Exercises 7 · Logistic regression and time series

To turn in

Due Tuesday, April 14, 2015

(1) Energy consumption

Power companies are always trying to improve their statistical models for daily peak electricity demand. These models provide forecasts that help utilities in their economic and infrastructural planning. Underestimating demand can lead to low capacity, which results in poor power service, rolling brownouts, or even occasional blackouts. Overestimating demand, on the other hand, can lead to the construction of a plant that may not be needed for years to come. Many utilities do not earn enough to be able to cover such a plant without the revenues to pay for it.

In "peakdemand.csv", you have been given data collected by Duke Energy, the major provider of electricity in Raleigh, North Carolina. The data consist of daily observations of peak electricity demand in Raleigh, measured in megawatts, for four years: January 1st, 2006 through December 31, 2009. Accounting for the leap year in 2008, this makes for 1461 daily observations. In addition to the year, month, and day of the month for each observation, you also have the following variables:
PeakDemand: Actual peak demand for electricity in megawatts.
DailyTemp: Average temperature reading, in degrees Fahrenheit, over the course of the entire day.
Sat: coded 1 if that day was a Saturday, and 0 otherwise.
Sun: coded 1 if that day was a Sunday, and 0 otherwise.

Use this information to build a statistical a model for daily peak electricity demand. Try to find the simplest model that explains the data well. Write a short report explaining your analysis and your proposed model. Include only those pictures that you believe are necessary in making your case.

(2) Spam filtering

In this problem, you will use logistic regression to construct a basic e-mail spam filter, such as the kind that keeps all sorts of annoying stuff out of your inbox. On the course website you will find "spam-fit.csv" (with 3000 observations). Each row contains information about a single e-mail message, and each file has the same ten columns:

Columns 1–6 record the relative frequencies of particular words or strings: "remove," "order," "free," "meeting," "re:," and "edu." These are listed as percentages. For example, an entry of 1 under the "free" column means that 1% of the words in the e-mail were the word "free."

Columns 7 and 8 record the relative frequencies of two punctuation marks: the semicolon (;) and the exclamation mark (!). These are also listed as percentages, as above; an entry of 1 under the semicolon column means that 1% of characters in the e-mail were semicolons.

Column 9 is the average length of consecutive strings of capital letters. For example, an e-mail that contained only the phrase "BUY OUR PRODUCT!" would have an average run length of 13 capital letters. (Punctuation marks and spaces do not count as interruptions.) A normal sentence, such as "We should think about buying their product," has an average run length of 1 capital letter (the first letter of the sentence).

Column 10 indicates whether the e-mail was spam/commercial e-mail (1) or not (0).

(A) Using only the data in "spamfit.csv", fit a logistic regression model to predict whether an e-mail is spam. Comment briefly (1-2 sentences) on why you think the coefficients have the values they do. (For example, pick one with a large positive coefficient and explain why; then pick one with a large negative coefficient and explain why.)

(B) Use your fitted regression equation to compute the probability that each e-mail in "spamfit.csv" is a spam message. These are your in-sample fitted values. Suppose that your e-mail program filters all messages that the model judges to be spam with greater than 50% probability. There are two kinds of errors one could make using such a threshold: a false positive, where one wrongly declares a non-spam message to be spam; and a false negative, where one wrongly allows a spam message to pass through the filter. Compute the following three error rates that characterize your model's performance on the "spam-fit" data:

(1) The false positive rate (FPR), or the fraction of non-spam messages (true nulls) that were flagged as spam.

(2) The false negative rate (FNR), or the fraction of spam messages that were not flagged as spam.

(3) The false discovery rate (FDR), or the fraction of false positives among all messages flagged as spam. That is, if K is the raw number of false positives, and M is the number of total positives (i.e. messages flagged as spam), the FDR is the ratio K/M.

Note: if you want to check whether the values in some vector x (like model predictions!) exceed a threshold, you can use the command `x > 0.5`. This will return a TRUE for every entry in x that exceeds 0.5, and a FALSE for every entry that doesn't.

(C) Now use the same fitted regression equation to compute the probability that each e-mail in "spamtest.csv" is a spam message. (Do not fit a new regression model to spamtest; rather, use the model you estimated from spamfit to generate your predictions for spamtest.) Again, suppose your e-mail program filters all messages that the model judges to be spam with greater than 50% probability. Compute the same three error rates from Part B. (Remember the `predict` function.)