Exercise I

AMTH/CPSC 445a/545a - Fall semester 207

Published: Thursday, September 7, 2017
Due: Thursday, September 21, 2017

Compress your solutions into a single zip file titled `<lastname and initials>_ assignment1.zip`, e.g. for a student named Tom Marvolo Riddle, `riddletm_assignment1.zip`. Include a single PDF titled `assignment1.pdf` and any MATLAB or Python scripts specified. Your homework should be submitted to Canvas before Thursday, September 21, 2017 at 1:00 PM.

Programming assignments should use built-in functions in MATLAB or Python; In general, Python implementations may use the `scipy` stack [1]; however, exercises are designed to emphasize the nuances of data mining algorithms - if a function exists that solves an entire problem (either in the MATLAB standard library or in the `scipy` stack), please consult with the TA before using it.

Problem 1

Provide an example application where each of the following tasks would be useful. Do not reuse examples from class.

1. Classification
2. Regression
3. Clustering
4. Anomaly detection
5. Association rule
6. Sequential pattern analysis

Include your answers in a PDF titled `assignment1.pdf`.

Problem 2

Classify the following attributes as binary, discrete, or continuous. Also classify them as qualitative (nominal or ordinal) or quantitative (interval or ratio). Some cases may have more than one interpretation so briefly indicate your reasoning if you think there may be some ambiguity.

1. Time in terms of AM or PM
2. Brightness as measured by a light meter

3. Brightness as measured by people's judgement
4. Angles as measured in degrees between 0 and 360
5. Bronze, Silver, and Gold medals as awarded at the Olympics
6. Height above sea level
7. Number of patients in a hospital
8. ISBN number for books
9. Military rank
10. Distance for the center of campus
11. Density of a substance in grams per cubic centimeter
12. Coat check number

Include your answers in a PDF titled `assignment1.pdf`.

Problem 3

Using the WAV file `mail.wav` posted under `exercises/ex1` on Canvas (this file is also available on the course website), write a script in MATLAB or Python called `problem3.m` or `problem3.py`, respectively, to perform the following.

1. Load the WAV file as a vector
2. Create a figure with a plot of the vector
3. Save the plot as a PNG file
4. Using the command `spectrogram` in MATLAB (or `scipy.signal.spectrogram` in Python) compute a spectrogram for window sizes $512, 1024, 2048$. The output should be frequency-time matrix with complex values
5. Replace the values of the frequency-time matrix with their absolute value, and restrict the matrix to the first 100 rows
6. Plot the resulting real-value matrix using `imagesc` in MATLAB (or `matplotlib.pyplot.imshow` in Python) with the y axis inverted.

Include the four plots in your PDF titled `assignment1.pdf`.

Remark. Due to differences in implementation between Python and MATLAB, this problem is easier to solve in MATLAB. In Python the window size can be controled by the parameter `nperseg`, and the result images may be different than the MATLAB implementation. Mathworks and Scipy publish fantastic documentation resources for using these functions. These are available for MATLAB at `https://www.mathworks.com/help/matlab/`. For Python, `https://www.scipy.org/docs.html` is available.

Problem 4

Download the wine dataset from https://archive.ics.uci.edu/ml/machine-learning-databases/wine/. Write a script called `problem4.m` or `problem4.py` performs the following.

1. Load the data [Hint: there are built in functions to read CSV files].
2. Pick two of wines, and create a star plot of the attributes of each.

Include the two plots in PDF titled `assignment1.pdf`.

Remark. The star plots of each wine should be in their own figure. Pay attention to the range of your plots. You may not use any pre-built star plot functions (including built-in ones in the aforementioned Python packages or MATLAB). Full points will be awarded for more detailed plots that include descriptive statistics, labels, and a title.

Problem 5

Let S^k denote the k-dimension sphere in $\mathbb{R}^{k+1}$ defined

$$S^k = \{x = (x_1, x_2, \ldots, x_k, x_{k+1}) \in \mathbb{R}^{k+1} : \|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_k^2 + x_{k+1}^2} = 1\}.$$

Given a set of points $X \in \mathbb{R}^d$, the set of pairwise distance D is defined

$$D = \{\|x - y\|_2 \in \mathbb{R} : x, y \in X.\}.$$

Write a script called `problem5.m` or `problem5.py` which does the following for $k = 1, 2, 3$.

1. Generate 1000 points X_k uniformly at random on $S^k \subset \mathbb{R}^{k+1}$.
2. Compute the set of pairwise distances D_k for X_k.
3. Create a histogram with 25 bins of equal width of D_k
4. Save the histogram as `eqwidth_`k`.png`.
5. Create a histogram with 25 bins each containing an equal number of points of D_k.
6. Save the histogram as `eqpoints_`k`.png`.

Include the six plots in your PDF titled `assignment1.pdf`.

Remark. Refer to [2] for an algorithm to generate points uniformly on an N-dimensional sphere. Consider normalizing your histograms. As before, full points will be awarded for fastidiously labeled solutions.

Problem 6

This problem uses the text file `tweets.txt` posted in `exercises/ex1` on Canvas (this file is also available on the course website). The purpose of this exercise is to write a script called `problem6.m` or `problem6.py` that builds a document-term representations of tweets and then analyzes correlations in them. For this exercise, each of the 188 lines (i.e., tweet) of the text file should be considered as a separate document, while the words in the text file of length at least 5 will act the terms. The text file only contains lower case letter, numbers, and whitespace characters. A word is any sequence of alpha-numeric characters separated by one or more whitespace characters. The script should do the following steps:

1. Read and parse the file so into a cell array (or list) of tweets, where each tweet is itself a cell array (or list) of words. Call this array (or list) `tweets`.
2. Find the ten most frequent terms in all the tweets and store them in a cell array (or list) called `terms`.
3. Verify that the most frequent term is `iphone` and ignore it in the next steps, so the script only considers the next 9 terms
4. Build a 188×9 document-term matrix between tweets and terms and call it `A`.
5. Build a 9×9 correlation matrix between terms, based on `A`, and call it `C`.
6. Build a 9×2 cell array (or list of lists) where the first column contains the terms you used in steps 4-5 in alphabetical order, and the second column contains for each of these terms, its most correlated term (excluding itself) based on `C`. Call this cell array (or list of lists) `pairs`.
7. Print the term pairs in `pairs`.

Include the 9 pairs of printed words in your PDF titled `assignment1.pdf`.

References

[1] "The scipy stack specification." [Online]. Available: https://www.scipy.org/stackspec.html

[2] G. Marsaglia *et al.*, "Choosing a point from the surface of a sphere," *The Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 645–646, 1972.