CS 540: Introduction to Artificial Intelligence
Homework # 6

Assigned: 3/12
Due: 4/02 before class

Hand in your homework:

- There are two questions in this homework. You are free to use any programming language (Java, Python, MATLAB, etc) or other software (Excel, SPSS) to solve them. We recommend you use Python, as it is easy to process text and perform matrix computation.
- Please answer the questions in a single pdf file. Please typeset your homework, do not submit handwritten+scan answers. The pdf file should be submitted on Canvas.
- If you've written code to solve a certain part of a problem, you should include the code in your pdf file. The code for all parts should go in an appendix.
- If you haven't written any code to solve the problems, you should describe in detail how you use any other tools in the appendix.

Late Policy:

All assignments are due at the beginning of class on the due date. One (1) day late, defined as a 24-hour period from the deadline (weekday or weekend), will result in 10% of the total points for the assignment deducted. So, for example, if a 60-point assignment is due on a Wednesday 9:30 a.m., and it is handed in between Wednesday 9:30 a.m. and Thursday 9:30 a.m., 6 points will be deducted. Two (2) days late, 25% off; three (3) days late, 50 off. No homework can be turned in more than three (3) days late. Written questions and program submission have the same deadline.

Collaborative Policy:

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TAs, and instructor in order to help you answer the questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers or code fragments from anyone or anywhere
- not to allow your answers to be copied
- not to get any code on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

Problem 1. Natural Language Processing [30 points]

In this question, you will get familiar with some basic ideas in natural language processing. Download the **news.zip** file provided with the homework assignment. Unzip it on your computer, you should see 511 text files. This is the news data set from `http://mlg.ucd.ie/datasets/bbc.html`. We call the whole collection the corpus. In this question, you need to break the corpus into "computer words". It is OK if these computer words do not fully agree with our natural language definition of words: For example, you may have a token like `However,` with that comma attached, or something like `1-2`. You may assume `the` and `The` are different computer words.

Hint: For Python users, you may use `collections.Counter` in python to solve this problem. `matplotlib` is an useful Python plotting library. For Java users, `Pattern` and `split` are useful for processing text.

1. (5 points) How many computer word tokens (occurrences) are there in the corpus? How many computer word types (distinct words) are there in the corpus?

2. (5 points) Sort the word types by their counts in the corpus, from large to small. List the top 20 word types and their counts.

3. (6 points) Let the word type with the largest count be rank 1, the word type with the second largest count be rank 2, and so on. If multiple word types have the same count, you may break the rank tie arbitrarily. This produces $(r_1, c_1), \ldots, (r_n, c_n)$ where r_i is the rank of the ith word type, and c_i is the corresponding count of that word type in the corpus; n is the number of word types.

   - (2 points) Plot r on the x-axis and c on the y-axis, namely each (r_i, c_i) is a point in that 2D space (you can choose to connect the points or not).
   - (2 points) Plot $\log(r)$ on the x-axis and $\log(c)$ on the y-axis. Use e or 10 as base.
   - (2 points) Briefly explain what the shape of the two curves mean.

4. (5 points) Now we want to use $tf \cdot idf$ to capture the key words in one text. Recall the lectures (page 15 in slides and section 6 in notes). What is the $tf \cdot idf$ of the word "contract" in the text "098.txt"? Find top 10 words with the highest $tf \cdot idf$ in the text "098.txt". (Hint: Please use 10 as base when computing idf)

5. (6 points) The document "098.txt" is represented by feature vector v_1, the document "297.txt" is represented by feature vector v_2. What is the cosine similarity of v_1 and v_2 if we use **bag-of-words** representation? What is the cosine similarity if we use $tf \cdot idf$? Are the values of cosine similarity from the two methods same? Why or why not?

6. (3 points) Discuss *two* potential major issues with your computer words, if one wants to use them for natural language processing.

Problem 2. Principal Component Analysis [30 points]

Bob plans to buy a car. He decides to use principal component analysis to help him make the decision. Download the **cardata.csv** provided with the homework assignment. The data are car information from the websites. Each of the 356 lines (not including the first line) in the file represents a car. In each line, the first column is the car's model. The second column is the car's category (Minivan, Wagon, Sedan, Sports, SUV). The 3rd to 13th columns are values such as retail price, dealer price, engine,..., etc.

Hint: Feel free to use open source tools to help you solve this problem. For Python users, numpy is an useful tool for matrix computation and reading data from csv. `numpy.linalg.eigh` is for computing the eigenvalues and eigenvectors. `numpy.cov` is for estimating a covariance matrix. You may also use scikit-learn to perform the PCA. For Matlab users, a PCA demo is available on the course website. `Jama` is a Java implementation for computing eigenvalues and eigenvectors.

1. (4 points) Bob wants to use a vector $\mathbf{x}_i (i = 1, ..., 356)$ to represent each car. For each car, he uses the 2nd column as the label, and use 3rd to 13th columns are the feature value. What is the dimension of the vector? What is the mean value of the Retail($) and horsepower before centering and normalizing?

2. (5 points) Bob uses the following ways to center and normalize data. First he calculates the sample mean μ and standard deviation $\hat{s}$ (You may use `numpy.std` to compute the standard deviation):

   $$\mu = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$$

   $$\hat{s} = \sqrt{\frac{\sum_{i=1}^{n} (\mathbf{x}_i - \mu)^2}{n-1}}$$

   where $n = 356$. Then he centers and normalizes the data using

   $$\mathbf{x}_i := \frac{\mathbf{x}_i - \mu}{\hat{s}}$$

   After centering and normalizing the data, Bob begins to compute the covariance matrix and performs an eigen value decomposition. He sorts the eigenvalues from large to small. What are the corresponding **first** and **third** eigenvectors?

3. (6 points) In the first eigenvector, which coordinates are positive? Which feature do they refer to? What does it mean that the coordinates are positive?

4. (8 points) Create a scatter plot with each of data points of **Minivan**, **Sedan**, **SUV** projected on to the first two principal components. The horizontal axis should be the first principal component v_1, and the vertical axis is the second principal component v_2. Each individual should be projected onto the subspace spanned by v_1 and v_2. Please use a different color for different categories and include a legend which represents the labels. An example can be found here.

5. (7 points) Based on the plot in the last question, which cars are clustered most strongly? What can Bob conclude based on the plot?