Homework 3

CSC522: Automated Learning and Data Analysis
Dr. Thomas Price

Spring 2019

Instructions

- **Due Date:** Apr 2, 2019
- **Total Points**: 80
- Make sure you clearly list each team member's names and Unity IDs at the top of your submission.
- Your submission should be a single zip file containing a PDF of your answers, your code, and a README file with running instructions. Please follow the naming convention for your zip file: G(homework group number)_HW(homework number), e.g. G1_HW3.
- If any question is unclear, please post a question on Piazza. If you make any assumptions in your answer, you must state them explicitly (e.g. "Assuming the data in question is normally distributed...").
- If a question asks you to explain or justify your answer, **give a brief explanation** using your own ideas, not a reference to the textbook or an online source.
- If this homework needs to be updated for any reason, **we will post the update on Piazza** and announce it in class, so please stay alert.
- In addition to your group submission, please also *individually* submit your Peer Evaluation form on Moodle, evaluating yours and your teammates' contributions to this homework.
- While you will submit *one* homework as a team, you are responsible for *all* of the content on the homework. We recommend attempting to solve each question individually.

R Programming Submission Instructions

- Make sure you clearly list each team member's names and Unity IDs at the top of your submission.
- Your code should be named *hw*3*.R*. Add this file, along with a README to the zip file mentioned in the first page.
- Failure to follow naming conventions or programming related instructions specified below may result in your submission not being graded.
- Carefully read what the function names have been requested by the instructor. **In this homework or the following ones, if your code does not follow the naming format requested by the instructor, you will not receive credit.**
- For each function, both the input and output formats are provided in the *hw*3*.R*. Function calls are specified in *hw*3*_checker.R*. Please ensure that you follow the correct input and output formats. Once again, if you do not follow the format requested, you will not receive credit. It is clearly stated which functions need to be implemented by you in the comments in hw3.R
- You are free to write your own functions to handle sub-tasks, but the TA will only call the functions he has requested. If the requested functions do not run/return the correct values/do not finish running in specified time, you will not receive full credit.

- DO NOT set working directory (`setwd` function) or clear memory (rm(list=ls(all=T))) in your *hw*3*.R* code. TA(s) will do so in their own auto grader.
- The TA will have an autograder which will first run `source(hw3.R)`, then call each of the functions requested in the homework and compare with the correct solution.
- Your code should be clearly documented.
- To **test you code**, step through the `hw3_checker.R` file. If you update you code, make sure to run `source('./hw3.R')` again to update your function definitions. You can also check the "Source on save" option in R Studio to do this automatically on save.
- Please DO NOT include `install.packages()` or `install_github()` in your `hw3.R`. Comment them out. Please note, we are specifying the allowed packages, which means that we already have them installed on our test machine. Having uncommented `install.packages` or `install_github` in your code would result in a penalty of 5 points.

Problems

1. BN Inference (12 points) [**Song Ju**]. Compute the following probabilities according to the Bayesian net shown in Figure 1. (∼ means not)

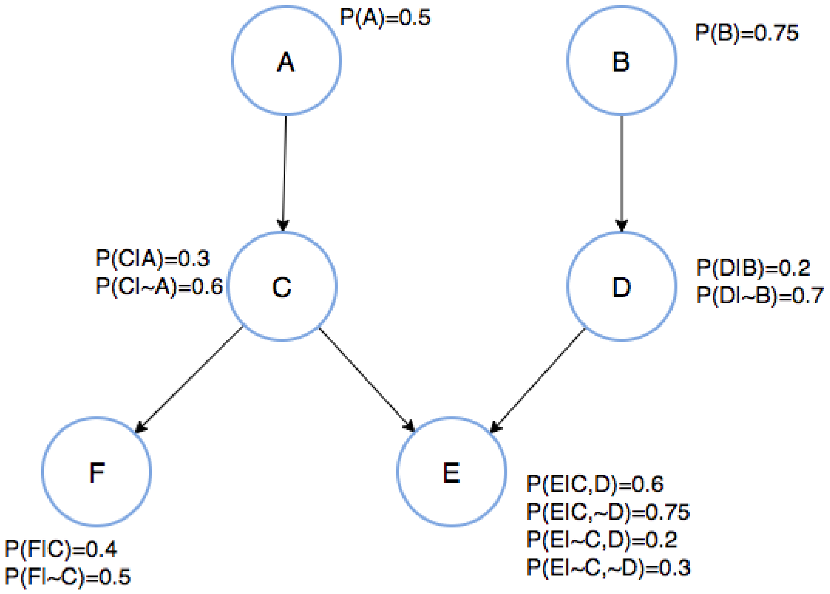

Figure 1: BN Inference

   (a) Compute $P(D, B|A)$. Show your work.
   (b) Compute $P(C)$. Show your work.
   (c) Compute $P(F)$. Show your work.
   (d) Compute $P(B, \sim C, D, E, F)$. Show your work.

2. SVM Theory (20 points) [**Song Ju**].

   (a) Support vector machines (SVM) learn a decision boundary leading to the largest margin between classes. In this question, you will train a SVM on a tiny dataset with 4 data points, shown in Figure 2. This dataset consists of two points with Class 1 (y = 1) and two points with Class 2 (y = -1). Each data point has two non-class attributes: $X1$ and $X2$.

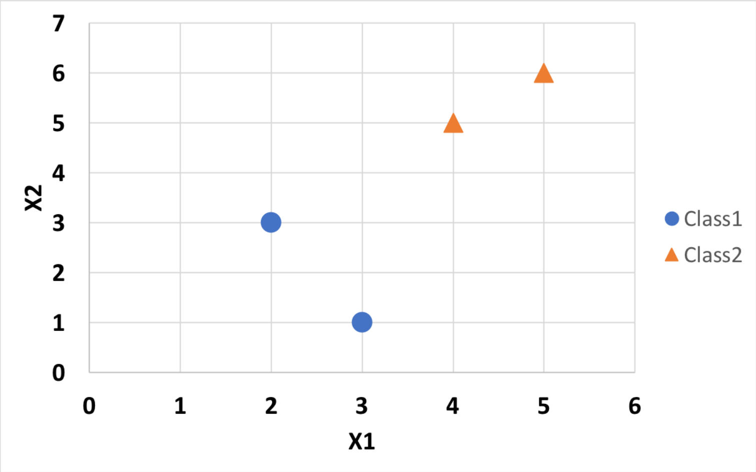

Figure 2: SVM

i. Assume that $w_1 = w_2$. Find the weight vector $\mathbf{w}$ and bias b for the decision boundary of the SVM. What is the equation corresponding to this decision boundary?

ii. Circle the support vectors and draw the decision boundary.

(b) Given 2-dimensional data points $X^i, i \in [1, 2, 3, 4]$ as shown in Table 1, in this question, you will employ the kernel function for SVM to classify these four data points.

Data ID	x_1	x_2	y
X^1	0	1	-1
X^2	0	-1	-1
X^3	1	0	1
X^4	-1	0	1

Table 1: Four Data Points

i. Suppose the kernel function is: $K(X, Z) = (1 + 2 \cdot X \cdot Z)^2$, where X and Z indicate two data points. This kernel is equal to an inner product $\phi(X) \cdot \phi(Z)$ with a certain function, ϕ. Calculate the function ϕ.

ii. Transform the four given data points $X^i, i \in [1, 2, 3, 4]$ to the higher dimensional space using the function ϕ that you derived in part (i). Report $\phi(X^i)$ for $i \in [1, 2, 3, 4]$.

iii. Assume that the four transformed data points that you got from part (ii) are all support vectors. Apply Lagrange multipliers to determine the *maximum margin linear decision boundary* in the transformed higher dimensional space. **Note**: this will involve solving a system of equations.

3. Linear Regression (15 points) [**Ruth Okoilu**].

(a) Given the following three training data points of the form (x, y): $(2, 5)$, $(0, -2)$, $(3, -3)$, estimate the parameters for linear regression of the form $y = w_1x^2 + w_0$.
Note that x is squared in the formula.

i. Determine the values of w_1 and w_0 and show each step of your work.

ii. Calculate the training RMSE for the fitted linear regression.

4. Programming (33 points) [**Krishna Gadiraju**] In this question, you will be performing a variety of machine learning operations - regression and classification.

(a) **PART-1: Regression** (15 points)

i. **Dataset description:** You are provided a dataset with 20 variables. Variables $x1 - x19$ refer to the independent variables, while variable y is your dependent variable. Training data is stored in the file `data/regression-train.csv`, and test data is stored in the file `data/regression-test.csv`.

ii. **Note:** The TA will use a different version of `data/regression-test.csv`. The format (independent variables $x1 - x19$, dependent variable y) will be similar, but TA's file may contain different number of data points than the file supplied to you. Please ensure you take this into account, and do not hard code any dimensions.

iii. In this exercise, you will apply three different types of regression methods to the dataset supplied to you, and then compare their results:

A. **Learning:** You will write code in the function `alda_regression()` to train simple linear regression, ridge regression and lasso regression models. Detailed instructions for implementation and allowed packages have been provided in `hw3.R`. Note that for the lasso and ridge regression models, you will be using crossvalidation to tune the λ hyperparameter.

B. **Comparison:** You will write code in the function `regression_compare_rmse()` to compare the three regression models from above. Detailed instructions for implementation and allowed packages have been provided in `hw3.R`

(b) **PART-2: Classification** (18 Points)

i. **Dataset description:** You are provided a dataset with 5 variables. Variables $x1 - x4$ refer to the independent variables, while variable *class* is your class variable. Training data is stored in the file `data/classification-train.csv`, and test data is stored in the file `data/classification-test.csv`.

ii. **Note:** The TA will use a different version of `data/classification-test.csv`. The format (independent variables $x1 - x4$, dependent variable *class*) will be similar, but TA's file may contain different number of data points than the file supplied to you. Please ensure you take this into account, and do not hard code any dimensions.

iii. In this exercise, you will apply two different types of classification methods to the dataset supplied to you, and then compare their results:

A. **Support Vector Machine:** In this exercise, you will use cross validation to tune hyperparameters for four different types of kernels : linear, radial basis, polynomial and sigmoid kernels. You will write code in the function `alda_svm()`. Detailed instructions for implementation and allowed packages have been provided in `hw3.R`

B. **Comparison:** You will write code in `classification_compare_accuracy()` to compare all 4 SVM kernels. Detailed instructions for implementation and allowed packages have been provided in `hw3.R`.

NOTE: Your entire solution `hw3.R` should not take more than 3 minutes to run. Any solution taking longer will be awarded a zero.