CSE 347/447 Data Mining: Homework 1

Due on 11:59 PM, September 14, 2017

Submission Instructions

The submission should be one .zip file containing: one .pdf file with your answers to Q1 - Q2, and two .py files (Imputer.py and Cleaner.py) with your answer to Q3 and Q4. Please name your file as "HW1.zip".

Q1 (6pt) Question 2 in 2.6 Exercises (page 89 of TSK)

Q2 (4pt)

(1)

It is desired to partition customers into similar groups on the basis of their demographic profile. Which data mining problem is best suited to this task?

(2)

Suppose in problem (1) the merchant already knows for some *of the customers whether or not they have bought widgets. Which data mining problem would be suited to the task of identifying groups among the remaining customers, who* might *buy widgets in the future?*

(3)

Suppose in problem (2) the merchant also has information for other items bought by the customers (beyond widgets). Which data mining problem would be best suited to finding sets of items which are often bought together with widgets?

(4)

Suppose that a small number of customers lie about their demographic profile, and this results in a mismatch between the buying behavior and the demographic profile, as suggested by comparison with the remaining data. Which data mining problem would be best suited to finding such customers?

Q3 Imputing Missing Data (5pt, Programming)

For various reasons, many real world datasets contain missing values, often encoded as blanks, NaNs or other placeholders. The scikit.learn package (http://scikit-learn.org/stable/) includes an Imputer class that provides basic strategies for imputing missing values, either using the mean, the medium, or the most frequent values of the row or the column in which the missing values are located.

Here we intend to enhance this Imputer class with another imputing strategy: linear regression (Details of linear regression is referred to Appendix D of TSK). For simplicity, we make the following assumptions: given an $n \times m$ data matrix,

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}$$

(a) missing values only happen in the last column of the data matrix; (b) the values of attributes roughly follow a linear model (for $i = 1, 2, \ldots, n$):

$$x_{im} = a_0 + a_1 x_{i1} + a_2 x_{i2} + \ldots + a_{m-1} x_{im-1} + \varepsilon_i$$

where $a_0, a_1, \ldots, a_{m-1}$ are coefficients and ε_i is the error term (following an unknown normal distribution $\mathcal{N}(0, \sigma^2)$). Using linear regression, we can estimate $a_0, a_1, \ldots, a_{m-1}$ from data objects that are complete. Then we can estimate the missing attribute value x_{im} (in the sense of expectation) as

$$x_{im} = a_0 + a_1 x_{i1} + a_2 x_{i2} + \ldots + a_{m-1} x_{im-1}$$

You are required to realize this idea as the function impute as provided in the partially implemented Python file Imputer.py. This function takes as input an NumPy array possibly with missing values in the last column (missing values are encoded as NaNs) and outputs an NumPy array with missing values imputed. You are allowed to use the linear regression module in scikit-learn [1]. You can test your code by running the command "python Imputer.py".

Q4 Cleaning IMDB Data (5pt, Programming)

The raw IMDB data comes in as a text file, which is formatted as follows:

```
"'Allo 'Allo!" (1982) {Firing Squashed (#8.5)} UK
"'Allo 'Allo!" (1982) {Fleeing Monks (#7.3)} UK
"'Allo 'Allo!" (1983) {Flight of Fancy (#3.4)} UK
"'Allo 'Allo!" (1983) {Forged Francs & Fishsellers (#5.15)} UK
```

Instead of all the brackets, tabs, and adverted commas, we would like to have three neat tab separated columns and put them in the order of: Production Year, Country, and Title. Duplicate entries are removed. If a title appears in multiple years, one entry for each year is shown. For example, for the sample data above, we want the formatted data as:

```
1982	UK	'Allo 'Allo!
1983	UK	'Allo 'Allo!
```

You are required to complete a function clean in the Python file Cleaner.py, which takes each line of input and performs the required formatting (hint: use Python's regular expression package: re). You can test your code by running the command "python Cleaner.py < imdb.txt".

[1] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html