CS57800 Statistical Machine Learning

Homework 1

Due: Sep 15, 2015 on Tuesday

1 Foundations

1. Consider the planes $x_1+x_2+3x_3 = 4$ and $x_1+2x_2+4x_3 = 5$ in $\mathbb{R}^3$. Find parametric equations for the line of intersection of these two planes.

2. Given three points $P(0,0,0)$, $Q(1,-1,1)$, $R(4,3,7)$, find a vector which is orthogonal to the plane through P, Q, and R.

3. Differentiate the following equations.
   (a) $f(x) = (3x^2)(x^{\frac{1}{2}})$
   (b) $f(x) = (e^{2x} + e)^{\frac{1}{2}}$
   (c) $f(x) = [\ln(5x^2 + 9)]^3$

4. Find $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$.
   (a) $f(x,y) = xy^3 + x^2y^2$
   (b) $f(x,y) = xe^{2x+3y}$

5. We say that $f(n) \prec g(n)$ if $g(n)$ grows faster than $f(n)$. Order the following functions by $\prec$ from the the lowest to the highest:
   $(\frac{5}{3})^{2n}$, 10^8, $\sqrt{n^3}\log^2 n$, $2^{\log_2 n}$, $\log^4\sqrt{n}$, $2^{3\log_2 n}$, 2^n

6. Suppose you roll three dice. Compute the followings: (a) the expected value of the sum of the rolls, (b) the expected value of the product of the rolls, and (c) the variance of the sum of the rolls.

2 Programing: Decision Tree

General Programing Requirements

You should implement your solution using Python or JAVA. You may **not** use anybody else's code; you **must** implement your own version! We will use **MOSS** (Measure Of Software Similarity) from Stanford University to check for plagiarism. You should submit your source code files along

with your typed HW report. Any external packages are **not** allowed (e.g. scikit-learn for python), but feel free to use any internal packages (e.g. sys, os, math in python). The TAs should be able to compile and run your code. (And also keep in mind the JAVA version on lore.cs.purdue.edu is 1.5 while python is 2.6)

Task Description

In this section, you will develop a decision tree classifier for medical data. You will be working with a database of instances describing patients who have been tested for breast cancer. You will develop a decision tree classifier that can determine whether growths are malignant or benign, based on the results of tests taken by a patient. The data set was derived from the Wisconsin breast cancer corpus.

You can find the data set for this assignment, as well as code template (just for python) , on Piazza, under the tab "resources" within the section "Homework". Each sample in the data consists of 9 attributes, each of which ranges from 1 to 10, as well as a binary label that is 0 or 1. The file breast-cancer-wisconsin.names describes the attributes and also contains information about the history of the data set. Also a template for your coding is provided, named as "example.py".

You are required to complete the missing part of the provided template (or write your own version). Please read the materials provided by the instructor to understand the details of decision tree before you start to implement it.

The data set has been separated into three pieces, namely training (60 samples), validating (20 samples) and testing (20 samples), and we added some noisy to the data set during separation. Use the training part to build your decision tree and use the validating data set to tune your hyper-parameter(s). And finally use testing data set to test your final model.

You are required to write a report as analysis of your experiments as well.

(Hint: For a decision tree, using the Object Oriented Characteristics of the programing language can make your life much more easier)

Submission Instructions:

You are required to use LaTeX to type your solutions to questions, and report of your programing as well. Other formats of submission will **not** be accepted. A template named "homework.tex" is also provided for your convenience.

After logging into lore.cs.purdue.edu (physically go to the lab or use ssh remotely, you are all granted the accounts to CS lore machines during this class), please follow these steps to submit your assignment:

1. Make a directory named *'your Name_your Surname'* and copy all of your files there.
2. While in the upper level directory (if the files are in /homes/dan/dan_goldwasser, go to/homes/dan), execute the following command:

`turnin -c cs578 -p HW1 *your_folder_name*`

(e.g. your prof would use: `turnin -c cs578 -p HW1 dan_goldwasser` to submit his work)

Keep in mind that old submissions are overwritten with new ones whenever you execute this command.

3. You can verify the contents of your submission by executing the following command:

   `turnin -v -c cs578 -p HW1`

   Do **not** forget the -v flag here, as otherwise your submission would be replaced with an empty one.

Your submission should include the following files:

1. The source code in python or java (and the executable .jar file for java).
2. Your evaluation & analysis in .pdf format (together with your HW solutions in it).
3. A README file containing your name, instructions to run your code and anything if you would like us to know about your program (like errors, special conditions, etc).