set and validation set/holdout set. train model on training set and validate on holdout/validation set.

5.2.1 leave out one cross validation similar to validation set, but instead of 50/50 or 80/20 split, leave just 1 value out. do this repeatedly n times then average the errors

has less bias than validation, because the training set has almost all the information in it. potentially expensive to implement.

5.1.3 kfold cross validation - divide the total set into k groups (folds), first fold = validation set and remaining k-1 groups are used to train the model. figure 5.5 shows a 5 fold. This is almost identical to the LOOCV except it has multiple leave outs.

5.1.4 bias-variance trade off for k-fold cross-validation

k fold is more accurate than LOOCV because of bias variance tradeoff.

5.2 Bootstrap

bootstrap can be used to estimate the error of an estimator. repeatedly sample the dataset. sample with replacement.

5.3 cross validation and the bootstrap.

5.3.1 the validation set approach

6 Linear model selection and regularization

linear regression can be improved by not using least squares and usng another framework - why would we do this? because other fitting methods might yield better prediction accuracy and model interpretability

page 205