CS57300 Homework	— Homework 1 , Problem 1 —	(January 21, 2016)
Shaobo Fang (0024260324)		

Q1:
(a). Show that the variance of var(X+Y) = var(X) + var(Y) + 2 cov(XY), where cov is the covariance between X and Y.
(b). State one condition that makes $\frac{cov(X,Y)}{\sqrt{var(x)*var(y)}}$. (c). State one condition that makes $\frac{cov(X,Y)}{\sqrt{var(x)*var(y)}}$.

A:
(a).

$$\begin{aligned} Var[X+Y] &= \mathbb{E}(X+Y)^2 - (\mathbb{E}(X+Y))^2 && (1)\\ &= \mathbb{E}(X^2+2XY+Y^2) - [(\mathbb{E}[X])^2 + 2*\mathbb{E}[X]\mathbb{E}[Y] + (\mathbb{E}[Y])^2] && (2)\\ &= \mathbb{E}(X^2) - (\mathbb{E}(X))^2 + \mathbb{E}(Y^2) - (\mathbb{E}(Y))^2 + 2[\mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)] && (3)\\ &= var(X) + var(Y) + 2*cov(X,Y) && (4) \end{aligned}$$

(b). Let's take an intuitive example: assume $\mu_x = \mu_y = 0$, and $\sigma_x = \sigma_y = a$. The condition that makes $cov(X,Y)$ maximum is $\mathbf{X} = \mathbf{Y}$:

$$cov(X,Y) = \mathbb{E}(XY)/a^2 = \frac{a^2}{a^2} = 1$$

(c). Similarly as the example in (b), The condition that makes $cov(X,Y)$ minimum is $\mathbf{X} = -\mathbf{Y}$:

$$cov(X,Y) = \mathbb{E}(XY)/a^2 = -\frac{a^2}{a^2} = -1$$

Q2:
(a). Let X_i be i.i.d following Gaussian distributions $N(\mu, \sigma^2)$. Find maximum likelihood estimator μ and σ.
(b). Prove the conditional Bayesian rule of:

$$P(B|A,C) = \frac{P(A|B,C)P(B|C)}{P(A|C)}$$

.
A:

(a). The probability density function of X_i is:

$$f(x_i;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

The likelihood function is:

$$L(\mu,\sigma) = \sigma^{-n}(2\pi)^{-n/2} exp^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}$$

The log-likelihood function is:

$$log(L(\mu,\sigma)) = constant - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

The MLE for μ is:

$$Set: \quad \frac{\partial log(L(\mu,\sigma))}{\partial \mu} = 0$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} Xi = \bar{X}_i$$

The MLE for σ is:

$$Set: \quad \frac{\partial log(L(\mu,\sigma))}{\partial \sigma} = 0$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i-\hat{\mu})^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

$$\rightarrow \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

(b).

$$P(A,B|C) = \frac{P(A,B,C)}{P(C)} \tag{5}$$

$$= \frac{P(A,B,C)}{P(A,C)} * \frac{P(A,C)}{P(C)} \tag{6}$$

$$= P(B|A,C) * P(A|C) \tag{7}$$

$$P(A,B|C) = \frac{P(A,B,C)}{P(B,C)} * \frac{P(B,C)}{P(C)} \tag{8}$$

$$= P(A|B,C) * P(B|C) \tag{9}$$

$$P(B|A,C) * P(A|C) = P(A|B,C) * P(B|C) \tag{10}$$

$$\rightarrow P(B|A,C) = \frac{P(A|B,C)P(B|C)}{P(A|C)} \tag{11}$$

Q3:
(a):(i). What is the probability that Sheldon has WNV?
(ii). The WNV virus is fatal in 5% of the cases. What is the probability that Sheldon will die this year? Assume a fatality rate of any cause (car accident, etc.) of 0.1% that is independent of whether or not Sheldon has WNV.
(b). Alice and Bob are playing a simple dice game. Each rolls one dice and the one with higher number wins. If the numbers are the same, they roll again. If Alice just won, what is the probability that she rolled a '4' ?
A:

(a) (i): It is crucial to interpret the question:

1. False positive rate of '1/10,000' is defined as: Giving the test result is positive, there is 1/10,000 chance the truth is negative.
2. 'The test correctly identifies the presence of WNV is 0.95' is a statement of conditional probability. (Conditions: The person carries WNV and conduct the test. Outcome: Correctly Identifies). The mathematical:

$$P(\text{“}Tested\ WNV\ positive''|\text{“}Carries\ WNV\ and\ conducts\ testing'') = 0.95$$

For Sheldon's case the conditional probability does not apply as we are not sure if he really carries WNV. So unfortunately for Sheldon, as the false positive rate is extremely low at 1/10,000, there is 9,999/10,000 chance he is infected with WNV.

$$\rightarrow \text{The probabaility of Sheldon has WNV is } 99.99\%$$

(ii):
As Sheldon already been identified with WNV, we first calculate the probability that he dies of WNV this year:

$$P_{WNV} = 0.9999 * 0.05 = 0.049995$$

The probability he dies in general due to all the causes:

$$P_{general} = 0.001$$

The intersection that general death was due to WNV is:

$$P_{generalWNV} = 2000/300,000,000 * 0.005 = 3.33 \times 10^{-8}$$

The total probability that Sheldon die this year is:

$$P_{WNV} + P_{general} - P_{generalWNV} \approx 0.050995 = 5.0996\%$$

Hence in general Sheldon will has a 5.0995% probability to die this year.
(b). First, we list all the probabilities that Alice will win (with each number rolled):

1. Alice Rolled '1': 0
2. Alice Rolled '2': 1/6
3. Alice Rolled '3': 2/6
4. Alice Rolled '4': 3/6

5. Alice Rolled '5': 4/6
6. Alice Rolled '6': 5/6

$$P(Alice\ Rolled\ `4'\ and\ Won|Alice\ Won) = \frac{Alice\ Rolled\ `4'\ and\ Won}{P(Alice\ Won)}$$

$$P(Alice\ Rolled\ `4'\ and\ Won|Alice\ Won) = \frac{Alice\ Rolled\ `4'\ and\ Won}{\sum_{i=1}^{6} P_{Alice\ rolled\ i\ and\ won}}$$

$$P(Alice\ Rolled\ `4'\ and\ Won|Alice\ Won) = \frac{4/6}{1/6 * (1+2+3+4+5)}$$

$$P(Alice\ Rolled\ `4'\ and\ Won|Alice\ Won) = \frac{4}{15}$$

Q4:
(a). Plot the ECCDF of the Marvel Characters.
(b). Let A be the adjacency matrix connecting characters to comic books, where $A_{i,j}$ has character i appearing on comic book j. Let A^T be the transpose of matrix A.
(i): What does $W = AA^T$ represent, what is the largest degree?
(ii): What does $W = A^TA$ represent, what is the largest degree?
(iii): What is $U = A^TA$
(iv): Report relationship between x_i and $D_{i,i}$.
A:

(a).

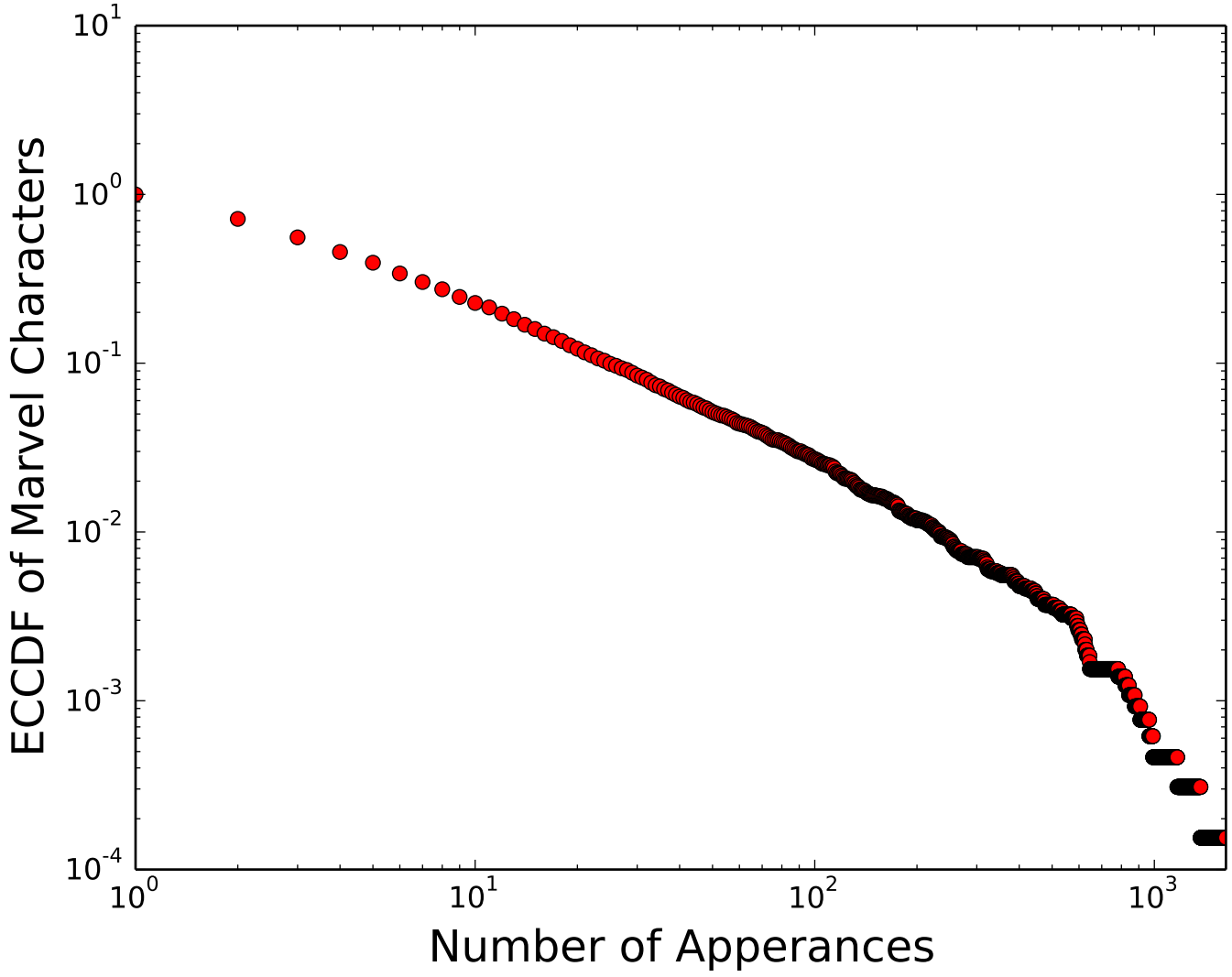

Figure 1. The ECCDF Plot of the Marvel Characters

The Python Code Used to Generate the ECCDF

```
# Here the example code starts
import numpy as np
import os
import shutil
import glob
from matplotlib import use
import scipy.io as sio


marvel_file = open('marvel.txt')
marvel_file.readline()
# Avoid using the xterminal to create plots (needed if plotting on Scholar)
character_indx_list = []
character_show_up = []
for line in marvel_file:
        #print line.index(" ")
        space_loc = line.index(" ")
        tmp_character = line[0:space_loc]
        #print tmp_character
        character_indx_list.append(int(tmp_character))

max_indx = max(character_indx_list)
min_indx = min(character_indx_list)

character_indx_showup = np.zeros(max_indx)
#print len(character_indx_showup)

marvel_file = open('marvel.txt')
marvel_file.readline()
for line in marvel_file:
        #print line.index(" ")
        space_loc = line.index(" ")
        tmp_character = line[0:space_loc]
        character_indx_showup[int(tmp_character) - 1] += 1

#print character_indx_showup
#print max(character_indx_showup)
#print min(character_indx_showup)
#print len(character_indx_showup)
character_counter = 0;
for each in character_indx_showup:
        if each != 0:
                character_counter += 1

y_CDF = []
x = []
for occurrences in range(int(min(character_indx_showup)), int(max(character_indx_showup))+
        y_CDF.append(0)
        x.append(occurrences)
        for increment in character_indx_showup:
                if increment >= occurrences:
                        y_CDF[occurrences] += 1
        y_CDF[occurrences] = float(y_CDF[occurrences])/float(character_counter)

#print y_CDF
```

```
#print len(y_CDF)

use("Agg")
import matplotlib.pyplot as plt
import pylab as pylab
import math

plt.xlim([1,max(x)])
plt.xlabel("Number of Apperances", fontsize=18)
plt.ylabel("ECCDF of Marvel Characters", fontsize=18)
plt.loglog(x,y_CDF,"ro")
plt.savefig('ECCDF_plot_marvel.pdf')


A_matrix = np.zeros([int(max_indx), int(19428 - 6487 + 1)])
print A_matrix.shape
marvel_file = open('marvel.txt')
marvel_file.readline()
for line in marvel_file:
        space_loc = line.index(" ")
        tmp_character = line[0:space_loc]
        tmp_comic = line[space_loc:]
        #print tmp_character, tmp_comic
        #print line
        A_matrix[int(tmp_character) - 1, int(tmp_comic) - 6487] = 1
#plt.matshow(A_matrix, fignum=100, cmap=plt.cm.gray)
#plt.show()

A_matrix[0,0] = 1

#W = np.mat([[1,0], [0,1]]) * np.transpose([[1,0], [0,1]])

A_np = np.mat(A_matrix)

#print A_np[1,2]

size =  A_np.shape
#print size
#print type(size)
#print size[1]

#W = np.mat(A_np) * np.transpose(A_np)
#U = np.transpose(A_np) * np.mat(A_np)
#print W
#print type(np.mat(A_matrix))
#print type(W)

#np.savetxt('w_matrix.txt', W)
#np.savetxt('u_matrix.txt', U)

def matrix_mult(A,B):
        size_A = A.shape
        size_B = B.shape
        product_size = int(size_A[0])
        length = int(size_A[1])
        product = np.zeros([product_size, product_size])
        for ii in range(0,product_size):
```

```
                print ii
                for jj in range(0, product_size):
                        tmp_sum = 0;
                        for mm in range(0,length):
                                #print 'Index: ', jj, mm, ii
                                tmp_sum = tmp_sum + B[mm,jj]*A[ii,mm]
                        product[ii,jj] = tmp_sum

        return product

A = np.mat(A_np)
B = np.transpose(A_np)
#W = matrix_mult(A,B)
#print A[1,:].shape
#print B[:,1].shape

#sio.savemat('saved_struct.mat', {'A': A})

W = np.dot(A,B)
#U = np.dot(B,A)
#np.savetxt('w_matrix.txt', W)
#np.savetxt('U_matrix.txt', U)
#print W.shape
```

(b)
(i). $W = AA^T$ is the character-character similarity matrix (regarding appearances on comic books). The entity name of the hero associated with the largest ID is 5306. We look up ID 5306, and the answer is **5306 "SPIDER-MAN/PETER PAR"**.
(ii). $W = A^TA$ is the comic book-comic book similarity matrix (regarding the characters). The largest degree would be associated with the book with title. **6496 "COC 1"**.
(iii). $U^T = (A^TA)^T = A^TA$ (iv). By the definition of eigenvectors $\vec{x}$:

$$A\vec{x} = \lambda\vec{x}$$

In our case, $A = P$, $\lambda = 1$. Basically the task is to find the eigenvectors whose eigen values are '1'.
(Tried to use Python to conduct eigen decomposition, found Python is not very good dealing with matrix calculation...) **As the dimensionality of matrix P was quite huge, to efficiently decompose the eigen vectors, we used Matlab.** Then, easily we can extract the eigen vectors that are associated with the eigen values '1'.
There are more than one solutions to our problem, for this dataset.
Based on observation, we have found the relationships between x_i and D_{ii}. The corresponding relationship is:

$$\sum_i x_i D_{i,i} = 1$$

Discussion: Although Matlab has many disadvantages against other tools, it is indeed very powerful dealing with matrix operations. A matrix of relatively large size can be computed rather efficiently, compared to the using standard Python library. It would be good if there could be a workshop on how to deal with matrices of large dimensions in Python. Matlab Code Used for this section attached as below.

```
close all
clear all
clc

raw_data = load('saved_struct.mat');
A = raw_data.A;
```

```
A(1,1) = 1; % Make a minor correction that for some reason Python give wrong
W = A*A';

mat_size = size(W);
D = eye(mat_size(1));

for ii = 1:1:mat_size(1)
        D(ii,ii) = sum(W(ii,:));
end

for ii = 1:1:mat_size(1)
    D(ii,ii);
end

P = inv(D)*W;
[V,D_eig] = eig(P);

cnt = 1;
for ii = 1:1:mat_size(1)
    if D_eig(ii,ii) == 1
        d_vec(cnt) = ii;
        c_columns(:,cnt) = V(:,ii);
        cnt = cnt + 1;
    end
end

for jj = 1:1:(cnt-1)
    error(jj) = sum(P*c_columns(:,jj) - c_columns(:,jj));
end

error
d_vec

for ii = 1:1:mat_size(1)
    D_diagonal(ii) = D(ii,ii);
end

for jj = 1:1:(cnt - 1)
figure
subplot(2,1,1)
plot(D_diagonal)
subplot(2,1,2)
plot(c_columns(:,jj))
norm(c_columns(:,jj));
end
```