CS 542: Machine Learning **Spring 2018**

Problem Set 2

Lecturer: Prof. Peter Chin *Due: Feb 22, 2018*

◇ Please email your written problems (must be typed), code and report to `kenzhou@bu.edu` by 23:59PM on the due date.

◇ Late policy: there will be a penalty of 10% per day, up to three days late. After that no credit will be given.

1. **(60 points) Written Problems**

(a) (15 points) Bishop 3.3

(b) (15 points) Bishop 3.11

(c) (15 points) Bishop 3.14

(d) (15 points) Bishop 3.21

2. **(120 points) Programming**

(a) (60 points) Linear Regression

We are given data used in a study of the homicide rate (`HOM`) in Detroit, over the years 1961-1973. The following data were collected by J.C. Fisher, and used in his paper "Homicide in Detroit: The Role of Firearms," Criminology, vol. 14, pp. 387-400, 1976. Each row is for a year, and each column are values of a variable.

FTP	UEMP	MAN	LIC	GR	NMAN	GOV	HE	WE	HOM
260.35	11.0	455.5	178.15	215.98	538.1	133.9	2.98	117.18	8.60
269.80	7.0	480.2	156.41	180.48	547.6	137.6	3.09	134.02	8.90
272.04	5.2	506.1	198.02	209.57	562.8	143.6	3.23	141.68	8.52
272.96	4.3	535.8	222.10	231.67	591.0	150.3	3.33	147.98	8.89
272.51	3.5	576.0	301.92	297.65	626.1	164.3	3.46	159.85	13.07
261.34	3.2	601.7	391.22	367.62	659.8	179.5	3.60	157.19	14.57
268.89	4.1	577.3	665.56	616.54	686.2	187.5	3.73	155.29	21.36
295.99	3.9	596.9	1131.21	1029.75	699.6	195.4	2.91	131.75	28.03
319.87	3.6	613.5	837.60	786.23	729.9	210.3	4.25	178.74	31.49
341.43	7.1	569.3	794.90	713.77	757.8	223.8	4.47	178.30	37.39
356.59	8.4	548.8	817.74	750.43	755.3	227.7	5.04	209.54	46.26
376.69	7.7	563.4	583.17	1027.38	787.0	230.9	5.47	240.05	47.24
390.19	6.3	609.3	709.59	666.50	819.8	230.2	5.76	258.05	52.33

It turns out that three of the variables together are good predictors of the homicide rate: `FTP`, `WE`, and one more variable.

Use methods described in Chapter 3 of the textbook to devise a mathematical formulation to determine the third variable. Implement your formulation and then conduct experiments to determine the third variable. In your report, be sure to provide the step-by-step mathematical formulation (citing Chapter 3 as needed) that corresponds to the implementation you turn in. Also give plots and a rigorous argument to justify the scheme you use and your conclusions.

Note: the file `detroit.mat` containing the data is given in the course website. To load the data into matlab workspace, use `load()` command. Least-squares linear regression in Matlab can be done with the help of the backslash (\) command.

(b) **[Nearest Neighbor – 60 points]**

For this problem, you will be implementing the k-Nearest Neighbor (k-NN) classifier and evaluating on a the `Lenses` and `Credit Approval` (CA) dataset the latter of which describes credit worthiness data (e.g., a binary classification).[1] I have split the available data into a training set `crx.data.training` and a testing set `crx.data.testing`.

The first step to working with the CA dataset is to process the data. In looking at the data description `crx.names`, a items of note is that there are some missing values, there exists both numerical and categorical features, and that it is a relatively balanced dataset (meaning a roughly equal number of positive and negative examples – not that you should particularly care in this case, but something you should look for in general). In typing the unix command

```
$ cut -f1 -d, crx.data.training | sort | uniq -c
  10 ?
 167 a
 375 b
```

we observe that there for the first feature, there are 10 instances with a missing value, 167 instances with the value *a*, and 375 instances with the value *b*. Secondly, consider the command

```
$ grep '+$' crx.data.training | cut -f1 -d, | sort | uniq -c
   3 ?
  81 a
 168 b
```

In this case, we are only considering the characteristics of feature 1 for the examples that are labeled positive.[2] While there are more sophisticated (and better) methods for imputing missing values, for this assignment, we will just use mean/median imputation. This means that for feature 1, you should replace all of the question marks with a *b* as this is the median value (regardless if you condition on the label or not). For real-valued

[1] `http://archive.ics.uci.edu/ml/datasets/Credit+Approval`

[2] Note that you will also have to do this with the testing data.

features, just replace missing values with the label-conditioned mean (i.e., $\mu(x_1|+)$ for instances labeled as positive).

The second aspect one should consider is normalizing features. Nominal features can be left in their given form where we define the distance to be a constant value (e.g., 1) if they are different values, and 0 if they are the same. However, it is often wise to normalize real-valued features. For the purpose of this assignment, we will use z-scaling, where

$$z_i^{(m)} \leftarrow \frac{x_i^{(m)} - \mu_i}{\sigma_i} \tag{2.1}$$

such that $z_i^{(m)}$ indicates feature i for instance m (similarly $x_i^{(m)}$ is the raw input), μ_i is the average value of feature i over all instances, and σ_i is the corresponding standard deviation over all instances.

i. Write down (on the written part of your solution) exactly how you imputed missing values for each feature (i.e., replaced all missing values of feature 1 with `b`) for the CA dataset. Note that you are free to impute the values using statistics over the entire dataset (training and testing combined) or just training, but please state your method. Create a program that can be run on the BU computing environment with the command
   `./process crx.data.training crx.data.testing`
   which will produce two files `crx.training.processed` and `crx.testing.processed`.

ii. Write a k-NN algorithm with L2 distance, $\mathcal{D}_{L2}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_i (a_i - b_i)^2}$, program that can be run with the commands
   `./run k lenses.training lenses.testing`
   `./run k crx.training.processed crx.testing.processed`
   respectively where k is an integer indicating the k in your k-NN algorithm. The output of your program should be the testing file with an additional comma-separated field indicating the prediction of your classifier. For example, if you labeled instance 3 of the lenses data as `2`, the output would be
   `3,1,1,2,1,3,2`
   Note that we are defining $\mathcal{D}_{L2}$ to have a component-wise value of one for categorical attribute-values that disagree and 0 if they do agree (as previously implied).

iii. Generate a table that reports the accuracy on both data sets for at least two different values of k in each case. I have included a small script that you can use to calculate the accuracy and check if your code works (the numbers are made up just for instructional purposes).

```
$ ./run 1 lenses.training lenses.testing | perl accuracy.pl
3 / 6 = 0.5
```

iv. The code you submit must be your own. If you find/use information about specific algorithms from the Web, etc., be sure to cite the source(s) clearly in your source-code. You are not allowed to submit code downloaded from the internet (obviously).