EEP596A Advanced Intro. to Machine Learning | **University of Washington**
Winter 2021 | **Dept. of Electrical & Computer Engineering**

Homework 1. Due **Saturday Jan 23rd, 11:59pm** Electronically

Prof: J. Bilmes <bilmes@uw.edu>
TA: Lilly Kumari <lkumari@uw.edu>

Friday, Jan 15th 2021

General Instructions.

This homework consists of two parts, a write-up part that needs to be turned in using a single pdf file, and also a programming part that needs to be turned in using a zipped Jupyter notebook file (or files). A Jupyter notebook file has extenion `.ipynb` and a zip file has extension `.zip` and should contain one or more `.ipynb` files.

Doing your homework by hand and then converting to a PDF file (by say taking high quality photos using a digital camera and then converting that to a PDF file) is fine, as there are many jpg to pdf converters on the web. Alternatively, you are welcome to use Latex (great for math and equations), Microsoft Word, and Google Docs, or hand-written paper, as long as the final submitted format is a single pdf.

For the plots requested in the programming session, you can either save them as pictures and insert them manually into the writeup, or directly export the completed jupyter notebook to a pdf file (in jupyter notebook, "File→Download as→PDF via LaTex") and copy it in to your writeup.

Some of the problems below might require that you look at some of the lecture slides at our web page (https://canvas.uw.edu/courses/1431528).

Note that the due dates and times are often in the evenings.

As mentioned above, for the programming problems, you need to submit your code (written in python as a Jupyter notebook) and the answers to the non-coding questions should also be included in the pdf write-up. Your code answers must to be in python, no other language is accepted.

Neatness and clarity count! : Answers to your questions must be clearly indicated in all cases. Not only correctness, but clarity and completeness is necessary to receive full credit. Justify you answers. A correct answer does not guarantee full credit and a wrong answer does not guarantee poor credit, hence show all work and justify each step, thinking "clarity" and "neatness" along the way. If we can't understand your answer, or if your answers are not well and neatly organized, you will not receive full credit.

All homework is due electronically via the link https://canvas.uw.edu/courses/1431528/assignments. This means that on canvas you turn in two files: (1) **a pdf file** with answers to the writeup questions, and (2) **a zip file with python code (in jupyter notebook files)**. **Please do not submit any fewer or any more than these two files.**

Problem 1. Simpson's Paradox [30 points]

Imagine you and a friend are playing the slot machine in a casino. Having played on two separate machines for a while, you decide to swap machines between yourselves, and measure for differences in "luck". The wins/losses for you and your friend for each machine are tabulated below.

Machine 1	Wins	Losses
You	40	60
Friend	30	70

Machine 2	Wins	Losses
You	210	830
Friend	14	70

Assuming that the outcomes of playing the slot machine are independent of their history and that of the

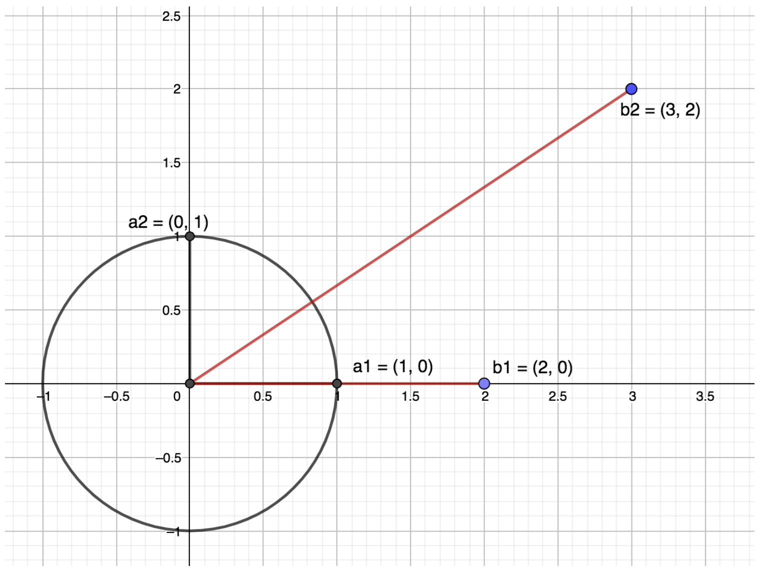

Figure 1: Problem 2

other machine, answer the following questions.

Problem 1(a). (10 points) Estimate the winning probability of you and your friend for each of the machines. Compare your winning probability with your friend's on different machines. Who is more likely to win ?

Problem 1(b). (5 points) Estimate the overall winning probability of you and your friend in the casino (assume that there are only two slot machines in the casino). Who is more likely to win ?

Problem 1(c). (10 points) Compare your conclusions from (1) and (2). Can you explain this result theoretically (Hint: write down the relationship between the probabilities in (1) and (2))?

Problem 1(d). (5 points) When will the conclusions of (1) and (2) be the same ?

Problem 2. Linear Algebra Refresher [30 points] On the 2D plane, we have four vectors (all with the starting point on the origin) $a_1 = (1, 0)$, $a_2 = (0, 1)$, $b_1 = (2, 0)$ and $b_2 = (3, 2)$. See Fig. 1

Problem 2(a). (5 points) Write down one 2×2 matrix W that transforms a_1 to b_1, and a_2 to b_2. v

Problem 2(b). (5 points) Write down one rotation matrix V, which rotates clockwise by α degrees such that $tan(\alpha) = 2$. Then, write down one matrix Σ, which scales the x-axis by 4. Finally, write down one rotation matrix U, which rotates counter-clockwise by β degrees, such that $tan(\beta) = 1/2$. Multiply three matrices together, namely $U\Sigma V$, what do you discover?

Problem 2(c). (10 points) Compute the eigenvalues and the corresponding eigenvectors of W^TW. Now consider the unit circle. Suppose every point on the unit circle gets transformed by W, what do you get after the transformation (consider the matrix factorization in the previous question)? What relationship do you find between the eigenvalues and the transformed shape?

Problem 2(d). (10 points) Compute the determinant of W. What's the area of the shape transformed from the unit circle? Can you think of some hypothesis about the relationship between the determinant and the area of a transformed shape? Based on such hypothesis, can you use one sentence to explain that the determinant of a product of two matrices is equal to the product of the determinants of the two matrices, or in other words $det(AB) = det(A)det(B)$?

Problem 3. Programming Problem: Density Estimation of Multivariate Gaussian [40 points]

If you have any questions or problems about the below, please post to our discussion board (https://canvas.uw.edu/courses/1431528/discussion_topics).

Before you start: please install anaconda python (python version 3.x (e.g., 3.6 or above) is strongly recommended) by following the instructions on https://www.anaconda.com/download/, and then install scikit-learn, numpy, matplotlib, seaborn, pandas and jupyter notebook in anaconda, for example, by running command "conda install seaborn". Note some of the above packages may have already been installed in anaconda, depending on which version of anaconda you just installed.

If you are not familiar with python and numpy, please check the following tutorials: http://cs231n.github.io/python-numpy-tutorial/ and https://docs.scipy.org/doc/numpy-1.15.0/user/quickstart.html

In this problem, you will estimate a multivariate Gaussian distribution from 2D points sampled from a multivariante Gaussian distribution, and learn how to use python and matplotlib to generate simple plots in ipython notebook. If you are not familiar with jupyter notebook, please check a quick tutorial on https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html.

We provide an ipython notebook "2d_gaussian.ipynb" for you to complete. In your terminal, please go to the directory where this ipynb file is located, and run command "jupyter notebook". A local webpage will be automatically opened in your web browser. Click the above file to open the notebook.

You need to complete everything below each of the "TODO"s that you find (please search for every "TODO"). Once you have done that, please submit the completed ipynb file as part of your included .zip file.

In your writeup, you also need to include the plots and answers to the questions required in this session. Please include the plots and answers in the pdf file in your solution.

Problem 3(a). (10 points) Run the first cell in the ipython notebook to generate 1000 points X (a 1000×2 matrix, each point has two coodinates) sampled from a multivariante Gaussian distribution. Estimate the mean and covariance of the sampled points and report them in your writeup.

Problem 3(b). (5 points) Plot the histogram for the x-coordinates of X and y-coordinates of X respectively. You can use the plt.hist() function from matplotlib.

Problem 3(c). (5 points) From the histogram, can you tell whether the x-coordinates of the 1000 points (the first column of X) follows some Gaussian distribution? If so, estimate the mean and variance. How about the y-coordinates? If so, estimate the mean and variance.

Problem 3(d). (10 points) Sample 1000 numbers from a 1D Gaussian distribution with the mean and variance of the x-coordinates you got from (3). Sample another 1000 numbers from a 1D Gaussian distribution with the mean and variance of the y-coordinates. You can use np.random.normal from numpy. Generate a new 2D scatter plot of 1000 points with the first 1000 numbers as x-coordinates and the second 1000 numbers as y-coordinates. Comparing to the original 1000 points, what is the difference about their distribution? What causes the difference?

Problem 3(e). (5 points) Plot a line segment with $x = [-10, 10]$ and $y = 3x + 1$ on the 2D-Gaussian plot. The np.linspace() function may be helpful. Project X onto line $y = 3x + 1$ and plot the projected points on the 2D space.

Problem 3(f). (5 points) Draw the histogram of the x-coordinates of the projected points. Are the x-coordinates of the projected points sampled from some Gaussian distribution? If so, estimate the mean and variance.