Project-1

Probability Distributions and Bayesian Networks

Name: Armaan Goyal
UBIT Name: armaango
Person Number: 50170093

Probability Distributions and Bayesian Networks

Introduction to Probability Distributions

Uncertainty is the key concept in the field of pattern recognition. It arises both through noise on measurements, as well as through the finite size of data sets. Probability theory provides a consistent framework for the quantification and manipulation of uncertainty and forms one of the central foundations for pattern recognition.[1]

In order to use the concepts of probabilities in the field of pattern recognition and machine learning, the knowledge of and the ability to apply the concepts of Probability distributions can prove quite helpful. Practically applying probability theory, the probability distributions can be defined in several ways as follows,

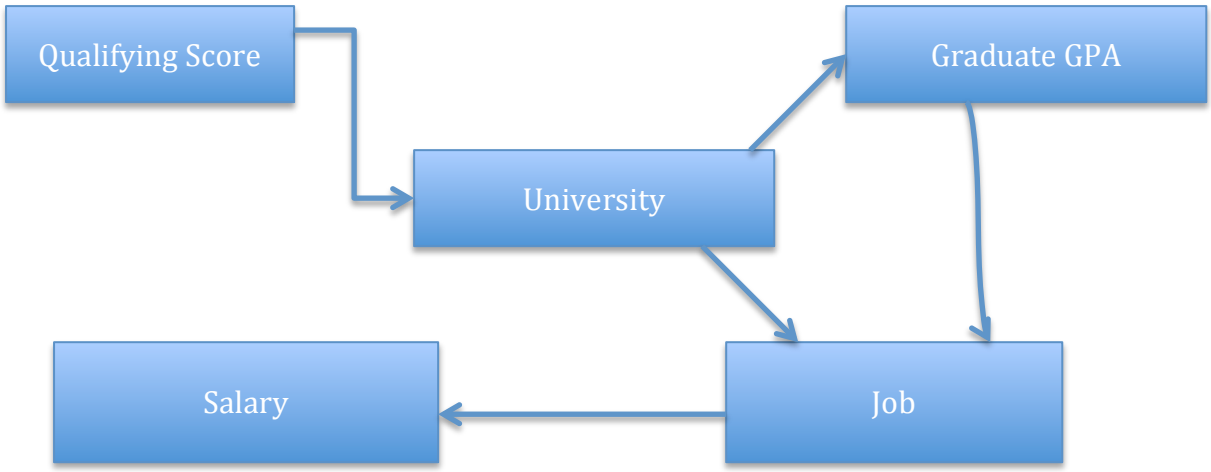
- by giving a valid probability mass function or probability density function
- by giving a valid cumulative distribution function or survival function
- by giving a valid hazard function
- by giving a valid characteristic function
- by giving a rule for constructing a new random variable from other random variables whose joint probability distribution is known.[1]

A probability distribution can either be univariate or multivariate. A univariate distribution gives the probabilities of a single random variable taking on various alternative values; a multivariate distribution (a joint probability distribution) gives the probabilities of a random vector—a set of two or more random variables—taking on various combinations of values.[1] During our project we worked on the given data set first considering the distribution to be univariate and then later taking into account the multivariate nature of the data, we created the required Bayesian Network and calculated its log likelihood.

Introduction to Bayesian Networks:

A Bayesian network is a probabilistic graphical model that is used to depict a set of random variables and their conditional dependencies via a directed acyclic graph (DAG)[2]. For example, a Bayesian network could represent the probabilistic relationships between the education, degree, job and salary of a large group of people. Such a network could be used

to determine or predict the end result given the various factors that affect the probability of the output.



The above figure denotes an example Bayesian Network showing dependencies of various factors on the final output i.e. Salary.

Problem Introduction:

The project's primary aim is to introduce us to the use of MATLAB software to evaluate different statistical variables like **'mean' , 'variance', 'co-variance', 'standard deviation' , 'correlation**'. These variables are then to be used to calculate loglikehood of the given data set. Further we need to use the calculated values to construct compact representations of joint probability distributions known as Bayesian networks. These networks will then be evaluated by calculating their likelihood and comparing that with our initial calculation and finally identify an optimal network that satisfies some basic given requirement.

Dataset:

The dataset provided for the analysis was a data of around 50 US universities with certain general defining characteristics like the **Computer Science Ranking Score, Research Overhead(as a percentage), Administration Base Salary(in $), Tuition Out State(in $) and Number of students in the program**. The data has been obtained from the top 100 Computer Science graduate programs in the US according to the **US News and World Report** website, **Chronicle.com** from each **department's web page** and provided in an xlsx format.

Solution Approach:

The given problem was divided into several parts and each problem was to be approached step by step.

Data Import and manipulation:
The first step required us to import the given data set into our workspace or our script, which can be accomplished using the inbuilt **xlsread()** function. Further we extract the required 4 variable vectors from the imported data using matrix splicing and we get 4 matrices of the order 49*1 and are labeled as V1 to V4.

Basic Computations:
These vectors are fed as input to several pre defined functions in MATLAB to get the required outputs like mean, variance, standard deviation, covariance matrix and correlation matrix.

The next step is to plot the pairwise scatter plots of the given data set variables:

- **CSScoreUSNews**
- **Research Overhead**
- **AdminBasePay**
- **Tuitionoutstate**

We use the **scatter()** function to plot the graphs of all possible pairs of the above variables and obtain 6 scatter graphs which are as follows:

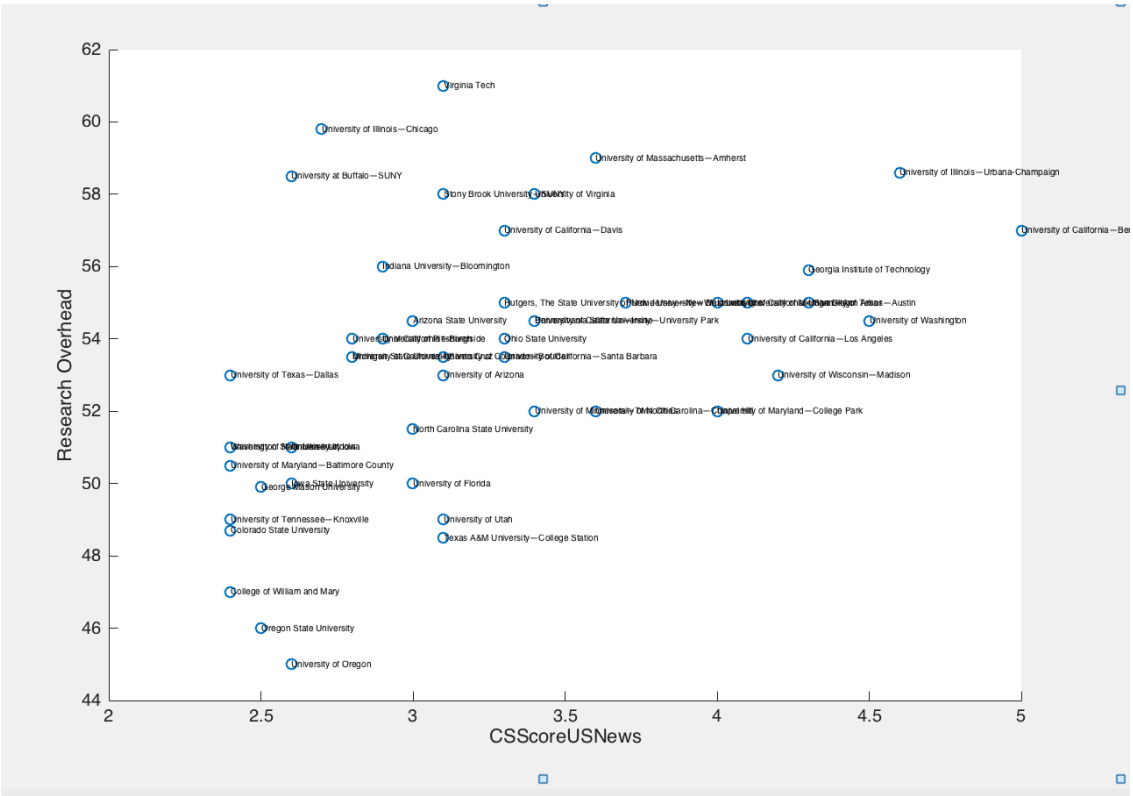


Fig 1. Plot of CS Score US News vs Research Overhead

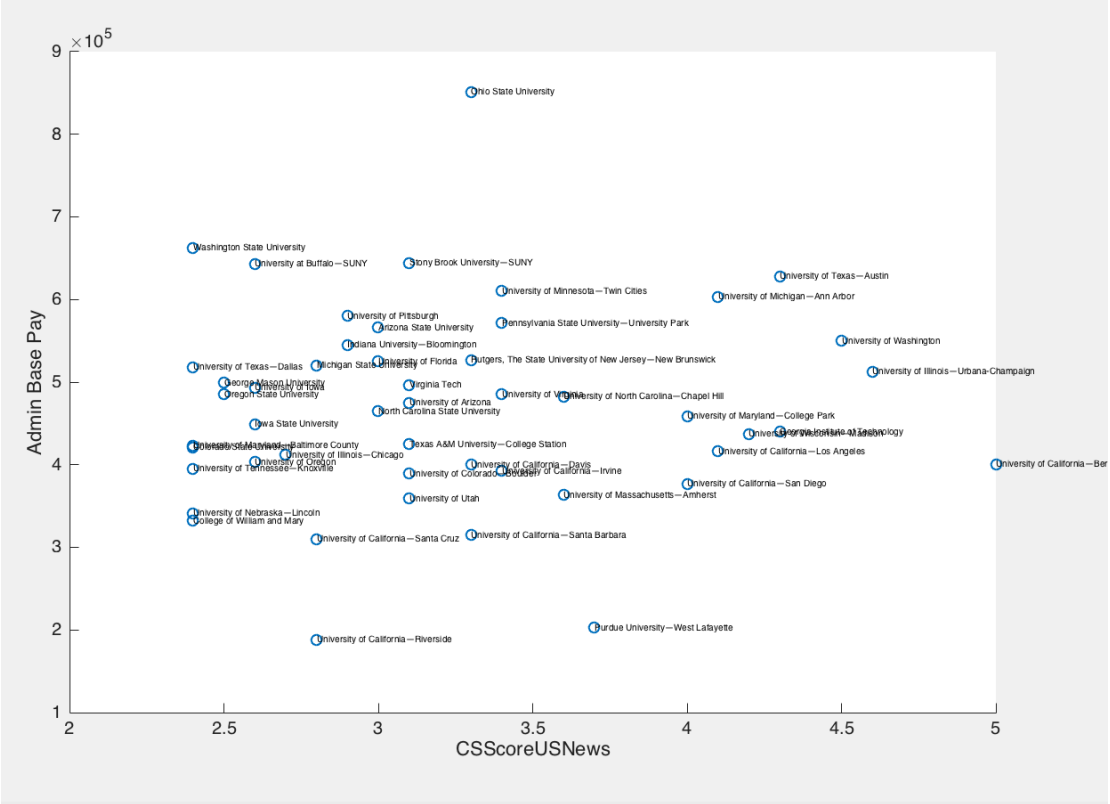


Fig 2. Plot of CS Score Us News vs Admin Base Pay

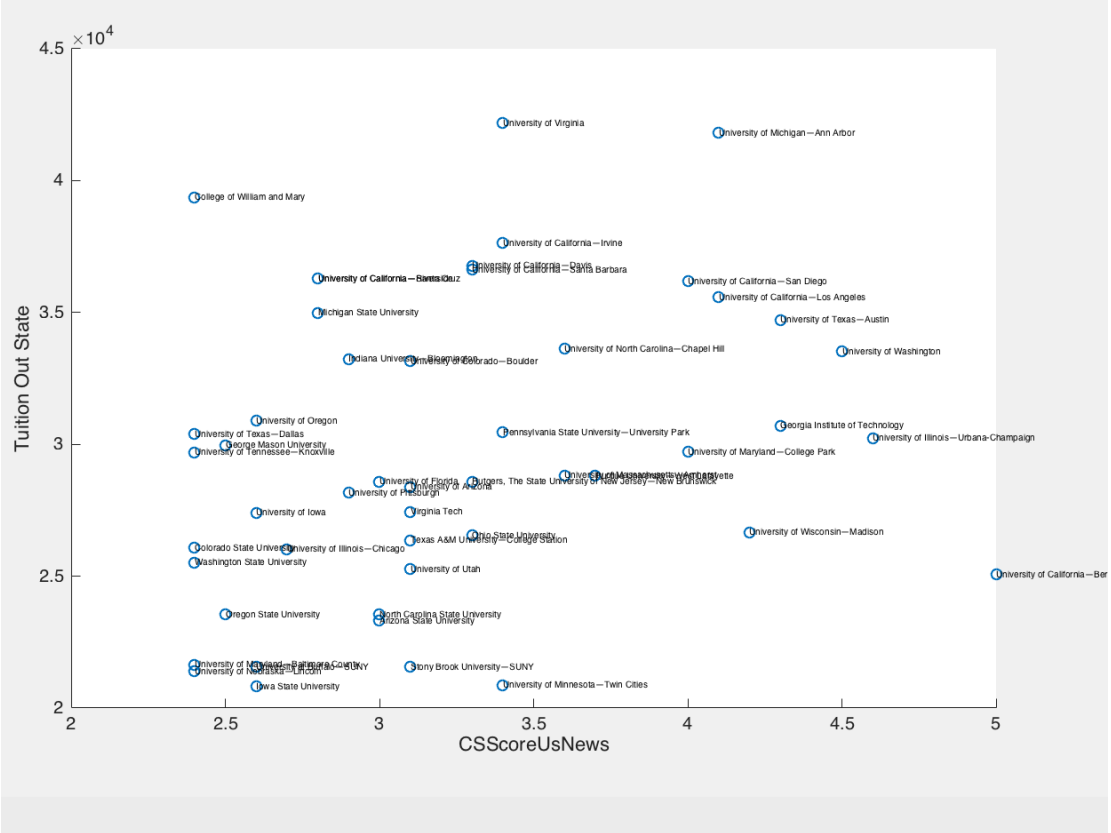


Fig 3. Plot of CS Score US News vs Tuition Out State

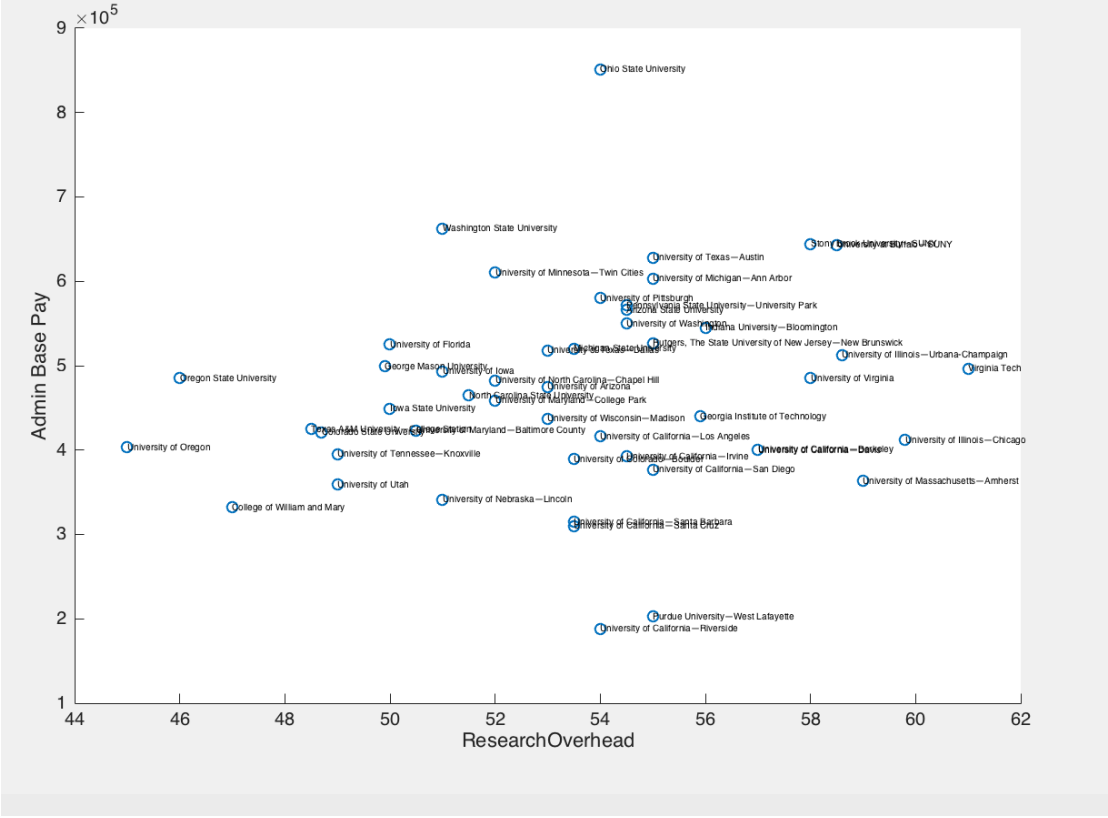


Fig 4. Plot of Research Overhead vs Admin Base Pay

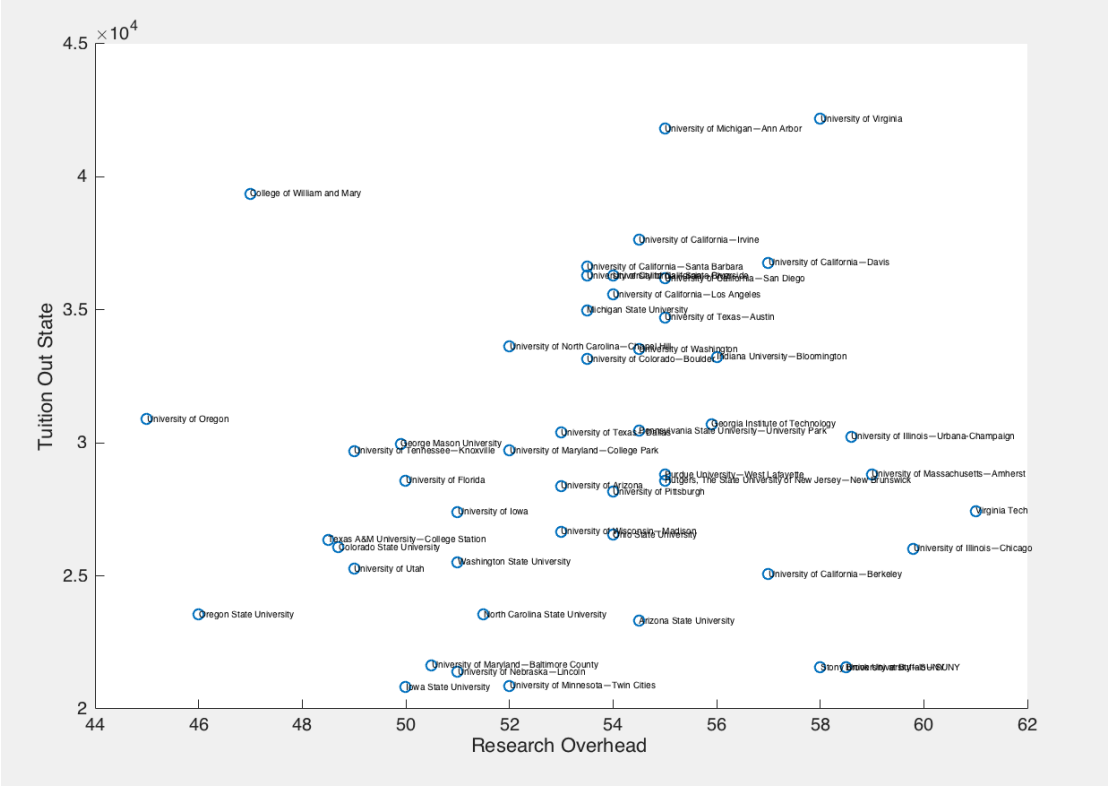


Fig 5. Plot of Research Overhead vs Tuition Out State

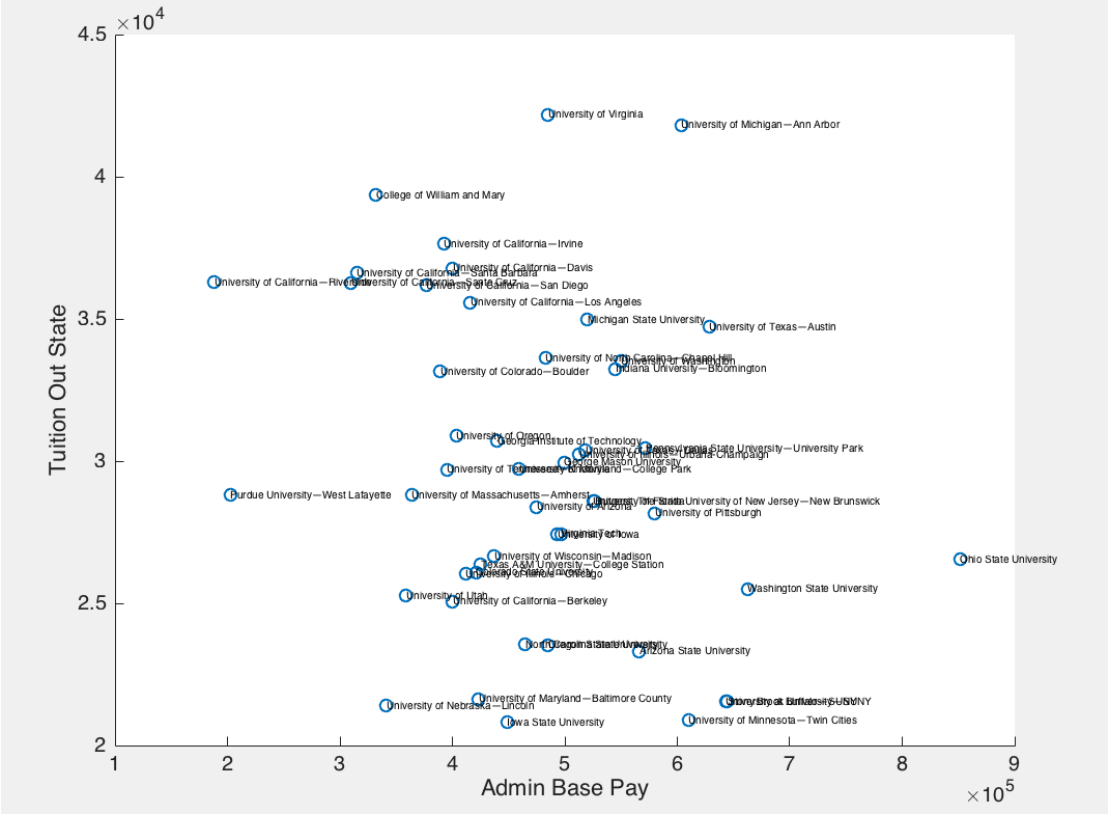


Fig 6. Plot of Admin Base Pay vs Tuition Out State

We are asked to find the most correlated and least correlated pairs of variables, which can be deduced by reading the correlation matrix generated earlier.

On reading the matrix, we figure out the following:

Most correlated pair- V1 and V2 i.e. CSScoreUSNews and ResearchOverhead
Least correlated pair-V1 and V3 i.e. CSSCoreUSNews and AdminBasePay

Log Likelihood calculation assuming Independent data:

Likelihood is the hypothetical probability that an event that has already occurred would yield a specific outcome. The concept differs from that of a probability in that a probability refers to the occurrence of future events, while likelihood refers to past events with known outcomes. [3]

For this step of the project we are required to assume the given data set variables to be independent of each other or univariate and compute the log likelihood of the data. (We use log likelihood for ease of computation when compared to likelihood)
This is performed by calculating the probability density of each vector independently using the **normpdf()** function and then taking the log of each of those values and summing the computed values. This can be accomplished by using a simple for loop for each of the 4 data set variables.

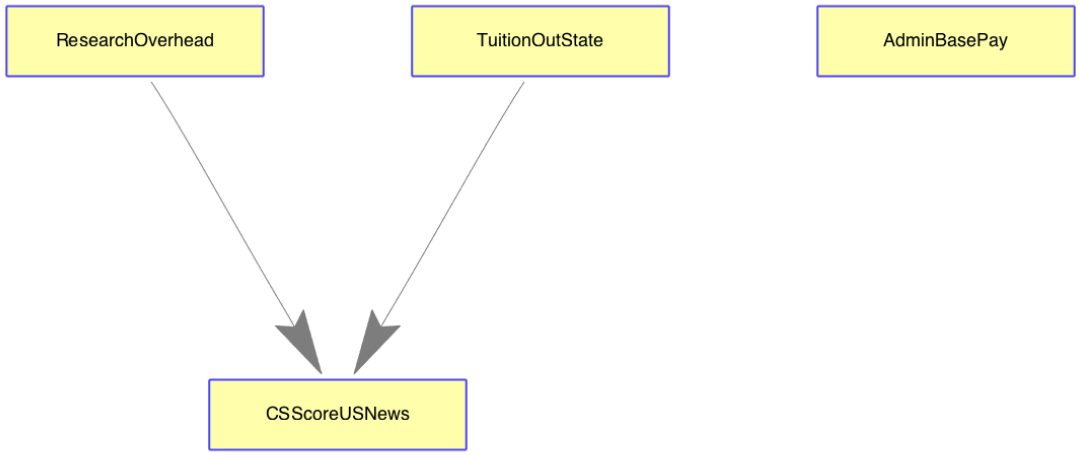
Creating Bayesian Networks and log likelihood calculation

Based on the correlation values obtained by the correlation matrix computed earlier, we sort the pairs of variables in a decreasing order of correlation values. We now know that the most correlated pair will definitely have an edge from one variable to the other and the least correlated pairs can be left as unrelated and hence not linked. Based on our analysis of the numerical values of the following data:

Pairwise Data	**Correlation values**
V1 – V2	0.4655
V1 – V3	0.0482
V1 – V4	0.2794
V2 – V3	0.1575
V2 – V4	0.1496
V3 – V4	-0.2453

Where V1-CS Score US News
V2-Research Overhead
V3-Admin Base Pay
V4-Tuition out state

We can see the order of correlation of the various variables. We can construct a sample Bayesian Network using this data and assuming the directions based on the analysis of the given data set.



This is a valid Bayesian Network and it can be represented in the form of a matrix of 0s and 1s as follows.

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

This is a valid BN but not the optimal solution as required by the loglikelihood comparison. We can calculate the log likelihood of this network using multivariate probability distribution of correlated pairs and then taking the log and adding those values. But here we need to be careful to include conditional probabilities, as the variables are dependent on each other.
For our assumed graph the log likelihood calculation can be done by the following formula

$$\text{logLikelihood} = \text{Sum}(\log(P(V2) * P(V4) * P\left(\frac{V1}{V2,V4}\right))$$

For calculating the conditional probabilities we use the **mvnpdf()** function instead of the **normpdf()** function.
The output obtained is the **BNlogLikelihood**.

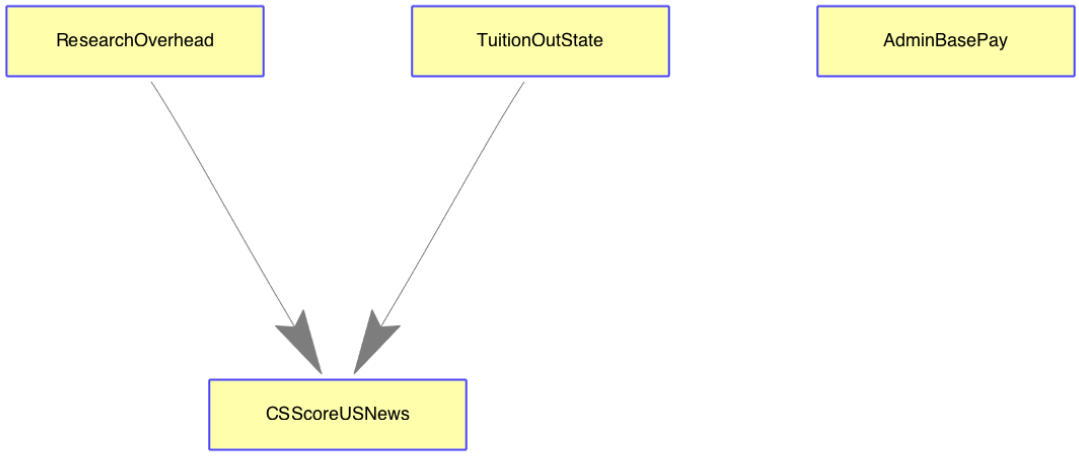
Optimal solution generation and results:

There are several ways to calculate the optimal solution. One method is brute force or hit and trial where we keep assuming BN graphs and keep calculating the log likelihood of each graph to find which one gives the highest value.

For all such graphs we try and calculate the log likelihood and keep comparing that with our given log likelihood of the data that we computed earlier. If the log likelihood of the BN graph matrix is larger we can say that it is an optimum solution and the matrix thus obtained represents an optimum Bayesian Network.

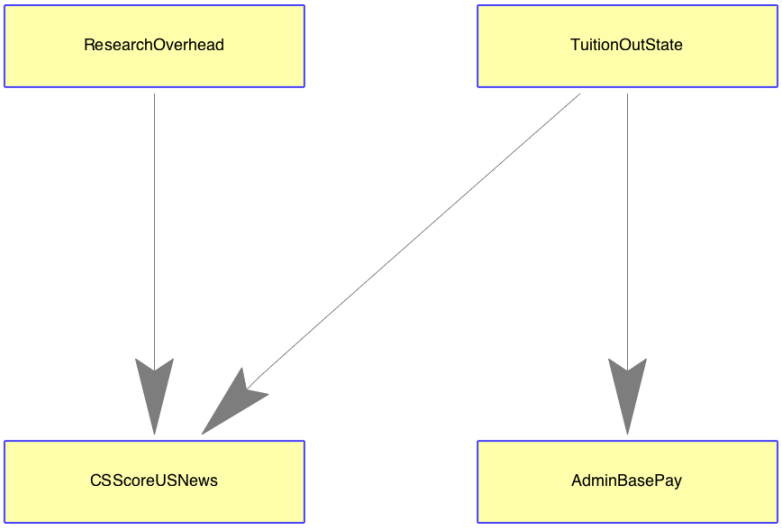
I tried the hit and trial approach for 3 such graphs as shown below, each time either including a lesser-correlated pair or changing the direction of the arrows and submitted them to the auto grader script to see if I get the required optimal solution.

For the first two iterations the log likelihood value obtained was not optimal.



Graph-1

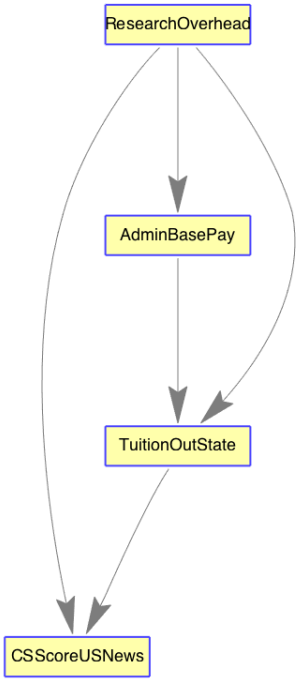
BNgraph = [0 0 0 0; 1 0 0 0; 0 0 0 0; 1 0 0 0]



Graph-2 BNgraph = [0 0 0 0; 1 0 0 0; 0 0 0 0; 1 0 1 0]

For the third graph I got the optimal likelihood value as specified by the auto grader.
(This optimal matrix is the matrix I used in my code to generate the loglikelihood values)

The plot of the 3rd matrix is as below:



Graph-3 BNgraph = [0 0 0 0; 1 0 1 1; 0 0 0 1; 1 0 0 0]

The mathematical notation for calculating loglikelihood is as follows:

$$\text{logLikelihood} = \text{Sum}(\log(P(V2) * P\left(\frac{V3}{V2}\right) * P\left(\frac{V4}{V3,V2}\right) * P\left(\frac{V1}{V2,V4}\right))$$

Conclusion:

From the analysis of the above computations we can observe that there may be more than 1 Bayesian Networks that provide the optimal loglikelihood value and they can be obtained by hit and trial. Also it can be observed that if we keep on increasing the connections, the value of loglikelihood starts moving towards the optimal value.

References:

[1] **https://en.wikipedia.org/wiki/Probability_distribution**

[2] **https://en.wikipedia.org/wiki/Bayesian_network**

[3] **http://mathworld.wolfram.com/Likelihood.html**