University of Pittsburgh
CS1675 Introduction to Machine Learning
Professor Milos Hauskrecht

Handout 1
January 28, 2021

Problem assignment 1

Due: Thursday, February 4, 2021

Problem 1. Matrix operations

Let us assume:

$$v = \begin{bmatrix} 9 & 5 & 10 \end{bmatrix}$$

$$u = \begin{bmatrix} 4 \\ 1 \\ 3 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$$

$$B = \begin{bmatrix} 7 & 1 & 9 \\ 2 & 2 & 3 \\ 4 & 8 & 6 \end{bmatrix}$$

$$C = \begin{bmatrix} 8 & 6 & 5 \\ 1 & -3 & 4 \\ -1 & -2 & 4 \end{bmatrix}$$

Please calculate (you may use Matlab):

- $u^T * u$
- $u * u^T$
- $v * u$

- $u + 5$
- A^T
- $B * u$
- B^{-1}
- $B + C$
- $B - C$
- $A * B$
- $B * C$
- $B * A$

Report the results.

Problem 2. Exploratory data analysis

In this problem we will explore and analyze the dataset *pima.txt* (please see the Canvas link). To do the analysis you will need to write short Matlab programs to generate the answers. Keep the code you write for future problem sets. Also you will be asked to submit some of it with this assignment.

The *pima.txt* is described in the file *pima_desc.txt* The dataset consists of 8 attributes and a binary attribute defining the class label, the presence of diabetes. Data entries are organized in rows such that attributes come first and the class label is last. Answer the following questions with the help of Matlab:

- (a) What is the range (minimum and maximum value) for each of the attributes? Hint: use Matlab's functions *min* and *max*.
- (b) What is the mean and standard deviation of each attribute? Hint: use Matlab's functions *mean* and *std*.
- (c) Split *pima.txt* data into two data subsets - one that includes only examples with class label "0", the other one with class "1" values. Hint: use Matlab's function *find* to split the data. Calculate and report the mean and standard deviations of each attribute (columns 1-8) on these two subsets. Analyze the means and standard deviations of attribute values and select the attribute you think should be most helpful in discriminating the two classes. Include the attribute name in the report and explain why you think the attribute is the best for discriminating the two classes.

- (d) Calculate and report correlations between the first 8 attributes (in columns 1-8) and the target class attribute (column 9). Use Matlab's *corrcoef* function to do the calculations. What is the attribute with the highest (positive) correlation to the target attribute? Do you think it is the most or the least helpful attribute in predicting the target class? Explain.

- (e) Calculate all correlations between 8 attributes (using the corrcoef function). Which two attributes have the largest mutual correlation in the dataset?

While the analysis using basic statistics as performed above conveys a lot of information about the data and lets us make some conclusions about the importance of attributes for prediction or their mutual relation, it is often very useful to inspect the data also visually and get more insight into various shapes and patterns they hide. In the following we will inspect the data using histograms and 2D scatter plots.

- (f) **Histogram analysis** gives us more information about the distribution of attribute values. Write (and submit) a Matlab function *histogram_analysis* that takes the data for an attribute (as a vector) and plots a histogram with 20 bins using Matlab's hist function.

- g Analyze attributes in the data using the new function. Answer the following questions: Which histogram resembles most the normal distribution? In your report show at least two histograms, including the choice you picked as the most normally distributed attribute.

- (h) Histogram analysis function you wrote in part (f) lets you plot the distribution of values for any input data. So we can use it to look at attribute distributions for class 1 and class 0 individually and compare them. Similarly to part (c) divide pima dataset into two datasets, one with instances corresponding to class 0 and the other one corresponding to class 1. For each attribute in columns 1 and 8 plot two histograms of the attribute values, one for class 1 and the other one for class 0. Compare the two histograms for each attribute. Based on the pairs of histograms choose an attribute you think should be most helpful in discriminating the two classes. Include the attribute name and the histograms for that attribute for class 1 and class 0 in the report. Explain why you think the attribute is the best.

- (i) **2D Scatter plots** plots let us inspect the relations between pairs of attributes. Write (and submit) a function *scatter_plot* that takes pairs of values for two attributes and plots them as points in 2D (use Matlab function scatter to do the plot). Analyze the pairwise relations between 8 attributes in the pima dataset using the scatter plot function. Answer the following questions. What scatter plot would you expect to see for the two dimensional space if the two attributes are independent and random? Do you see any interesting non-random patterns among the pairs? Include two scatter

graphs you think show some interesting dependences or patters. Explain why you think these are interesting? Do not forget to include with every plot the corresponding attribute names.

Problem 3. Data preprocessing

Before applying learning algorithms some data preprocessing may be necessary. In this problem we explore three preprocessing methods: transformation of categorical values to (safe) numerical representation, normalization of continuous values, and discretization of continuous values.

- (a) Assume you have an attribute with 8 categorical values {brown, blue, white, red, yellow, orange, green, black}. Devise one-hot encoding of the values and explain in the report how values are mapped. Use the mappings to convert the following vector of attribute values to one hot representation and include the results in the report:

$$\begin{bmatrix} red \\ black \\ yellow \\ red \\ green \\ blue \\ blue \end{bmatrix}$$

- (b) Write (and submit) function *normalize* that takes an unnormalized vector of attribute values and returns the vector of values normalized according to the data mean and standard deviation. Briefly, to calculate the normalized value we apply following formula:

$$x_{\text{norm}} = \frac{x - \mu_x}{\sigma_x}.$$

where x is an unnormalized value, μ_x is the mean value of the attribute in the data and σ_x its standard deviation. (Please note this is a different normalization compared to the one illustrated in the lecture !) Test your function on attribute 3 of the pima dataset. Report the mean, and the standard deviation for the attribute, and normalized values of the attribute 3 for the first five entries (rows) in the dataset.

- (c) Write (and submit) a function *discretize_attribute* that takes a vector of attribute values, and number k (number of bins) as inputs and assigns each value to one of the k bins. Bins should be of equal length (cover intervals of the same length) and should cover the full range of values that are determined by the min and the max operations on the vector. Every bin is given a numerical label such that the smallest value is in bin 1 and the largest attribute value is in bin k. The bin label represents the result of

discretization. Test your function on attribute 3 of the pima dataset. Assume we use 10 bins. Report new (discretized) values of the attribute 3 for the first five entries in the dataset.

Problem 4. Splitting data into training and testing sets

In this problem we write functions supporting the splitting of the dataset into the training and testing sets.

- (a) Write (and submit) a function *divideset1* that takes the dataset (represented as a matrix) and the probability p_{train} of selecting the data entry (a row in the matrix) into the training set. The function should return two non-overlapping datasets: the training and testing data, such that every entry is selected to the training set randomly with probability p_{train}. Test your *divideset* function on the pima dataset. Run the function 20 times with probability $p_{\text{train}} = 0.66$. Report the size (number of instances) of training sets obtained in each run, and at the end report the average size.
- (b) If your code to part a is correct, you should see some variation in the size of the training sets. Write (and submit) a function *divideset2* that takes the dataset (represented as a matrix) and the probability p_{train}, and returns two non-overlapping datasets: the training and testing data, that mimic closely the distribution defined by p_{train}. Basically, your *divideset2* function should decide first on the number of examples that will go into training and test sets and after that choose randomly examples that will go into each set. The algorithm, if you run it, should always give you different training and test sets but their sizes should stay the same. Hint: use randperm function to implement *divideset2*.