IST 597: Homework 2 Solutions

Justin Silverman

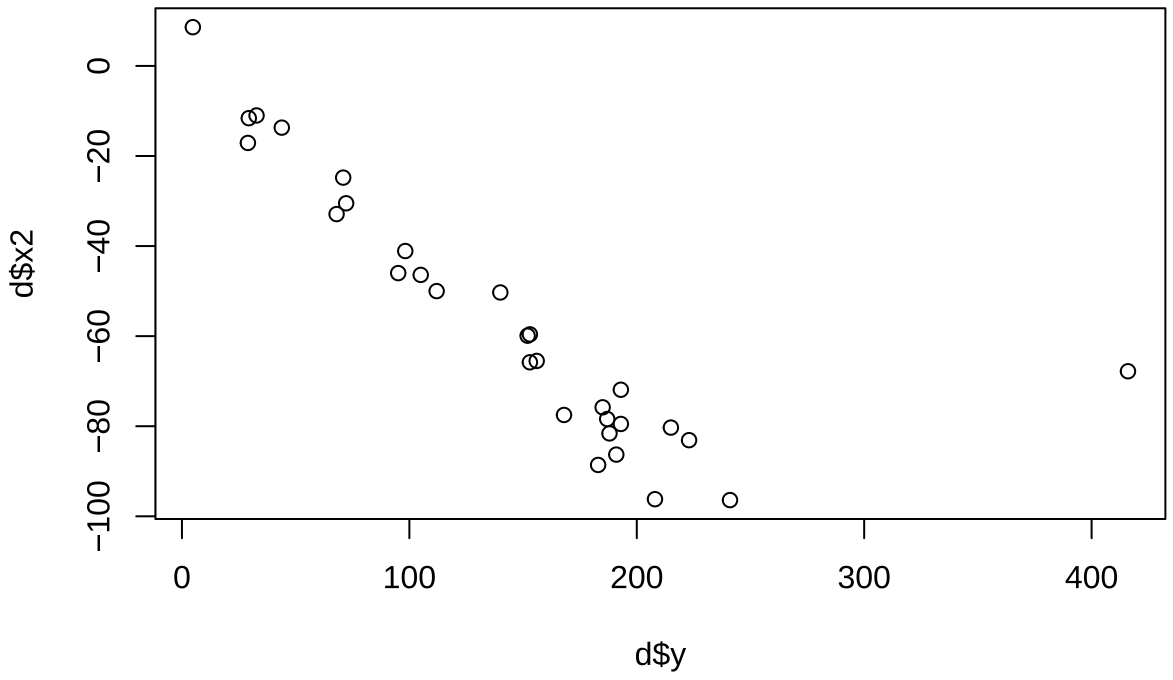
Problem 1

The key to this problem is the very large outlier in the training data. In essence, something is wrong with that observation (the 25th observation), or its a very odd observation that is not representative of the others. It has a huge effect on $\hat{\beta}$ and will negatively effect the generalization of your predictive model.

Its generally easy to spot outliers when working with univariate y and univariate x, but it can be much harder with multivariate data. That doesn't mean you shouldn't try.

That said, even simple plots of the training data here can give you some sense that there may be an outlier. I made this data to be somewhat obvious

```r
d <- read.csv("data_train_hw2_problem1.csv")
plot(d$y, d$x2)
```



Residual plots are incredibly useful. They can reveal key areas where your model doesn't fit the data (e.g., if you have outliers or heteroskedasticity). A nice thing about this particular residual plot (y vs. $\hat{y}$) is that it is a simple two dimensional visualization even if you have thousands of predictors.

```r
(fit <- lm(y ~ x1 + x2 + x3-1, data = d))
```

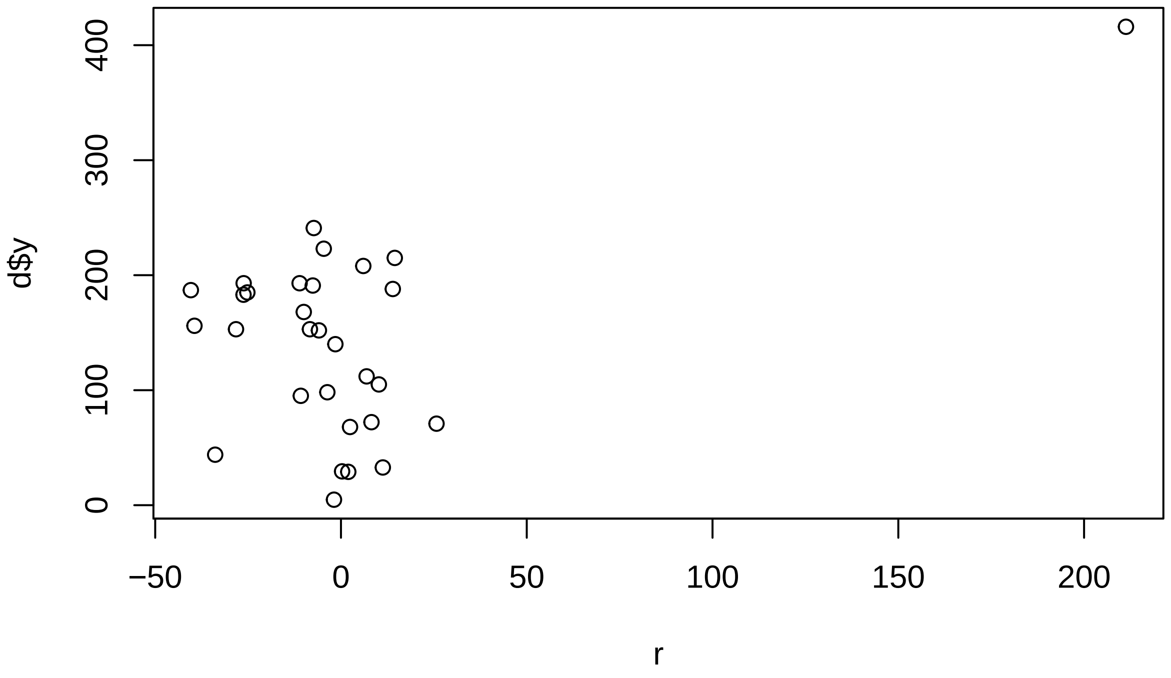
```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 - 1, data = d)
##
## Coefficients:
##     x1      x2      x3
##  1.689  -2.203  39.299
```

```r
r <- residuals(fit)

# Training set MSE (which is the same as \hat \sigma^2)
mean(r^2)
```

```
## [1] 1794.453
```

```r
# Now visualize the residuals.
plot(r, d$y)
```



Potential solutions here include: - remove the outlier from the training data in this case this would be a perfect solution as the simulated outlier does not come from the model but is probably more likely to be a measurement error / mistake. - use methods robust to outliers (like robust regression) - use cross-validation on your training dataset! This could have shown you that your model did not generalize well, well before you broke into the hold-out testing data. This is generally a good procedure to get into the habit of doing. Unfortunately in removing the data point and refitting the model, we can't be sure we aren't now just over-fitting to the testing data. What if there really are many outliers in production data? Then we may want methods like robust regression that just handle / flag those naturally.

Another lesson here: Always consider that your dataset likely contains some errors. Data is not perfect, it will serve you well to ask questions regarding how the data is collected.

Problem 2

$X^T X$ (assuming X is column centered [each column has the mean value subtracted so the column mean is zero]) is the covariance between your features. A zero eigenvalue signals co-linearity in the features.

```r
d <- read.csv("data_train_hw2_problem2.csv")
X <- as.matrix(d[,2:4])
eigen(t(X) %*% X)
```

```
## eigen() decomposition
## $values
## [1] 7.070820e+01 1.599727e+01 7.706111e-15
##
## $vectors
##              [,1]         [,2]          [,3]
## [1,] -0.413792610 -0.002924534  9.103665e-01
## [2,] -0.910343741 -0.006433976 -4.138029e-01
## [3,] -0.007067457  0.999975025 -1.301043e-18
```

Note the third eigenvalue is (essentially) zero – up to numerical precision. This dataset has perfect co-linearity, in fact $x_2 = 2.2 * x_1$. That is x_1 and x_2 have identical information. OLS breaks down in this setting. Some linear regression software uses the pseudo-inverse rather than the inverse of the matrix $X^T X$, this is a strong assumption as there are many possible values of β that equally well explain the data – but the use of the pseudo-inverse essentially just picks one arbitrarily. Same general idea with optimization approaches to this problem: the software may stop at something it calls an optimal estimate for β, but that optima won't be unique and will be just one arbitrary choice. Beware of interpreting regression coefficients in this setting.

Problem 3

There is no trick / lesson here. Just a pure applied machine learning question. There are tons of ways to approach this problem. Students are scored here purely based on test set error.