NYC DATA SCIENCE ACADEMY

NYC Data Science Bootcamp Fall 2015

Unsupervised Machine Learning

Question #1: Principal Component Analysis

Load the `heptathlon` dataset from the `HSAUR` library into your workspace. A heptathlon is a combined track and field event-based contest for women. This dataset contains scores on each event for the 1988 olympic heptathlon competition held in Seoul.

1. Create a scatterplot matrix of all variables in the dataset. Briefly comment on the nature of the data.
2. It will help to have all event scores going in the "same direction" (i.e., a higher event score implies a better performance, and a lower event score implies a worse performance). To do so, transform the hurdle and running variables by subtracting the original scores for each heptathlete from the maximum score of each of those variables.
3. Create a scatterplot matrix of all the event score variables in the "same direction." Briefly comment on the nature of the data.
4. Create a scree-plot of your newly created dataset that doesn't include the `score` variable. From this plot, describe how you determine the number of principal components to extract in three different ways.
5. Extract the appropriate number of principal components from your dataset that does not include the `score` variable and save this object.
6. What is the variance of the each of your extracted principal components?
7. How much variability in the original dataset is captured by each of your extracted principal components?
8. Create a plot of the principal component loadings against each other.
9. Use the object you created in part 5 and the plot in part 8 to help construct interpretations for each principal component vector.
10. Create a scatterplot of each of the competitor's results projected onto the reduced dimensions.

11. Comment on any observations that appear to be outliers. Who are these competitors and why do they appear to be outliers?

Question #2: Ridge Regression

Read in the `08prostate.txt` dataset into your workspace. This dataset comes from a study by Stamey et a. (1989) of prostate cancer, measuring the correlation between the level of a prostate-specific antigen and some covariates. The included variables are the log-cancer volume, log-prostate weight, age of patient, log-amount of benign hyperplasia, seminal vesicle invasion, log-capsular penetration, Gleason score, and percent of Gleason scores 4 or 5; the response variable is the log-psa.

1. Create a training set of approximately 80% of your data and a test set of approximately 20% of your data (**NB:** Use `set.seed(0)` so your results will be reproducible.)
2. Fit a slew of ridge regression models **on your training data** by checking across a wide range of lambda values. Save the coefficients of these models in an object.
3. Plot the coefficients of these models and comment on the shrinkage.
4. Perform 10-fold cross-validation **on your training data** and save the output as an object. Once again, use `set.seed(0)`. (**NB:** You can manually define the values of lambda to as you did in part 2).
5. Create and interpret a plot associated with the 10-fold cross-validation completed in part 4.
6. What is the best lambda?
7. What is the **test** MSE associated with this lambda value?
8. Refit the ridge regression using the best lambda **using every observation in your original dataset**. Briefly comment on the coefficient estimates.
9. What is the overall MSE for the model you fit in part 8? How does this compare to the MSE you found in part 7?

Question #3: Lasso Regression

Continue using the `08prostate.txt` dataset you already loaded into your workspace.

1. Repeat the entire analysis performed in question #2, except use the method of lasso regression instead.
2. Compare and contrast your ultimate ridge and lasso models. Which would you choose to use? Why?

Question #4: K-Means

Read in the `08protein.txt` dataset into your workspace. This dataset contains protein consumption information from 1973 on nine different food groups across 25 different European countries.

1. Use the following commands to read the data into your workspace appropriately and scale the variables:

```
protein = read.table("08protein.txt", sep = "\t", header = TRUE)

protein.scaled = as.data.frame(scale(protein[, -1]))

rownames(protein.scaled) = protein$Country
```

2. Create and interpret a scree-plot for the within-cluster variance for various values of K used in the K-means algorithm.
    a. Why might this graph indicate that K-means is not truly appropriate to model the data?
3. Create and store 5 different K-means solutions that run the algorithm only 1 time each. (**NB:** Use `set.seed(0)` so your results will be reproducible.)
4. Create and store 1 K-means solution that was selected from running the algorithm 100 separate times. (**NB:** Use `set.seed(0)` so your results will be reproducible.)
5. Plot the 6 different solutions from part 3 and 4 with:
    a. `Cereals` on the x-axis.
    b. `RedMeat` on the y-axis.
    c. Colors for the different cluster assignments.

    d. Labels for the total within-cluster variances.
6. Plot the solution from part 4 with:
    a. `Cereals` on the x-axis.
    b. `RedMeat` on the y-axis.
    c. A label for the total within-cluster variance.
    d. Points for the centroids of each cluster.
    e. A horizontal line at 0.
    f. A vertical line at 0.
    g. Text listing the country for each observation in your dataset (instead of points), colored by the different cluster assignments. **Hint:** Use `type = "n"` when creating the `plot()`. Then, use the `text()` function in tandem with the `rownames()` function.
7. Interpret the clustering solution based on the graph you created in part 6.

Question #5: Hierarchical Clustering

Continue using the `08protein.txt` dataset you already loaded into your workspace.

1. Calculate and store pairwise distances for each observation in the dataset.
2. Fit hierarchical clustering solutions using single, complete, and average linkage.
3. Visualize the dendrograms created in part 2.
    a. Give an argument as to why single linkage might not be good to use.
    b. Give an argument as to why complete linkage might be good to use.
4. Cut your complete linkage tree into 2 groups.
    a. Visualize the solution overlaid on top of the dendrogram.
    b. Interpret the clusters by aggregating across the median.
5. Cut your complete linkage tree into 5 groups.
    a. Visualize the solution overlaid on top of the dendrogram.
    b. Interpret the clusters by aggregating across the median.