CSE474/574: Introduction to Machine Learning(Fall 2017)

Instructor: Sargur N. Srihari
Teaching Assistants: Jun Chu, Junfei Wang, Hengtong Zhang, Tianhang Zheng

**********September 13, 2017**********

Project 1: Probability Distributions and Bayesian Networks
Due Date: Monday, September 25

1 Overview

Machine Learning methods are based on probability theory and statistics. This project concerns probability distributions of several variables. We will use Python to evaluate sufficient statistics: mean and variance of univariate distributions and covariance and correlation coefficient of pairs of variables. We will then use these statistics to construct compact representations of joint probability distributions known as Bayesian networks. Then we will evaluate the goodness of these representations by using the concept of likelihood. Finally we will use the Bayesian networks to answer some queries.

For this project, students are required to submit a project report and the complete Python code they write to generate the results. Students are encouraged to try and implement all kinds of numerical and statistical experiments and show interesting findings using plots or tables. It is also encouraged to include numerical programming tricks in the report. Students are free to reference any methods or formulas in the project description. Students should highlight the creative parts in their report and not repeat what is already in the project description. Students will get bonus points if it is shown that they have done some innovative work.

1.1 Methods

Mathematical expressions necessary for this project are given in Appendix 1 (Section 5 of this project description.

The data set for this task are consists of multivariate data (vector) as described in Appendix 2 (section 6). Each data vector has a label associated with it.

2 Task

1. Compute for each variable ((CS Score, Research Overhead, Admin Base Pay, Tuition)) its sample mean, variance and standard deviation.
   Related variables: `mu1`, `mu2`, `mu3`, `mu4`, `var1`, `var2`, `var3`, `var4`, `sigma1`, `sigma2`, `sigma3`, `sigma4`

2. Compute for each pair of variables their covariance and correlation. Show the results in the form of covariance and correlation matrices. Also make a plot of the pairwise data showing the label associated with each data point. Which are the most correlated and least correlated variable pair?
   Related variables: `covarianceMat`, `correlationMat`

3. Assuming that each variable is normally distributed and that they are independent of each other, determine the log-likelihood of the data (Use the means and variances computed earlier to determine the likelihood of each data value.)
   Related variables: `logLikelihood`

4. Using the correlation values construct a Bayesian network which results in a higher log-likelihood than in 3.
   Related variables: `BNgraph`, `BNlogLikelihood`

5. Using the Bayesian network to determine some interesting conditional probabilities.

3 Deliverables

There are three parts in your submission:

1. Report

   The report describes your implementations and results using graphs, tables, etc. Write a concise project report, which includes a description of how you obtained the Bayesian network(s). Your report should be edited in PDF format. Additional grading considerations will include creativity in interpreting your statistics, and the clarity and flow of your report. Highlight the innovative parts and do not include what is already in the project description. You should also include the printed out results from your code in your report.

   Submission:

   Submit the PDF on a CSE student server with the following script:

   `submit_cse474 proj1.pdf` for undergraduates

   `submit_cse574 proj1.pdf` for graduates

   In addition to the PDF version of the report, you also need to hand in the hard copy version on the first class after due date or else your project will not be graded.

2. Code

   The code for your implementations. Code in Python is the only accepted one for this project. You can submit multiple files, but the name of the entrance file should be main.py. All Python code files should be packed in a ZIP file named `proj1code.zip`. After extracting the ZIP file and executing command `python main.py` in the first level directory, the program should print all the related variables according the following format:

   ```
   UBitName = jchu6
   personNumber = XXXXXXXX
   mu1 = 0.23
   mu2 = 0.45
   ```

```
...
covarianceMat =
[[0.1 0.2]
 [0.4 0.5]]
...
```

Floating point numbers should be rounded to 3 significant figures.

The following are the variables you need to include in the printed results.

`UBitName`: Your UBIT name.

`personNumber`: Your person number.

`mu1`, `mu2`, `mu3`, `mu4`, `var1`, `var2`, `var3`, `var4`, `sigma1`, `sigma2`, `sigma3`, `sigma4`: All scalars.

`covarianceMat`, `correlationMat`: Two 4-by-4 matrices.

`logLikelihood`: A scalar.

`BNgraph`: A 4-by-4 matrix representing the acyclic directed graph showing the connections of the Bayesian network. Each entry of the matrix takes value 0 or 1.

`BNlogLikelihood`: A scalar showing log-likelihood produced by your Bayesian network. The higher it is, the better score you get. Of course, it must match with the structure of of network.

Submission:

Submit the Python code on a CSE student server with the following script:

`submit_cse474 proj1code.zip` for undergraduates

`submit_cse574 proj1code.zip` for graduates

4 Due Date and Time

The due date is **September 25, 11:59PM**. After finishing the project, you may be asked to demonstrate it to the TAs if your results and reasoning in your report are not clear enough.

5 Appendix 1: Useful Mathematical Formulas

5.1 Mean, Variance and Standard Deviation

The sample mean μ of a univariate distribution of variable X with N samples $x(i), i = 1, ..N$ has the form

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x(i) \tag{1}$$

The sample variance σ^2 is computed as

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} [x(i) - \mu]^2 \tag{2}$$

where σ is referred to as the standard deviation.

Corresponding Python functions:
numpy.mean() : Average or mean value of array;
numpy.var() : Variance;
numpy.std() : Standard deviation;

5.2 Statistics of a pair of variables

The sample covariance of a pair of variables X_1, X_2 with samples $x_1(i), x_2(i), i = 1, ..N$ is

$$\sigma_{12} = \frac{1}{N-1} \sum_{i=1}^{N} [x_1(i) - \mu_1][x_2(i) - \mu_2] \tag{3}$$

The correlation coefficient is the normalized covariance given by

$$\rho_{12} = \frac{\sigma_{12}}{\sigma_1 \sigma_2} \tag{4}$$

The sample covariance matrix of a set of d variables $\mathbf{X} = \{X_1, ..X_d\}$ is

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \cdots & \sigma_{2d} \\ \sigma_{13} & \sigma_3^2 & \sigma_3^2 & \cdots & \sigma_{3d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{1d} & \sigma_{2d} & \sigma_{3d} & \cdots & \sigma_d^2 \end{bmatrix}$$

Corresponding Python functions:
numpy.cov() : Covariance;
numpy.corrcoef() : Correlation coefficients;

5.3 Normal Density

The Gaussian (or normal) distribution of a continuous random variable X with mean μ and variance σ^2, denoted as $x \sim \mathcal{N}(\mu, \sigma^2)$, has a probability density function (pdf) of the form

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{5}$$

The multivariate form of this distribution for a vector $\mathbf{x}$ of d variables, mean vector μ and covariance matrix Σ, denoted $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$ is

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right] \tag{6}$$

Corresponding Python functions:
scipy.stats.norm.pdf : Normal probability density function;

5.4 Normalization

For a univariate population that is normally distributed and known mean and standard deviation, it is useful to convert it to a standard normal distribution $\mathcal{N}(0,1)$ by replacing X by $\frac{X-\mu}{\sigma}$.

5.5 Cumulative Distribution Function (cdf)

A probability can be determined from a cdf, which in turn can be determined from a pdf as follows:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(x)dx \tag{7}$$

Thus a probability of X within a small interval $\pm\delta$ is:

$$P(X - \delta \leq X \leq X + \delta) = F(x + \delta) - F(x - \delta) \tag{8}$$

The multivariate version of cdf is straight-forward. For example, with two variables X_1 and X_2 with pdf $p(x_1, x_2)$ the cdf is

$$F(x_1, x_2) = P(X_1 \leq x_1, X_2 \leq x_2) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} p(x_1, x_2)dx \tag{9}$$

5.6 Log-likelihood function

Given N independent samples $\mathbf{x}_1, ..\mathbf{x}_N$ from a probability distribution $p(\mathbf{x})$, the log-likelihood of observing the samples is given by

$$\mathcal{L}(\mathbf{x}_1, ..\mathbf{x}_N) = \sum_{i=1}^{N} log\ p(\mathbf{x}_i) \tag{10}$$

Note that the log-likelihood is a scalar value. Assume that the four variables are continuous and satisfy the Gaussian distribution.

5.7 Conditional Probabilities

Given two variables X_1 and X_2 the *sum rule* of probability is:

$$p(x_1) = \sum_{Val(x_2)} p(x_1, x_2) \tag{11}$$

where Val is the set of values taken by its argument. The sum rule allows us to obtain the marginal probability $p(x_1)$ from the joint probability $p(x_1, x_2)$.

The *product rule* of probability is:

$$p(x_1, x_2) = p(x_1|x_2)p(x_2) \tag{12}$$

from which we get the *chain rule*

$$p(x_1, x_2, x_3) = p(x_1|x_2, x_3)p(x_2|x_3)p(x_3) \tag{13}$$

5.8 Bayesian Network Factorization

Given a Bayesian network G of N variables $\mathbf{X} = \{X_1, ..X_d\}$, the joint probability distribution is given by

$$p(\mathbf{X}) = \prod_{i=1}^{N} p(X_i|pa(X_i)) \tag{14}$$

where $pa(X_i)$ are the parent variables of X_i.

6 Appendix 2: Data Set

Data for this project is multivariate., with four variables X_1, X_2, X_3, X_4. There is a fifth variable X_5 which has missing values. They are provided to you separately as an Excel spreadsheet titled UniversityData.xls.

For your interest, most of this data was obtained from the following data sources:

1. X_1 =CS-ranking-score:
   Each value corresponds to the score of a public university according to a survey (They are a subset of the top 100 Computer Science graduate programs in the US according to the US News and World Report). See
   `http://grad-schools.usnews.rankingsandreviews.com/best-graduate-schools/top-science-schools/computer-science-rankings`

2. X_2 =Research-Overhead (percentage):
   These correspond to the portion of research grants retained as infrastructure/administrative costs by the university.
   From each university's website

3. X_3 =Administrator Base Salary ($):
   `http://chronicle.com/factfile/ec-2015/#id=table_public_2014`
   (This link may not work if outside campus)

4. X_4 = Tuition (Out-of-State) ($):
   `http://colleges.usnews.rankingsandreviews.com/best-colleges/rankings/`

5. X_5 = No. of CS Graduate Students in Fall 2015:
   Mostly from each department's website

Note: If you find that any data entered is inaccurate, please inform the instructor and we will update the data set for the entire class.