Algorithmic Learning Theory
MSIS 26:711:685
Homework 4

Instructor: Farid Alizadeh
Due Date: Friday December 13, 2019, at 11:50PM

last updated on December 8, 2019

Instructions: Please read carefully

Please answer the following questions in **electronic** form and upload and submit your files to Sakai Assignment site, before the due date. Make sure to click on the **submit** button.

For this homework you should submit **three files** for each of questions 1, 2 and 3 as an R (or Python) scripts. For each part of questions 1, 2 and 3 have the necessary R/Python code along with answers to questions in the form of an output. For example, here is a sample R code for homework answers:

```
# Question 1)
print("Question 1a)\n")
knnModel <- knn (y ~ x, data = someData)
print(summary(knnModel))
print("Answer to Q1a):\n")
print("error rate is ...\n")
readline("Hit Enter to continue\n")
#Question 1b)
. . . .
```

Include ample comments explaining what you are doing for each part of your code.

For questions 3 you may use word processor software like Word or similar, or use hand-written answer. Either way, transform your solution into a **single pdf file for both questions** and upload and submit it on the course Sakai site.

1. **R/Python Project:** We wish to examine the relation between price and sales of two editions of a textbook. The two editions are hard cover and paperback. The paperback has a blue cover, and the hardcover version an orange one. Each data item is collected from a different bookstore. Each bookstore can carry only one of either paperback and hardcover versions of the text. The data contains the amount of weekly sales from bookstores across the country, the price at which the text was sold, and the type (paperback or hardcover).

   1a) Read the file `textbookSales.csv` and put it in the data frame called `priceData`. Plot the scatter plot of the data, coloring paperback points blue and hardcover points orange.

   1b) Assuming the relationship between price and sales is linear for both editions, set up a linear model with sales as the target (response) variable, and price and edition as the independent variables. Call the linear models `modelA`. Print a summary of the model.

   1c) How much of the variation in sales is determined by this model?

   1d) What is the relationship between price and sales for the hardcover edition, and what is this relationship for the paperback version? Extract this information from the model you derived in question 1b.

   1e) A bookstore reports sales of 123 copies of the text at the price of 142, but the edition is not reported. According to the Bayes rule which edition is more likely to have been sold by this bookstore? Clearly Justify your answer.

   1f) Now suppose the data were presented to you without the information about which edition was sold. You are to use the Expectation-Maximization technique to separate the set of books. We assume that we know there are two types of books, but pretend we do not know price/sales data corresponds to which edition. The objective is to use the EM algorithm to figure this out. Here is the outline of the EM algorithm in this case. The instructions are for R users; for Python, figure out the equivalent procedure:

   - In R make sure you have installed the `EMcluster` package (Figure out what the equivalent package is in Python.)
   - Use the `init.EM` function to initialize the EM process. As the input matrix you should pass the two-column matrix made up of price and sales data of `pricedata` (so no information about the edition feature is passed to `EMcluster` functions.) Save the output of `init.EM` in an object called `em1`.
   - Using `em1` run the `emcluster` function to estimate the latent variable. Assign the output of the `emcluster` function to an object called `em`.
   - Use the `assign.class` function, and again pass the price and sales columns of `pricedata` as a matrix. Assign the output to an object called `c`.
   - Plot the scatter plot of `pricedata`, price vs sales, but this time color with respect to classes produced by the `assign.class` function in `c`.

- Using the `table` function compare the classes produced by the EM method and the original classes in the Edition column of `pricedata`. How many are misclassified? What is the misclassification rate?
- Run the linear regression model again, but this time instead of edition use the classes produced by the EM algorithm, and stored in `c`. Compare the slope and intercept for each class, with the slope and intercept for the original classes paperback and hardcover.

2. **R/Python Project:** We continue with the movie data set you used in the previous homework.

2a) Read the original data into `movieDat` as you did in the previous homework. The column under the `genres` is in the `JSON` format (check Wikipedia to get familiar with this simple format.) Each movie may belong to several genres. You must parse this column for all movies, collect the set of all available genres, and for each one create a new binary feature whose name starts with `genre_`. So after this pre-processing you should have new features such as `genre_Action`, `genre_Adventure` etc. Since the format in which the genres are stored in this data set is `JSON`, you may wish to look into the relevant libraries in R and Python. In R you may wish to look at libraries `rjson` and `litejson` for utilities working with `JSON` format. In Python `import json` will load the necessary library items. See this page for more information on Python. Of course, you could ignore the JSON libraries and use direct string processing to extract genre names, but this may be more time-consuming. (For now ignore the other JSON features in the data.)

2b) As in the previous homework, extract only the numerical features and save it in the data frame `nmovieDat`. However, add all columns you generated in part a) for genres to `nmovieDat`. Finally, create a new column called `profit` which is revenue minus budget. Compute this column and add it to the `nmovieDat`.

2c) Once you create the `nmovieDat` data frame, divide the data into two groups of training and test sets. Choose, randomly 80% of the data and put them in a data frame called `nmovieDatTrain`. Put the remainder in a data frame called `nmovieDatTest`.

2d) Build a linear regression model called `lmmodel1` relating profit to only the numerical features (except budget and revenue, of course.) What is the percentage of variation in profit explained by `lmmodel1`?

2e) Now build a linear regression model called `lmmodel2` relating profit to all features of `nmovieDatTrain`. What percentage of the variation in profit is described by `lmmodel2`?

3. For this assignment we are going to test the binary classifications using SVM (for various kernels) and the logistic regression.

3a) Create a new feature called `incomeGenre`. The idea is we are going to lump together genres that tend to generate more revenue into one group and the rest into another. Under `incomeGenre` column put a "1" if the movie belongs to *two* or more of genres from the set: `Action, Adventure, Fantasy, Science Fiction, Crime, Drama, Thriller, Horror`. Make sure to make this new feature a categorical variable.

3b) Run the logistic regression modeling `incomeGenre` as a function of all numerical features (six in total). Use only the training data for this purpose, that is use only the same rows in the `nmovieDat` data frame. Name this model `logitModel1`. Print a summary of the model.

3c) Now predict this model on the test set, that is the data in `nmovieDatTest`. Compute and print the confusion table and classification error rate.

3d) Compute the BIC value of this model. In R you may use the `AIC()` function. You must supply the model and $\mathsf{k} = \ln(\mathsf{N})$ where N is the number of data points.

3e) Repeat parts 3b-3d, but this time use cross product B-Splines with `df=6`. Based on the BIC value, does the cross product model improve over the linear model?

3f) Repeat parts 3b-3c, but this time use an SVM model with polynomial kernel and degree 4.

3g) Repeat parts 3b-3c, but this time use an SVM model with radial basis kernel.