36-350: Data Mining

Lab 13

Date: November 22, 2002 **Due: end of lab**

---

1 Introduction

This lab teaches you how to construct a linear model with interactions among categorical variables.

There are 4 questions. For each one, submit your commands and a response from R demonstrating that they work. (Only hand in commands relevant to the question.) To submit a plot, click on the plot window and select

```
File -> Save as -> Postscript...
```

This saves the plot to a file which can be printed, incorporated into a Word document, or mailed to us as an attachment.

It will be useful to make a notepad file containing your R code, which you can edit ahead of time and then cut and paste into the R command window.

2 Starting R

Start R as in lab 1. On the class web page, go to "computer labs" and download the files for lab 13 into your work folder. Read the special functions into your running R application via the commands

```
source("lab13.r")
```

If this fails, check that the files were downloaded correctly.

3 The data

Safety is an important issue in purchasing a car, and every year the National Transportation Safety Board performs crash tests on new models. The dataset used in this lab is the result of crash tests on 274 cars, ranging over 1987–1992 model years. For each test, an instrumented dummy was seated in the car and the car was crashed into a barrier at 35mph. The amount of head injury to the dummy was stored in the variable `Head`, with other variables describing attributes of the car. The variables are:

Head	Head injury to the dummy
D.P	Dummy in the `Driver` or `Passenger` seat
Protection	Kind of protection:
	Driver and passenger airbags (`d&p airbags`)
	Driver-side airbag (`d airbag`)
	Motorized belts, Passive belts, Manual belts
Doors	Number of doors on the car (2 or 4)
Size	Car weight/size category:
	small, light, compact, midsize, heavy, SUV

All of the variables except `Head` are categorical.

Historical context: Around this time period, automatic protection was required by law, but auto makers strongly preferred automatic belts over airbags, despite mounting evidence that automatic belts were ineffective and sometimes worse than manual belts. Congress took up the issue, and eventually ruled in 1991 that all new cars starting in 1998 must have driver and passenger airbags.

Load this data via

```
load("Crash.rda")
```

This defines a matrix called `x`. The head injury variable has already been transformed with a logarithm to make its distribution symmetric.

4 Constructing the model

The problem is to construct an accurate and simple model for head injury as a function of the car variables. What makes this complicated is that all of the variables interact. To capture the interactions, our strategy is add new columns to `x` which represent cross terms between the original variables.

The basic tool for constructing a model is `lm`:

```
fit <- lm(Head~.,x)
```

This creates a linear model to predict `Head` from all other columns in `x`, including any new ones you may add.

Two important qualities of this model are the number of coefficients and the multiple R-squared, which are available by running

```
summary(fit)
length(coef(fit))
```

Note that the number of coefficients is much larger than 4, the number of predictor columns. This is because `lm` automatically adopts an indicator code for categorical variables. There are indicators for all categories except the first.

5 Assessing predictors

The quality of the predictors in the current model, as well as any future model you construct, can be determined from a partial residual plot. This is the command, same as in lab 9:

```
predict.plot(fit,partial=T)
```

For each category, the mean partial residual is plotted, along with a "1.64" error bar. A partial residual in this case means excluding the entire variable—all of its categories—from the model.

If the error bar does not include zero, the category has a significant effect on injury. If the error bars for two categories do not overlap, their effects on injury are significantly different (at the 95% level).

Note that `predict.plot` is more reliable than the p-values from `summary` for these kinds of judgements. This is because `summary` considers the significance of each category when all other categories are included in the model.

> **Question 1:** According to the linear model with no interactions, which protections make a difference to head injury? Which protections are significantly different from each other?

6 Adding interaction terms

An interaction term between categorical variables is simply an indicator for a combination of categories. For example, suppose the head injury is unusually high for two-door SUVs. The interaction term for this would be `Yes` when `Doors=2` and `Size=SUV`, and `No` otherwise. This term could be added to `x` as follows:

```
x[,"door2.suv"] <- factor.logical((x[,"Doors"]=="2") & (x[,"Size"]=="SUV"))
```

By fitting a new model and plotting partial residuals, you could determine if this term is useful. (It isn't.)

There are obviously many possible interaction terms. You need a plot which helps find the good ones. Partial residuals are good for dropping predictors. For adding interaction terms, you want to look at ordinary residuals. The first step is to create a new data matrix where `Head` is replaced by its residual from the current model. This is accomplished by

```
r <- residual.frame(fit,x)
```

Because the predictors are categorical, contour plots produced by `interact.plot` are not useful. Instead you use slice plots. Recall the command for a slice plot from lab 9:

```
predict.plot(Head ~ Protection | D.P, r)
```

It will print out "columns are ..." to help you when the column names cannot fit on the plot. It is possible to look at multiple slice plots at once, using the following syntax:

```
predict.plot(Head ~ Protection + Size + Doors | D.P, r)
```

If there were no interactions, the curves in these slice plots would all be flat. The places where the curves fail to be flat are interactions. (Notice the similarity to finding dependencies in a contingency table.) In particular, if there are *any* two categories on the same curve whose partial residuals are significantly different, then there is an interaction. The interaction is with the slicing variable (i.e. the color of the curve).

You should find a few interactions between `D.P` and the other variables. One by one, create an interaction term as above, fit a new model, and regenerate the plot. That interaction should now be gone (though others may have appeared).

Question 2: Find and eliminate all significant interactions between `D.P` and the other variables. To verify that your interaction terms are useful, make a partial residual plot as in the previous section. Turn in your code and your plot.

Question 3: Now find and eliminate all remaining interactions between `Doors` and the other variables. Make a new partial residual plot. Your R^2 should now be over 0.3, using at most 16 coefficients.

Question 4: Now remove any remaining irrelevant variables in the model. This can be done by removing the variable from `x`, then re-fitting the model:

```
x <- not(x,"Variable")
```

Turn in the summary of the final model and keep a copy for the homework.