University at Buffalo
The State University of New York

CSE574 Introduction to Machine Learning

Project 1

Amir Baghdadi
50135018

Introduction

In machine learning, the obtained results are uncertain in nature and the best way to interpret those results is to have a knowledge of probability which deals with uncertainty in the real world. In addition, the knowledge of statistics is an important factor for analyzing the outcomes from machine learning algorithms. In other words, probability and statistics are the languages of reasoning in the uncertain world of machine learning.

The purpose of this project was to calculate some statistics and probability concepts using the data of ranking for US universities in CS with four criteria of CS ranking based on US news, the value of Research Overhead, Admin Base Pay, and Out-of-state Tuition. In addition, the probability of the variables have been determined and, by considering the correlation between each variable and constructing a Bayesian network, this probability has been improved by obtaining a higher probability for the variables in terms of log-likelihood.

Tasks

Part 1)

Using Python 3, the mean, variance, and standard deviation have been calculated for each of the four variables of CS Score, Research Overhead, Admin Base Pay, and Tuition. The results of Python code are provided here:

```
mu = [mu1, mu2, mu3, mu4]
var = [var1, var2, var3, var4]
sigma = [sigma1, sigma2, sigma3, sigma4]

mu = [ 3.21000000e+00 5.34000000e+01 4.69000000e+05 2.97000000e+04]
var = [ 4.48000000e-01 1.26000000e+01 1.39000000e+10 3.07000000e+07]
sigma = [ 6.69000000e-01 3.55000000e+00 1.18000000e+05 5.54000000e+03]
```

where 1 to 4 corresponds to CS Score, Research Overhead, Admin Base Pay, and Tuition, respectively.

Part 2)

The covariance and correlation values for each pair of the variables have been determined and the results are shown in terms of matrices as well as a graph for correlation coefficients.

```
covarianceMat =
[[ 4.57000000e-01 1.11000000e+00  3.88000000e+03  1.06000000e+03]
[  1.11000000e+00 1.29000000e+01  7.03000000e+04  2.81000000e+03]
[  3.88000000e+03 7.03000000e+04  1.42000000e+10 -1.64000000e+08]
[  1.06000000e+03 2.81000000e+03 -1.64000000e+08  3.14000000e+07]]

correlationMat =
[[ 1.     0.456  0.0482  0.279]
[  0.456  1.     0.165   0.14 ]
[  0.0482 0.165  1.     -0.245]
[  0.279  0.14  -0.245   1.  ]]
```

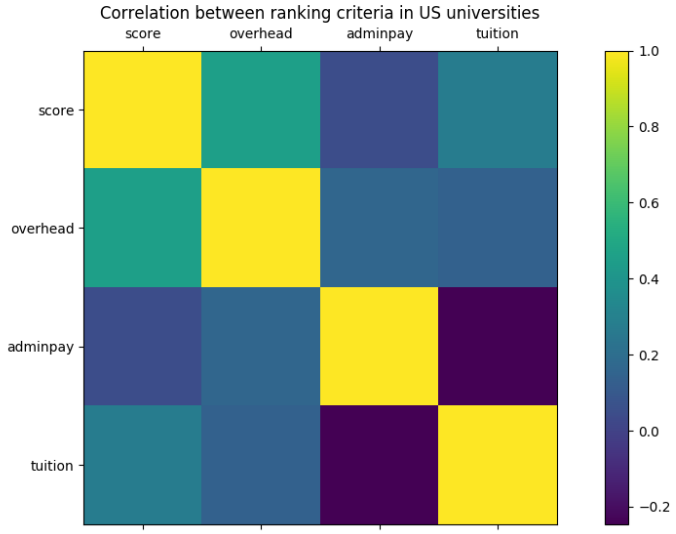


Fig. 1. Correlation graph between ranking the US universities

According to the correlation coefficient values and the graph above, the most correlated variables are CS Score and Research Overhead with the coefficient of 0.456 and the least correlated ones are CS Score and Admin Base Pay with 0.0482 as the coefficient of correlation. This is an interesting result in a sense that the earning of the university administrators does not help for a better ranking of the school, on the other hand, the research overhead will highly affect the university score and improves the ranking. These results indicate that increasing the research overhead can be considered as an important factor for improving the quality and research outcome which directly affects the score and ranking of the university.

Part 3)

The log-likelihood of the data was determined using the mean and variances computed in part 1 and using the relation below:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\sum|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \sum^{-1}(\mathbf{x}-\mu)\right] \quad (1)$$

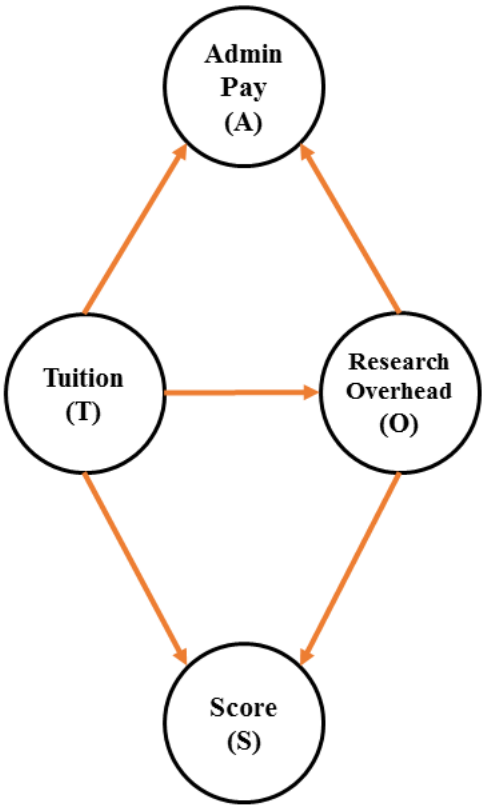
$$\mathcal{L}(\mathbf{x}_1,..\mathbf{x}_N) = \sum_{i=1}^{N} \log p(\mathbf{x}_i) \quad (2)$$

In order to calculate the log-likelihood, a function has been defined in the Python code which is provided in the appendix. The log-likelihood with 3 significant figures is:

```
logLikelihood = -1320.0
```

Part 4)

Considering the correlation coefficients from the correlation matrix, a Bayesian network was defined that provides a relationship between the variables. In defining this network, the value of 0.05 was used as a threshold for considering the relationship between the two variables. The network is shown in the figure below:



$\mathbf{X} = \{A, T, O, S\}$

$p(\mathbf{X}) = p(A|T,O)\ p(T)\ p(O|T)\ p(S|T,O)$

Fig. 2. The Bayesian network graph and the associated joint probability function

In this Bayesian network, the direction of each arrow is defined by a logical rationale. For instance, the correlation coefficient of 0.456 shows a relation between Research Overhead and Score and this is reasonable to consider that the Score is affected by the Research Overhead and not the other way. Therefore, the arrow in the Bayesian network was considered from the Research Overhead to Score. Similarly, the Admin Pay is affected by the Research Overhead knowing the correlation coefficient of 0.165 (>0.05) between the two variables. This Bayesian network can be expressed in terms of a 4-by-4 binary matrix representing the acyclic directed graph showing the connection of the Bayesian network. In this matrix, the rows show the direction of the arrow from a specific variable and the columns show that to a specific variable. `BNgraph` is representing this graph:

```
BNgraph =
                To score overhead adminpay tuition
From score       [[  0       0        0       0    ]
     overhead    [   1       0        1       0    ]
     adminpay    [   0       0        0       0    ]
     tuition     [   1       1        1       0   ]]
```

For instance, the array element (2, 1) which is shown in **bold** red (**1**) is representing an arrow from Overhead to Score.

Part 5)

Considering this Bayesian network, the joint probability will be defined as:

$$p(\mathbf{X}) = p(\mathrm{A}|\mathrm{T},\mathrm{O})\, p(\mathrm{T})\, p(\mathrm{O}|\mathrm{T})\, p(\mathrm{S}|\mathrm{T},\mathrm{O}) \tag{3}$$

where $\mathbf{X} = \{\mathrm{A},\mathrm{T},\mathrm{O},\mathrm{S}\}$ and A, T, O, and S stand for Admin Pay, Tuition, Research Overhead, and Score, respectively.

In order to find the values of each probability in the joint probability function, we have found the least squares solution for the linear Gaussian model parameters (i.e., β_i and σ^2). As an example for $p(Y|X_1,X_2) \equiv p(\mathrm{A}|\mathrm{T},\mathrm{O})$, variable $\theta = (\beta_0,\beta_1,\beta_2,\sigma^2)$ defines the linear Gaussian model for log-likelihood as follows:

$$L(\theta) = \log\big(p(Y|X_1,X_2)\big) = \sum_{n=1}^{49}\Big[-\tfrac{1}{2}\log(2\pi\sigma^2) - \tfrac{1}{2\sigma^2}(\beta_0 x_0[n] + \beta_1 x_1[n] + \beta_2 x_2[n] - y[n])^2\Big], \tag{4}$$

where $x_0[n] = 1$, $x_1[n]$ are the variables for Tuition (T), $x_2[n]$ for Research Overhead (O), and $y[n]$ for Admin Pay(A). By taking the partial derivatives of the log-likelihood function $L(\theta)$ with respect to β_i and σ

and setting to zero, the least squares solution are:

$$\beta = A^{-1}y, \tag{5}$$

where

$$A = \begin{bmatrix} \sum_{n=1}^{49} x_0[n]x_0[n] & \sum_{n=1}^{49} x_1[n]x_0[n] & \sum_{n=1}^{49} x_2[n]x_0[n] \\ \sum_{n=1}^{49} x_0[n]x_1[n] & \sum_{n=1}^{49} x_1[n]x_1[n] & \sum_{n=1}^{49} x_2[n]x_1[n] \\ \sum_{n=1}^{49} x_0[n]x_2[n] & \sum_{n=1}^{49} x_1[n]x_2[n] & \sum_{n=1}^{49} x_2[n]x_2[n] \end{bmatrix} \tag{6}$$

$$y = \begin{bmatrix} \sum_{n=1}^{49} y[n]x_0[n] \\ \sum_{n=1}^{49} y[n]x_1[n] \\ \sum_{n=1}^{49} y[n]x_2[n] \end{bmatrix}, \tag{7}$$

and by taking the derivative with respect to σ and setting the values to zero have:

$$\sigma^2 = \tfrac{1}{49}\sum_{n=1}^{49}(\beta_0 x_0[n] + \beta_1 x_1[n] + \beta_2 x_2[n] - y[n])^2 \tag{8}$$

Finally, the log-likelihood of the joint probability distribution $p(\mathbf{X})$ is:

$$log\ p(\mathbf{X}) = \log p(\mathrm{A}|\mathrm{T},\mathrm{O}) + \log p(\mathrm{T}) + \log p(\mathrm{O}|\mathrm{T}) + \log p(\mathrm{S}|\mathrm{T},\mathrm{O}) \tag{9}$$

By performing the above-mentioned algorithm on the data in Python the log-likelihood for this Bayesian network was:

```
BNlogLikelihood = -1300.0,
```

which is greater than the initial log-likelihood without considering a Bayesian network for having the joint probabilities.

Appendix 1:

Python code:

```
import numpy as np
import matplotlib.pyplot as plt
import xlrd
import pandas as pd
from statistics import mean, stdev, variance
from numpy.linalg import inv
import math
from math import log10, floor

print("UBitName = amirbagh")
print("personNumber = 50135018")

def round_sig(x, sig=3):  # rounding to 3 significant digits
    size = int(math.sqrt(x.size))
    x_round = np.zeros(shape=(size,size))
    for i in range(0,size):
        for j in range(0,size):
            if size == 1:
                x_round = round(x, sig-int(floor(log10(abs(x))))-1)
            if size > 1:
                x_round[i,j] = round(x[i,j], sig-int(floor(log10(abs(x[i,j
])))))-1)

    return x_round

# read data file
# please change this to your specific directory for reading the file
#df = pd.read_excel('C:/Users/Amir/Desktop/Project 1/proj1code/university
data.xlsx')

data_text = """
    5 57 400400 25064;
    4.6 58.6 512500 30228;
    4.5 54.5 550000 33513;
    4.3 55.9 440000 30698;
    4.3 55 628190 34722;
    4.2 53 437000 26660;
    4.1 54 416000 35580;
    4.1 55 603357 41811;
    4 55 376827 36180;
    4 52 459000 29720;
    3.7 55 202487 28804;
    3.6 59 363605 28813;
    3.6 52 482515 33624;
    3.4 54.5 571668 30452;
    3.4 54.5 392200 37635;
    3.4 52 610000 20876;
    3.4 58 485000 42184;
    3.3 54 851303 26537;
    3.3 55 526549 28591;
```

```
    3.3 57 400000 36774;
    3.3 53.5 315000 36624;
    3.1 58 644455 21550;
    3.1 48.5 425000 26356;
    3.1 53 475000 28379;
    3.1 53.5 389000 33151;
    3.1 49 358850 25267;
    3.1 61 496688 27444;
    3 54.5 566200 23312;
    3 51.5 464946 23551;
    3 50 525166 28591;
    2.9 56 544848 33241;
    2.9 54 580000 28168;
    2.8 53.5 520000 34980;
    2.8 54 188294 36286;
    2.8 53.5 310000 36276;
    2.7 59.8 411752 26030;
    2.6 50 448800 20816;
    2.6 59.5 643309 21550;
    2.6 51 493272 27409;
    2.6 45 403337 30888;
    2.5 49.9 499194 29960;
    2.5 46 485088 23540;
    2.4 47 332100 39360;
    2.4 48.7 421000 26077;
    2.4 50.5 423074 21642;
    2.4 51 341053 21388;
    2.4 49 394956 29696;
    2.4 53 518279 30378;
    2.4 51 662500 25510
"""
data = np.array(np.matrix(data_text))

mu = np.mean(data, axis=0)
var = np.var(data, axis=0)
sigma = np.std(data, axis=0)

# part 1
#get the values for a given column and set a variable for each value
score = data[:,0]
mu1 = round_sig(mu[0], sig=3)
var1 = round_sig(var[0], sig=3)
sigma1 = round_sig(sigma[0], sig=3)

overhead = data[:,1]
mu2 = round_sig(mu[1], sig=3)
var2 = round_sig(var[1], sig=3)
sigma2 = round_sig(sigma[1], sig=3)

adminpay = data[:,2]
mu3 = round_sig(mu[2], sig=3)
var3 = round_sig(var[2], sig=3)
sigma3 = round_sig(sigma[2], sig=3)
```

```
tuition = data[:,3]
mu4 = round_sig(mu[3], sig=3)
var4 = round_sig(var[3], sig=3)
sigma4 = round_sig(sigma[3], sig=3)

print("mu1 =", mu1)
print("mu2 =", mu2)
print("mu3 =", mu3)
print("mu4 =", mu4)

print("var1 =", var1)
print("var2 =", var2)
print("var3 =", var3)
print("var4 =", var4)

print("sigma1 =", sigma1)
print("sigma2 =", sigma2)
print("sigma3 =", sigma3)
print("sigma4 =", sigma4)

# part 2
# calculating covariance and correlation matrices
covarianceMat = round_sig(np.cov(data.T), sig=3)
correlationMat = round_sig(np.corrcoef(data.T), sig=3)

print("covarianceMat = ")
print(covarianceMat)

print("correlationMat = ")
print(correlationMat)

# plotting the correlation matrix
plt.matshow(correlationMat)

label = ['score', 'overhead', 'adminpay', 'tuition']
x_pos = np.arange(len(label))
plt.xticks(x_pos,label)

y_pos = np.arange(len(label))
plt.yticks(y_pos,label)

figure_title = 'Correlation between ranking criteria'
plt.text(1.5, -1.08, figure_title,
         horizontalalignment='center',
         fontsize=12)

plt.colorbar()
plt.show()

# part 3

mean = mu
cov = covarianceMat
```

```
llh_each_row = -((data-mu)/sigma)**2. / 2. - np.log((2.*math.pi)**0.5*sigm
a)
llh = np.sum(llh_each_row, axis=0)

print("logLikelihood = ",  round_sig(np.sum(llh), sig=3))

# part 4
# loglikelihood using Bayesian network

def BN_llh(children, parents):
    N = data.shape[0]
    all_ones = np.ones([N, 1])
    Phi = np.hstack([all_ones, data[:, parents]])
    Y = data[:, children]
    Phi_T = np.transpose(Phi)
    beta = np.linalg.solve(np.dot(Phi_T, Phi), np.dot(Phi_T, Y))
    sigma_square = np.mean([
        np.square(np.dot(np.transpose(beta), Phi[n, :]) - Y[n]) for n in r
ange(N)
    ])
    llh = -N/2. * (np.log(2. * math.pi * sigma_square) + 1.)
    return beta, sigma_square, llh

# Binary graph showing the relation between variables (Bayesian network)
BNgraph = np.matrix([[0, 0, 0, 0],
                     [1, 0, 1, 0],
                     [0, 0, 0, 0],
                     [1, 1, 1, 0]])
print("BNgraph = ")
print(BNgraph)

# joint probabilities using Bayesian network
N = 49
# X = [CS Score, Research Overhead, Admin Base Pay, Tuition]
# p(adminpay|tuition,overhead)
BN_llh_1 = BN_llh([2], [3,1])[2]

# p(tuition)
BN_llh_2 = llh[3]

# p(overhead|tuition)
BN_llh_3 = BN_llh([1], [3])[2]

# p(score|tuition,overhead)
BN_llh_4 = BN_llh([0], [3,1])[2]

# log-likelihood from Bayesian network relations
BNlogLikelihood = round_sig(BN_llh_1 + BN_llh_2 + BN_llh_3 + BN_llh_4, sig
=3)
print("BNlogLikelihood = ", BNlogLikelihood)
```

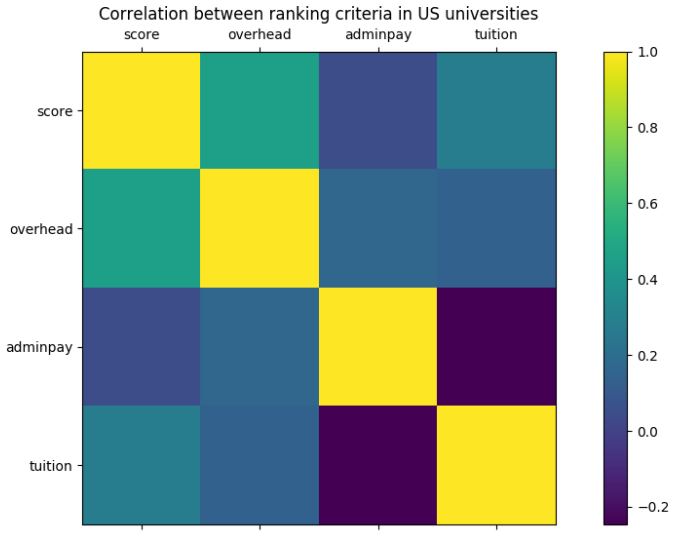
Code results:

```
UBitName = amirbagh
personNumber = 50135018

mu1 = 3.21
mu2 = 53.4
mu3 = 469000.0
mu4 = 29700.0
var1 = 0.448
var2 = 12.6
var3 = 13900000000.0
var4 = 30700000.0
sigma1 = 0.669
sigma2 = 3.55
sigma3 = 118000.0
sigma4 = 5540.0

covarianceMat =
[[  4.57000000e-01   1.11000000e+00   3.88000000e+03   1.06000000e+03]
 [  1.11000000e+00   1.29000000e+01   7.03000000e+04   2.81000000e+03]
 [  3.88000000e+03   7.03000000e+04   1.42000000e+10  -1.64000000e+08]
 [  1.06000000e+03   2.81000000e+03  -1.64000000e+08   3.14000000e+07]]

correlationMat =
[[ 1.      0.456   0.0482  0.279 ]
 [ 0.456   1.      0.165   0.14  ]
 [ 0.0482  0.165   1.     -0.245 ]
 [ 0.279   0.14   -0.245   1.    ]]
```



```
logLikelihood =  -1320.0


BNgraph =
[[0 0 0 0]
 [1 0 1 0]
 [0 0 0 0]
 [1 1 1 0]]

BNlogLikelihood =  -1300.0
```