36-350: Data Mining

Lab 10
Date: October 31, 2003 **Due: end of lab**

Interspersed throughout this lab are questions that you will have to answer at check-off.

1. Download the files for this lab from the course web page to the desktop:

   `http://www.stat.cmu.edu/~minka/courses/36-350/lab/`

2. Open a Word or Notepad document to record your work.

Start R

3. `Start -> All Programs -> Class software -> R 1.7.0`

4. Load the special functions for this lab:

   `File -> Source R code...`

   Browse to the desktop and pick `lab9.r` (it may have been renamed to `lab9.r.txt` when you downloaded it). Another window will immediately pop up for you to pick the `mining.zip` file you downloaded.

The dataset

5. The dataset and startup is the same as lab 9. Transform `Sold.4` and standardize all variables to mean zero and variance one.

Selecting interaction terms

6. Using `lm` from lab 9 and the functions below, make a linear model to predict `Sold.4` from all product prices.

7. Use `step.up` to automatically select important interaction terms. (Unimportant prices will be automatically dropped too.) `summary` shows the final model. There should be six bilinear terms. *Which bilinear term has the greatest weight in the model? Which has the least weight?*

8. Plot each term against partial residuals (as in lab 9). *Which bilinear term is least important? Does any bilinear term have greater importance than either of its individual predictors?*

Visualization

9. Make a contour plot matrix which shows the contribution of each interaction term to the model. Keep a copy of this and all following plots for the homework. *Find the 'strongest' and 'weakest' interactions that you identified earlier. Do the contours agree with that ranking?*

10. Make slice plots of the top 4 interactions. They should show the type of each interaction. (As a check, the contour plot of each interaction should be the same as in the plot above.) For each, you have to decide which is the better variable to slice on.

11. You can now get checked off.

Automatic formulas If `x` is a matrix, then formulas for use with `lm` can be generated via

```
formula(x)
expand.cross(x)
```

The first type has the last column as response and the rest as predictors. The second type also has all cross terms.

Adding bilinear terms If `fit` is a linear model,

```
fit2 = step.up(fit)
```

will try adding all possible bilinear terms and use AIC to select the best.

Contour plot matrix `interact.plot` is invoked similarly to `predict.plot`. If `fit` is a model and `x` is a matrix:

```
interact.plot(x)
interact.plot(fit)
interact.plot(fit,partial=T)
```

The first makes all pairwise contour plots of the response, the second plots residuals, the third plots partial residuals.

Slice plots To extract partial residuals for an interaction term:

```
r = partial.residual.frame(fit,"Price.8*Price.11")
```

This sets `r` to a matrix with three columns: `Price.8`, `Price.11`, and `Sold.4` (`Sold.4` is the partial residuals for a model with `Price.8` and `Price.11` completely excluded). Notice that you put `*` between the variable names, not `:`. Plots are made in the usual way:

```
color.plot(smooth(Sold.4~Price.8+Price.11,r),n=8)
slices(Sold.4~Price.8|Price.11,r)
predict.plot(Sold.4~Price.8|Price.11,r,n=4)
```