Data Science & Strategic Pricing

Instructor: Jacob LaRiviere, Director of Economics and Data Science at Amazon

Email: lghhager@uw.edu (TA)

Course Assignments & Reading

Course assignments should be knitted Rmarkdown files and turned in at the start of class on Canvas unless otherwise noted. Feel free to work in groups but everyone is required to turn in their own work with answers written in your own words. Keep in mind that in most cases a good answer is one precise sentence; quality is heavily favored over quantity. This will be graded on a full credit, half credit and no credit basis. All work must be typed. At the beginning of each class the professor will lead a discussion around these questions. Students will be called on, potentially at random, to add their insight. This part of class will contribute heavily to your course participation grade.

Week 4, due October 23

Assignment to be turned in. Please turn in your Rmarkdown output and answers on Canvas

1) Take a model that includes brand, feat, log(price), their interactions, lagged price, and demographics, and fit a LASSO using glmnet which is a workhorse R package for LASSO, Ridge and Elastic Nets.
    a. First remember to install the glmnet package and library to your R session.
    b. Remember to estimate a LASSO you must pass glmnet a matrix of data for candidate features and a vector as candidate outcomes:
    ```
    set.seed(720)
    lasso_v1 <- cv.glmnet(x, y, alpha=1)
    lasso_v1_cv <- glmnet(x, y, alpha=1)
    ```
    As an alternative to defining products in a dataframe, turning that into a matrix then passing that matrix to glmnet, you can cut out the middle man with this.
    ```
    X <- model.matrix(formula, df_RHS)
    # Note that df_RHS is a dataframe of just the features you'd like
    # to use as predictors of Y (this should include lagged price)
    # Now pass X to glmnet; documentation
    ```

    In addition to the variables in the original dataframe, try to create tons of new features that you think could plausibly be predictive of quantity sold. This could include lagged prices, interactions of several features, etc.
    c. Investigate the coefficients of the cross validated LASSO model. Code from class is here:
    ```
    x <- as.matrix(oj_cross[ ,5:17])
    y <- as.numeric(as.matrix(oj_cross[ ,4]))
    set.seed(720)
    #lasso_v1 <- cv.glmnet(x, y, alpha=1)
    lasso_v1 <- glmnet(x, y, alpha=1)

    #Results
    plot(lasso_v1)
    coef(lasso_v1, s=lasso_v1$lambda.min)

    # Now ready for cross validation version of the object
    cvfit <- cv.glmnet(x, y, alpha=1)
    #Results
    ```

```
plot(cvfit)
cvfit$lambda.min
log(cvfit$lambda.min)
coef(cvfit, s = "lambda.min")
```

Which are the parameters the cross validated LASSO model kicks out of the model? What is the ratio of number of features to number of observations? How might that relate to overfitting from "sampling error"?

d. Can you look that the glmnet objects and figure out what the out of sample (e.g., test set) average MSE was with the cross validated LASSO model relative to the model in 1.c?
e. What is the advantage of using LASSO for choosing model complexity as opposed to using your intuition as an economist?
    i. In what part of this process did you use your intuition as an economist? (*HINT: what's in the X matrix?)*

2) Now estimate the model with only the variables selected with the LASSO procedure but with OLS to avoid attenuation bias in the coefficients (similar to this paper).
    a. Let's return to the orange juice assignment and get very precise about how to interpret coefficients. What is the predicted elasticity in the following cases?
        i. For Dominicks when the lagged price is $1 (NOTE: did you interact lagged price with current period price?) If not, does lagged price impact the *elasticity* this period or *log move* this period.
        ii. For Tropicana
        iii. For Tropicana when its featured
        iv. What is the 95% confidence intervals for Tropicana
    b. Which product has the most elastic demand?
        i. Should that product have the highest markup over costs or lowest markup over costs? Why?
3) Go back to using logmove and log(price).
    a. Estimate a 3x3 matrix own price and cross price elasticities for Dominicks, Minute Maid, and Tropicana using only the current week's prices. Be sure to estimate separate models for sales of Dominicks, MM and Tropicana (e.g., you'll run three separate regressions with the same RHS variables but different LHS variables). It doesn't need to be overly complicated, but make sure there is an interpretable elasticity estimate. NOTE: This will require three different regressions & add in socio demographic controls for each store.
    b. Do the same but add in interactions for whether or not each brand is featured.
        i. How do the estimates change?
        ii. What product's sales suffer the most when Minute Maid is both featured and lowers its price?
    c. Which two products are the most competitive with each other?
        i. How did you infer that looking at the cross price elasticity?
        ii. What do you expect that to mean about the correlation of the prices of those two products? Would they be more correlated or less correlated than the price of other pairs of products?
4) Create a sales weighted price for orange juice by store.
    a. You'll first need to create actual sales (call it "Q") instead of log sales for the weighting and put it into your dataframe.
    b. You can use the weighted.mean() function for each store-week combination in the `dplyr` library. If works like this:

```
Df1 <- ddply(dataframe, c('var1','var2'),function(x) c(weighted_mean
= weighted.mean(x$price,x$Q)))
```

Here `'var1','var2'` are the two identifiers for the variables to create a weighted average by (store and week in our case), the function takes as an input "x" (which is the dataframe specified beforehand) then creates weighted_mean of `x$price` weighted by `x$Q`. You'll then need to merge this back in to the original dataframe. You can also calculate the weighted average manually.

5) Now use `oj$weighted_price` as the LHS variable in a regression tree to predict differences in sales weight prices with store demographics as RHS variables. Note that you'll only need to do for a single brand since weighted price and sociodemographic variables are identical across brands within a store.
    a. There are a couple libraries you'll need which you'll see in the lecture notes (`rpart, maptree, etc.`)
    b. There are two main pieces of code:

```
dataToPass<-
oj[,c("weighted_mean","AGE60","EDUC","ETHNIC","INCOME","HHLARGE","WORKWOM","HVAL150","
SSTRDIST","SSTRVOL","CPDIST5","CPWVOL5")]
#The above creates a dataframe from the existing one (with weighted mean merged back
in) which will then be passed into rpart (tree partitioning algorithm).

fit<-rpart(as.formula(weighted_mean ~ .),data=dataToPass,method="anova",cp=0.007)
#This is the code which will fit the tree.
```

    c. Play around with a couple different complexity parameters to get a feel for the data
        `draw.tree(fit) #This draws the tree`
    d. Choose three different leaves to group stores into based upon what explains sales weighted price.
        i. Assign each store to one of these leaves (we used this code previously).
            `dataToPass$leaf = fit$where #This assigns leaves to observations.`
6) Estimate the own price elasticities for each one of the store buckets/leaves using the preferred specification:

```
reg_int <- glm(logmove~log(price)*brand*feat,
data=oj_leaf_L)
```

    a. Now estimate cross price elasticities jointly with own price elasticities. This means you must create a dataframe which has the prices of all types of OJ at the store. (e.g., you should be able to use the `Trop_Cross` code you've used previously.
    b. You'll also have to run 3 separate regressions for each leaf for a total of nine regressions.

```
reg_int <- glm(logmove_D~log(price_D)*feat +
log(price_T)*feat + log(price_MM)*feat, data=oj_leaf_L_D)
```

    In this example, we are investigating the own and cross price elasticities for Dominick's brand (D) within leaf L.
        i. Save the coefficients for each leaf in a 3x3 matrix. The diagonals will be own price elasticities and the off diagonals will be cross price elasticities.
        ii. There will be a unique 3x3 matrix for each leaf.

iii. The 3x3 matrices WON'T be upper triangular because we're estimating three unique regressions for each leaf.

c. Comment on any differences between own and cross price elasticities by leaf.

7) Now let's use the elasticities to think about pricing differentials.
   a. In the leaf with the highest own-price elasticities, what should the markups be relative to the other leafs?
   b. How do cross-price elasticities vary with the highest versus lowest own price elasticity leafs?
      i. What does this imply about differences in markups within high versus low elasticity stores across brands?
      ii. Can you say anything about what this means for the timing of sales? Should they occur at the same or different times across stores?