CS 412 — Introduction to Machine Learning — Fall 2017

Assignment 1

Department of Computer Science, University of Illinois at Chicago

Instructor: Xinhua Zhang

Due: Oct 2, 2017 by 5 PM (CST)
Out: Sept 17, 2017

Instructions

This is an **individual** assignment**.** Your mark will be out of 100, and it contributes 8% to your final course score.

What to submit: You should submit to **Blackboard** a single zipped file (.zip or .tar) which includes the following files:

1. A single Portable Document Format (PDF) file, with file name Surname_UIN.pdf, where Surname is your last name and UIN is your UIC UIN. It can be either typed using WORD or latex, or scanned copies of handwritten submissions provided that the handwriting is neat and legible.

2. For the programming questions, please submit:

a) **Code**: All the code you wrote – in a form that the TA can run it. Please include plenty of comments. Put all code in one folder named Code/.
b) **ReadMe**: A ReadMe file that explains to your TA how to run your code. Name the file as ReadMe under the Code/ folder.

How to submit: You'll submit your homework online, via blackboard. Go to the class web site, and folder Assignments. The entry "Assignment 1" in this folder will have a link through which you can upload your homework. **Submit a single zipped file named**

Surname_UIN.zip or Surname_UIN.tar

Inside it, there should be a PDF file and a Code folder as mentioned above.

Resubmission: The submission site will be open up to **5 PM of Oct 2, 2017**. You are allowed to resubmit (upload a new version) until the deadline. Grading will be conducted on the **last** version submitted **before** the deadline.

Programming Languages:

The grading will be based on the output of executing your code on some test examples, along with the performance of algorithms that you plot in the assignment submission. Clarify and readability of your code will also count. So you may choose to use any programming language, subject to the following notes:

1. We will provide some utility functions, and they will only be in Matlab/Octave and Julia.

2. We will provide support for Matlab/Octave, Julia, and Python, but not any other language.

3. We will be happy to run Matlab, Octave, Julia, and Python. So if you use any other language, you might be asked to demonstrate running it with the TA if we think it appropriate.

Octave versus Matlab: If you have Matlab and you prefer using it, then we recommend using it. There won't be any disadvantage if you use Matlab (in fact it is more advantageous because Matlab is more efficient and stable than Octave).

Late Policy: Each student is allowed up to three (3) late days total for all homework assignments (not 3 late days for each homework assignment) with no penalty. Beyond these "free" late days, late submissions of homework will be penalized up to 50% if received within three (3) days of the deadline. No credit will be awarded for homework submitted beyond three (3) days after the deadline without prior permission for extenuating circumstances.

Cheating and Plagiarism: All assignments must be done individually. Remember that plagiarism is a university offence and will be dealt with according to university procedures. Please refer to the corresponding UIC policies: http://dos.uic.edu/docs/Student%20Disciplinary%20Policy.pdf

Latex primer: http://ctan.mackichan.com/info/lshort/english/lshort.pdf

CS 412: Introduction to Machine Learning
Fall 2017
Homework 1
Due: October 2, 5 PM

Problem 0. Your Background and Interests (5 points)

(a) What do you hope to gain by taking this course? **(2 points)**

(b) Is there a certain type of data that you are mainly interested in? **(1 point)**

(c) What is your past experience in probability and statistics? **(1 point)**

(d) What is your past experience using scientific programming language (e.g.: MATLAB, Python, R, Octave, Julia) or other programming languages? **(1 point)**

Problem 1. Nearest Neighbors: Training Set Size and Noise (40 points)

Please note before submitting your code : Since the experiments involve randomness, it is important to ensure that your results are replicable. To this end, your implementation should take one integer (or any numeric value) as a seed that is used to initialize the random number generators. See, e.g.
`https://www.mathworks.com/help/matlab/ref/rng.html`
`https://stackoverflow.com/questions/25006370/set-the-random-seed-in-julia-generator-of-random-numbers`

Getting Started

(i) **Octave**
After installing Octave to your system, download the supplement files for this homework from Piazza (`mnistData.mat, iris.csv, mnist1NNdemo.m, sqDistance.m, iris1NNboundary.m`). Make sure you change the Octave working directory (via e.g. `cd /home/bziebart/code/`) to the diretory where you put the files. From the Octave console run `pkg install -forge statistics` to install statistics package (note: some system already has it pre-installed).

(ii) **Julia**
After installing Julia, you may want to install Juno IDE (`http://junolab.org/`) to help code Julia. Download the supplement files for this homework from Piazza (`mnistData.mat, iris.csv, setup_package.jl, mnist1NNdemo.jl, iris1NNboundary.jl`). Make sure you change the Julia working directory (via e.g. `cd("/home/vganap2/code/")`) to the directory where you put the files. From Julia/Juno's console, run `include("setup_package.jl")` to install and compile all required packages.

Problem Statement:

In this problem, you will use Octave or Julia to evaluate a nearest neighbor classifier in two machine learning tasks. Octave can be obtained from `https://www.gnu.org/software/octave/download.html` and Julia can be downloaded from `http://julialang.org/downloads/`. Both Octave and Julia are free alternatives to MATLAB for scientific computing, with syntax similar to MATLAB. While Octave maintain compatibility with MATLAB, Julia takes different approach by changing the syntax to enable JIT compilation. The guideline for the syntax differences can be found in `http://docs.julialang.org/en/release-0.4/manual/noteworthy-differences/`.

(a) Optical character recognition (OCR)

How does the the classification error of the 1-Nearest Neightbor for the MNIST dataset change with the number of training examples? Plot the n-fold cross validation error for 1000 training examples where $n \in \{3, 10, 50, 100, 1000\}$. Please submit a print out of your plot and relevant code snippet. **(20 points)**

Note: Code example for creating line chart and image plot in Octave and Julia are given in the last part of `mnist1NNdemo.m` and `mnist1NNdemo.jl`

(b) Iris plant recognition

Within your working directory, run `iris1NNboundary` in Octave or `include("iris1NNboundary.jl")` in Julia to produce a plot of the decision boundary of 1-Nearest Neighbors classifier. You will be making use of and modifying this source code for this problem.

The iris data set has 150 examples and 3 classes. Now randomly choose m examples, $m \in \{10, 20, 30, 50\}$ and flip the class label of each of the examples. For instance if the true class label of example i is 1 then replace the class label with either 2 or 3. Now you will have 4 modified datasets. For each of the modified datasets, run the k-NN algorithm with $k = 3$ and then plot the decision boundary. Also report the training error.

Please submit a print out of your results end relevant code snippet.

Hints: (i) The help system can be accessed using: `help` ***`keyword`*** in Octave or `?`***`keyword`*** in Julia. (ii) Introduction to Octave: `http://math.jacobs-university.de/oliver/teaching/iub/resources/octave/octave-intro/octave-intro.html`. (iii) Learn Julia in Y Minutes: `https://learnxinyminutes.com/docs/julia/`.

Problem 2. Nearest Neighbors (15 points)

Consider the following dataset with six examples:

x_1	x_2	x_3	x_4	y
3	10	2	11	Red
17	-17	9	-1	Blue
-4	9	-2	-1	Red
4	0	2	-5	Blue
8	-1	6	-12	Blue
19	3	23	14	Red

(a) For a new testing example, $x_1 = 0.0$, $x_2 = 0.0$, $x_3 = 0.0$, $x_4 = 0.0$, write the distance to each of the training examples and indicate the prediction made by 1-NN and 3-NN using Euclidean distance. **(5 points)**

(b) For a new testing example, $x_1 = 0.0$, $x_2 = 0.0$, $x_3 = 0.0$, $x_4 = 0.0$, write the distance to each of the training examples and indicate the prediction made by 1-NN and 3-NN using Manhattan distance. **(5 points)**

(c) What is the Leave-One-Out-Cross-Validation (LOOCV) error rate of 1-NN using Manhattan distance on this dataset? Indicate which examples (if any) contribute to the error rate. **(5 points)**

Problem 3. Decision Trees (20 points)

Consider a binary class classification problem with following data pionts:

x_1	x_2	x_3	y
1	1	25	-
-1	-1	10	-
-1	1	7	-
1	-1	12	-
-1	-1	9	+
-1	-1	6	-
1	-1	22	-
-1	1	8	+
-1	-1	7	-
-1	1	9	+

In this problem we are going to build three (binary) decision trees based on three different impurity measures.

(a) Use the entropy impurity to create by hand a decision tree classifier for this data.

(b) Use the error rate to create a decision tree classifier for this data.

(c) Find the optimal (smallest height) decision tree for this data which perfectly classify all data points.

(d) Write your conclusion from previous parts.

Problem 4. Maximum Likelihood Estimation (20 points)

Given a dataset $\{x_1, x_2, ..., x_N\}$ of size N, derive the maximum likelihood estimate (as a function of $x_1, ..., x_N$) for:

(a) The lower and upper limits, a and b, of a uniform distribution,

$$f(x; a, b) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

(assuming each $x_i \in \mathbb{R}$). Show all of your work. **(10 points)**

(b) The λ parameter of a Poisson distribution,

$$f(x; \lambda) = \begin{cases} e^{-\lambda} \frac{\lambda^x}{x!}, & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2}$$

(assuming each $x_i \geq 0$). Show all of your work. **(10 points)**

Hints: (i) plotting some sample data may be helpful and calculus should not be required (a); (ii) maximizing the *log likelihood* provides the same parameter values and often provides a simpler path to a solution (b); (iii) $\log(ab) = \log a + \log b$; (iv) $\log e^a = a$.