ALDA Fall 2021
HW 4
10/11/2021

HW4 contains 6 questions. Please read and follow the instructions.

- **DUE DATE FOR SUBMISSION: 10/29/2021 11:45 PM**
- **TOTAL NUMBER OF POINTS: 100 + 30 bonus points**

- **NO PARTIAL CREDIT** will be given so provide concise answers.
- **Submissions and updates should be handled by the same person.**
- **You MUST manually add ALL team members in the submission portal when you submit through Gradescope.**
- Make sure you clearly list **your homework team ID, all team members' names and Unity IDs, for those who have contributed to the homework contribution** at the top of your submission.
- **[GradeScope and NCSU Github]:** Submit a PDF on GradeScope. **You must submit your code in Github, and give the instructors access.**

  Follow these Github instructions: Please use the same homework repository you created for ALDA class and shared with us. The only thing you need to do for HW4 is to **create a new folder named HW4**. **All code MUST be in HW4 folder before the deadline. No credit will be given if code is not submitted for a programming question.** In your PDF submitted on GradeScope, reference the script/function for each question (e.g. "For solution to question 2, see matrix.py"). **Include your team's GitHub repository link in the PDF.**
- The materials on this course website are only for use of students enrolled in this course and **MUST NOT** be retained or disseminated to others.
- By uploading your submission, you agree that you have not violated any university policies related to the student code of conduct (https://policies.ncsu.edu/policy/pol-11-35-01/), and you are signing the Pack Pledge: **"I have neither given nor received unauthorized aid on this test or assignment"**.

---

1. (15 points) [**BN Inference**] [**John Wesley Hostetter (Designed) & Tyrone Wu (Graded)**]

   Compute the following probabilities according to the Bayesian net shown in Figure 1.

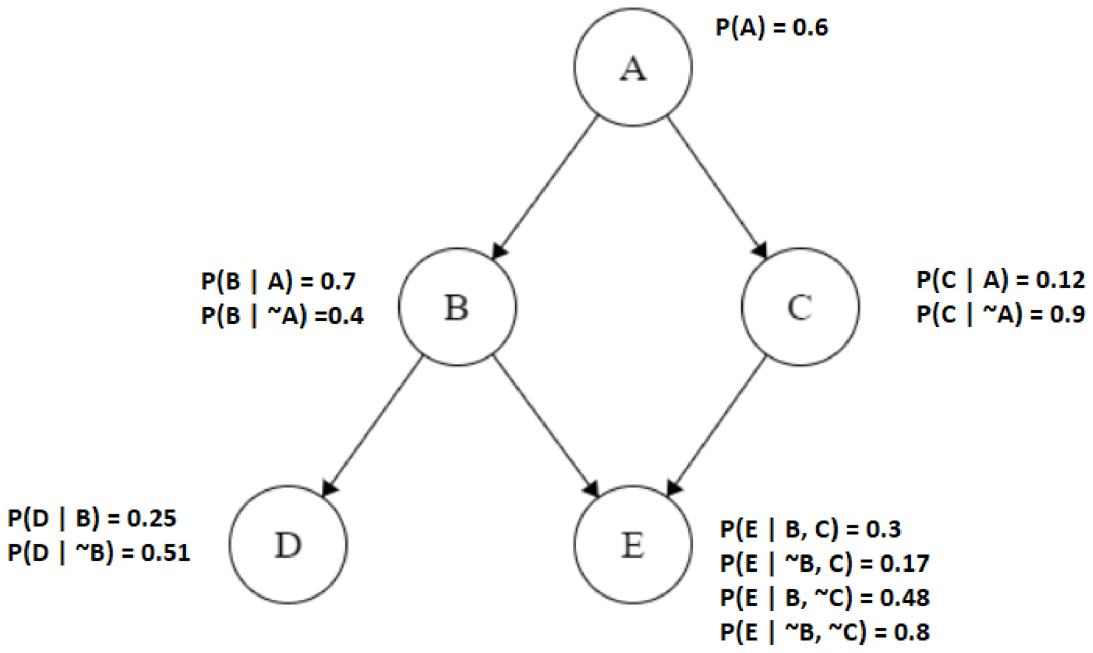

Figure 1: BN Inference

   (a) (4 points) Compute $P(A, \neg C, D, \neg E)$. Show your work.

   (b) (5 points) Compute $P(A \mid B)$. Show your work.

   (c) (6 points) Compute $P(\neg B \mid D)$. Show your work.

2. (20 points) [**D-Separation**] [**Designed and Graded by Ge Gao**] Conditional independence is a key concept in Bayesian Belief Network (BBN). Please answer the following conditional independence and d-separation questions using the graphs below.

   (a) (5 points) In Figure 2, is A d-separated by $\emptyset$ from B? If so, how? Justify your answer.

   (b) (5 points) In Figure 2, is $\{A, C\}$ d-separated by H from K? If so, how? Justify your answer.

   (c) (5 points) In Figure 2, is D d-separated by $\{B, J\}$ from E? If so, how? Justify your answer.

   (d) (5 points) In Figure 2, is $\{A, C, G\}$ d-separated by $\{H, K\}$ from $\{B, E, F\}$? If so, how? Justify your answer.

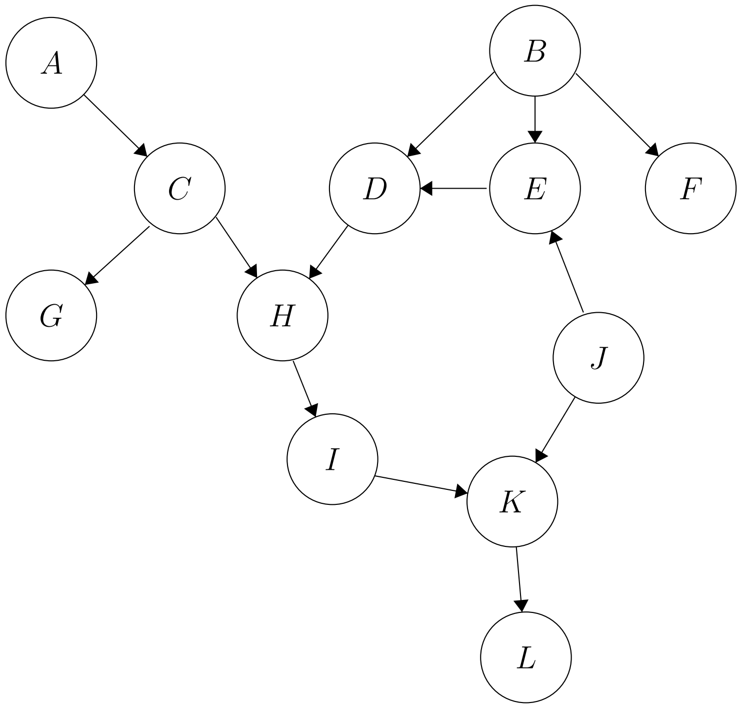

Figure 2: Bayesian Belief Network

3. (20 points) [**General Linear Regression**] [**Designed and Graded by Angela Zhang**]

   (a) (5 points) Given the following three data points of (x_1, x_2, y): $(0, 3, -3)$, $(1, 1, 4)$, $(3, 1, 5)$, try to use a linear regression $y = \beta_1 x_1^2 + \beta_2 x_1 x_2 + \beta_0$ to predict y. Determine the values of β_1, β_2 and β_0 by **manual calculation** and show each step of your work.

   (b) (15 points) [Programming Task] Apply the following three linear regressions:
   (1) $y = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_0$
   (2) $y = \beta_1 x_1^2 + \beta_2 x_2^2 + \beta_3 x_3^2 + \beta_0$
   (3) $y = \gamma_1 x_1^3 + \gamma_2 x_2^3 + \gamma_3 x_3^3 + \gamma_0$

   to the provided data file "generator_temperature.csv".

   The objective of the linear regression model is to predict the house price of unit area. Write code in Python to perform following tasks. Please *report your output and relevant code* in the document file and also include your code (ends with .py) in the .zip file.

   You are allowed to use `numpy`, `pandas`, `sklearn`'s LOOCV (Leave One Out Cross Validation), and any linear regression library to answer this question.

   i. (10 points) Load the data. Fit the whole dataset to the three linear regression models, respectively. *Report* the coefficients (α_s, β_s, γ_s) of the three models.

   ii. (5 points) Use **leave-one-out** cross validation to determine the RMSE (root mean square error) for the three models. Specifically, in each fold, fit the training data to the model to determine the coefficients, then apply the coefficients to get predicted label for testing data (You don't need to report the coefficients in each fold). *Report* RMSE for the three models. Based on the RMSE, which model is the best for fitting the given data?

4. (25 points) [**ANN**] [**Designed and Graded by John Wesley Hostetter**]

   Train, validate, and test a neural network model using the dataset in "ann_2021.zip", which contains training data, validation data, and test data. The goal here is to train the artificial neural network to recognize hand written digits. Use the Python neural networks package, Keras (i.e. `import tensorflow.keras`), for this problem. (All the output should be included in your report. Otherwise, your points are deducted.)

   (a) (3 points) Please briefly describe how to construct your working environments (e.g. language, package version, backend for neural networks, installation, etc.) in your report, and write how to execute your code in a 'README.md' file.

   (b) (2 points) Please apply a fixed random seed of 2021 in order to generate the same result every time. To do this, you will need to override the 'PYTHONHASHSEED' environment variable accessible via the `import os`. Then, write the following: `os.environ['PYTHONHASHSEED']='2021'`. You should also override the random seed generators found in the 'random', 'numpy', and 'tensorflow' libraries with the value of 2021.

   (c) (8 points) Create a body of code that iterates over a range of possible number of hidden neurons $X = (4, 16, 32, 64)$ that will be used to define a neural network. For each number of hidden neurons, $x \in X$, do the following:

      i. Define a neural network with its parameters as follows: activation function for hidden layer = 'relu', activation for output layer = 'sigmoid', loss function = 'binary_crossentropy', optimizer = 'adam', metrics = 'accuracy', epochs=5, batch_size=10. The model should have a single hidden layer, with the current number of hidden neurons in the loop, x, being assigned to define the size of the single hidden layer.

      ii. Fit the neural network with the provided training data (this is where you will need the epochs and batch_size parameters provided in the previous step).

      iii. Validate the neural network that has a hidden layer of size x using the given validation data. The validation accuracy is used to determine how many number of hidden neurons are optimal for this problem. You will want to save this value for later use.

      iv. Provide the essential code for the "neural network learning" and include descriptive comments in your report.

   (d) (5 points) Plot a figure, where the horizontal x-axis is the number of hidden neurons, and the vertical y-axis is the accuracy. Please plot both training and validation accuracy in your figure. (Note that the exact accuracy could be slightly different according to your working environments, however you can analyze the trend.)

   (e) (5 points) Provide a simple analysis about your results and choose the optimal number of hidden neuron from the analysis.

   (f) (2 points) Report the test accuracy using the given test data on the neural network with the optimal number of hidden neurons.

5. (20 points) [**SVM Programming**] [**Ge Gao (Designed) & Chengyuan Liu (Graded)**] In this question, you will employ SVM to solve a classification problem for the provided "svm_data_2021.csv". This data consists of 15 features and a label for each row that could be 0 or 1. Write code in Python to perform the following tasks. Please *report your output and relevant code* in the document file and also include your code (ends with .py) in the .zip file.

   (a) (1 point) Load data. Report the size of the positive (class 1) and negative (class 0) samples in dataset.

   (b) (2 points) Use *stratified random sampling* to divide the dataset into training data (80%) and testing data (20%). *Report* the number of positive and negative samples in both training and testing data.

   (c) (7 points) Using the data given in "svm_2021.zip" in that "train_data_2021.csv" as training data, "test_data_2021.csv" as testing data, take SVM with linear kernel as classifier (*third-party packages are allowed to be used*) and set the regularization parameter C as: [0.1, 0.2, 0.3, 0.5, 1, 2, 3, 5, 10], respectively. For each value of C, train a SVM classifier with the training data and get the number of support vectors (SVs). Generate a plot with C as the horizontal axis and number of SVs as the vertical axis. Give a brief analysis on the plot by explaining: 1) as C increases, how the number of SVs changes, and 2) why.

   (d) (10 points) Compare the performance of four different kernel functions: linear, polynomial, radial basic function (Gaussian kernel), and Sigmoid. For each type of kernel functions, train your SVM classifiers using the training data and evaluate the resulted SVM classifer using testing data by making a table to record *accuracy*, *precision*, *recall* and *f-measure* of the corresponding classification results.

   For different kernel functions, try to tune its parameters such that:

   - for the linear kernel, try to tune C.
   - for the polynomial kernel, try to tune C, degree, and coef0.
   - for the RBF kernel, try to tune C and γ.
   - for the Sigmoid kernel, try to tune C, coef0, and γ.

   More specifically, you should explore each of parameters within these ranges:
   $C \in [0.1, 0.2, 0.3, 1, 5, 10, 20, 100, 200, 1000]$
   degree $\in [1, 2, 3, 4, 5]$.
   coef0 $\in [0.0001, 0.001, 0.002, 0.01, 0.02, 0.1, 0.2, 0.3, 1, 2, 5, 10]$
   $\gamma \in [0.0001, 0.001, 0.002, 0.01, 0.02, 0.03, 0.1, 0.2, 1, 2, 3]$
   You are strongly encourage to use `sklearn`'s GridSearchCV() function; if you decide to use this library feature, please set the cross validation parameter to be 5 (argument of "None" is 5-fold cross validation by default). For each of the four types of kernel functions, please report the best result (using F-measure to break ties) and its corresponding optimal parameters. For this dataset, which kernel function will you choose?

6. **[30 Bonus Points] [SVM Theory] [Angela Zhang (Designed) & Tyrone Wu (Graded)]**

Given 3-dimensional data points $\langle \mathbf{X}^i, Y \rangle$, $i \in [1, 2, 3, 4]$ as shown in Table 1. In this question, you will employ the kernel function for SVM trained with these four data points. Let α_1; α_2; α_3, and α_4 be the Lagrangian multipliers associated with them as: α_i is associated with $\langle \mathbf{X^i}; y_i \rangle$.

Data ID	x_1	x_2	y
$\mathbf{X}^1$	-1	1	1
$\mathbf{X}^2$	-1	-1	1
$\mathbf{X}^3$	1	1	-1
$\mathbf{X}^4$	1	-1	-1

Table 1: Q5(b)

(a) (4 points) Suppose the kernel function is: $\mathbf{K}(\mathbf{X}^i, \mathbf{X}^j) = (2 * (\mathbf{X}^i \cdot \mathbf{X}^j) + 4 * (\mathbf{X}^i \cdot \mathbf{X}^j)^2)$, where $\mathbf{X}^i$ and $\mathbf{X}^j$ are two data points i and j. This kernel is equal to an inner product $\phi(\mathbf{X}^i) \cdot \phi(\mathbf{X}^j)$ with a certain function of ϕ. What is the function of ϕ?

(b) (3 points) Transform the four given data points $\mathbf{X}^i, i \in [1, 2, 3, 4]$ to the higher dimensional space via the function ϕ get from (i). Report your results.

(c) (10 points) Using the kernel function in (a), what *(dual) optimization problem* needs to be solved in terms of the α_is in order to determine their values?

(d) (10 points) Assume that the solution to the optimization problem shows that $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1/8$. Apply Lagrange multipliers to determine the *maximum margin linear decision boundary* in the transformed higher dimensional space. Note that the maximum margin linear decision boundary is defined as $w \cdot \phi(x) + b$, where the transformation ϕ is already defined by the kernel above and b is constant. (Show your work)

(e) (3 points) Use the maximum margin linear decision boundary from (d) to classify point (-2,-2).