CSE 40647/60647: Data Science **Fall 2018**

Homework 1: Written Assignments

Handed Out: August 21, 2018 *Due: September 7, 2018 11:55pm*

Save your homework submission as *NETID-hw1-written.pdf*.

1 Data Description (20 points)

Suppose the population size is $N = 1,000,000$. We sample $n = 9$ examples x_i ($1 \leq i \leq n$) from the data. Suppose the mean value of the sample data is $\mu = 10$ and the variance is $v = 18$. Now we sample one more example $x_{n+1} = 20$ from the data. So the sample size is $n + 1 = 10$. What is the new mean value μ' and the new variance v'?

Note that the result will be the same no matter what x_i ($1 \leq i \leq n$) are. You are expected to derive functions of calculating $\mu' = f(\mu, n, x_{n+1})$ and $v' = g(v, \mu, n, x_{n+1})$.

2 Data Reduction (20 points)

Suppose we have a normalized data matrix $\mathbf{A}$. We compute the largest singular value σ. If the correlation matrix is $\mathbf{C} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$, we compute the largest eigenvalue λ. Then tell which of the following is correct, and prove why.
(a) $\lambda = \sigma$; (b) $\lambda = \sigma^2$; (c) $\lambda = \log \sigma$; (d) $\lambda = \sqrt{\sigma}$.

CSE 40647/60647: Data Science **Fall 2018**

Homework 1: Programming Assignments

Handed Out: August 21, 2018 *Due: September 14, 2018 11:55pm*

Save your homework submission as *NETID-hw1-programming.zip*. The zip file has one pdf file *NETID-hw1-programming.pdf* and multiple code files.

You are allowed to use any programming language (Python recommended; R, C++, Java, etc.), however, the solutions will be in **Python**. You are allowed to use any public package (including Numpy and Scikit-learn) and any other kind of tools (Excel).

A Film Dataset

File name: Dataset-film-data.csv

Introduction: Suppose we have 1,000,000 films. We sample 150 films, so our dataset has 150 films (as the data objects). Each film has an ID ("f\$Digit"; \$Digit ∈ {1,. . . ,150}) and a genre (or called category/class) from {"ACTION", "ROMANCE", "COMEDY"}. Each film has four attributes. Each attribute is the average rating of the film given by a specific website. The attribute name is "AVGRATING_WEBSITE_\$Digit" (\$Digit ∈ {1,2,3,4}). The attribute values are numerical. Note that the rating scales can be different.

The first line is the header of attribute and label names. The following lines (rows) are object ID, attribute values, and label values of the data objects (films). The columns are separated by comma.

Example: The third line is "f2,4.9,3,1.4,0.2,ACTION": The film "f2" is an ACTION film. It is graded as 4.9 on Website 1, 3.0 on Website 2, 1.4 on Website 3, and 0.2 on Website 4.

Data Description (20 points)

The object-feature data matrix, which is denoted as **D**, has $m = 150$ objects and $n = 4$ features (and therefore, it has 600 values). Please use *Z-score normalization* to normalize the data matrix by each feature. The normalized data matrix is denoted as **A**. What are the maximum/minimum Z score for each feature?

Output: Write down the maximum/minimum Z score in the pdf. Save your code as NETID-hw1-1.py.

Data Visualization (30 points)

With the original data matrix **D**:

(1) generate *boxplot* for the first attribute "AVGRATING_WEBSITE_1;"

Output: Show the figure in the pdf. Save your code as NETID-hw1-2-1.py.

(2) generate *histogram* for the third attribute "AVGRATING_WEBSITE_3" (which will be considered as the second feature in the following pair-wise plots 4, 5, 6). Set the number of histograms/bins as 10;
Output: Show the figure in the pdf. Save your code as NETID-hw1-2-2.py.

(3) generate *bar chart* where the X-axis is the genre and the Y-axis is the mean value of average ratings of films given in the first attribute "AVGRATING_WEBSITE_1;"
Output: Show the figure in the pdf. Save your code as NETID-hw1-2-3.py.

(4) generate 2-dimensional *scatter plot* using attributes "AVGRATING_WEBSITE_1" and "AVGRATING_WEBSITE_3", where the marker types and colors should be different for different genres (node labels);
Output: Show the figure in the pdf. Save your code as NETID-hw1-2-4.py.

(5) generate *Q-Q plot* using attributes "AVGRATING_WEBSITE_1" and "AVGRATING_WEBSITE_3;"
Output: Show the figure in the pdf. Save your code as NETID-hw1-2-5.py.

(6) calculate the *KL divergence* between attributes "AVGRATING_WEBSITE_1" and "AVGRATING_WEBSITE_3." Hint: For each attribute, you need to generate a probability distribution. You can determine bins on the attribute value by yourself. For each bin, you have the frequency (which is the number of films that have an attribute value in the bin) can calculate the probability. For example, if the total number of films is 150, if there are 15 films whose value of attribute "AVGRATING_WEBSITE_1" is in the bin $[1.5, 2)$ (between 1.5 and 2), the probability is 0.1. Then you have two probability distributions: one for each attribute. Just call python libraries to calculate the KL divergence.
Output: Write down the KL divergence score in the pdf. Save your code as NETID-hw1-2-6.py.

Data Cleaning and Integration (15 points)

With the original data matrix **D**:
(1) calculate *correlation coefficients* $\rho_{i,j}$, which is ρ("AVGRATING_WEBSITE_i", "AVGRATING_WEBSITE_j"), between every pair of the attributes using covariance analysis.
Output: Write down the correlation coefficients in the pdf. Save your code as NETID-hw1-3-1.py.

With the Z-score normalized data matrix **A**:
(2) (the same as (1)) calculate *correlation coefficients* between every pair of the attributes using covariance analysis.
Output: Write down the correlation coefficients in the pdf. Save your code as NETID-hw1-3-2.py.

(3) Are the above results the same? And why?
Output: Write down your reasoning in the pdf.

Data Reduction (35 points)

(1) Principal Component Analysis (PCA) applied to this normalized data matrix $\mathbf{A}$ identifies the combination of attributes (principal components, or directions in the feature space) that account for the most variance in the data. Please plot (*scatter plot*) all the data samples on the 2 first principal components. Note that the marker types and colors should be different for different genres (node labels).
Output: Show the figure in the pdf. Save your code as NETID-hw1-4-1.py.

(2) Singular value decomposition (SVD) factorizes the normalized data matrix $\mathbf{A}$ to have singular values and singular vectors. Please plot (*scatter plot*) all the data samples on the 2 first left-singular vectors. Note that the marker types and colors should be different for different genres (node labels).
Output: Show the figure in the pdf. Save your code as NETID-hw1-4-2.py.

(3) Show the top-3 eigenvalues (in PCA) and the top-3 singular values (in SVD).
Output: Write down the values in the pdf.

(4) Can we use propagation-based method to estimate the first eigenvector/singular vector? Suppose we initialize with a normalized "left" vector $\mathbf{u}^{(0)} = \frac{\mathbf{1}_m}{\|\mathbf{1}_m\|} = [\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}, \ldots, \frac{1}{\sqrt{m}}]$, where $\|\mathbf{x}\|$ is the L-2 norm of $\mathbf{x}$. Then we have the "right" vector as $\mathbf{v}^{(0)} = normalize(\mathbf{A}^{\mathrm{T}}\mathbf{u}^{(0)})$. For the next iteration, we update the "left" vector as $\mathbf{u}^{(1)} = normalize(\mathbf{A}\mathbf{v}^{(0)})$. Then we iteratively compute $\mathbf{u}^{(t)}$ and $\mathbf{v}^{(t)}$ until t is some big number. Compare the vectors with the eigenvectors and singular vectors, and give your conclusion: Is it the first eigenvector or the first singular vector or neither? Note that the propagation method is capable to process super big matrices for dimensionality reduction though the size may not fit in the memory.
Output: Write down your reasoning in the pdf. Save your code as NETID-hw1-4-4.py.