بسمه تعالی

تکلیف اول درس یادگیری ماشین، گروه هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان

تاریخ ارائه: ۱۴۰۰/۱۲/۱ موعد تحویل: ۱۴۰۰/۱۲/۲۳

مسأله ۱ – معرفی

Write/type 2-3 paragraphs about yourself and your background on a separate page (detached from the rest of the homework) . Tell me why you are interested in Machine Learning. Please do not forget to put your name on both parts.

مسأله ۲ – تحلیل اکتشافی داده

In this problem we will explore and analyze the dataset *pima.txt* provided on the course web page. To do the analysis you will need to write short programs. Keep the code you write for future problem sets.

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?
- (b) What are the means and variances of every attribute.
- (c) 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.
- (d) Calculate all correlations between 8 attributes (using the corrcoef function). Which two attributes have the largest mutual correlation in the dataset?
- (e) Assume we want to predict a target class given all attributes. What do you think, does it help or not in prediction to have 2 attributes that are fully correlated? Explain.

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 or their relation, it is often very useful to inspect the data 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 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. Analyze attributes in the data using the function. Which histogram resambles most the normal distribution? In your report show at least two histograms, including the choice you picked as the most normally distributed attribute.
- (g) **2D Scatter plots** plots let us inspect the relations between pairs of attributes. Write 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. Is there a scatter plot that indicates possible linear dependency between two variables? Select two random scatter plots and include them in the report. With every plot include the corresponding attribute names.

مسأله ۳ – پیش پردازش داده

Before applying learning algorithms some data preprocessing may be necessary. To practice Matlab we will write programs for two possible preprocessing tasks: normalization and discretization of continuous values.

- (a) Write a 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. The normalized value should be:

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

  where x is an unnormalized value, μ_x is the mean value of the attribute in the data and σ_x its standard deviation. Test your function on attribute 3 of the pima dataset. Report normalized values of the attribute 3 for the first five entries in the dataset.
- (b) Write a function *discretize_attribute* that takes a vector of attribute values, and assings each value to one of the k bins. Bins are of equal length and should cover the range of values that is 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.

مسأله ۴ – افراز مجموعه داده

In this problem we practice (a) splitting of the dataset along an attribute value and (b) a random splitting of the dataset into the training and testing sets.

- (a) Split *pima.txt* data into two data subsets - one that includes only examples with class label "0", the other one with class "1" values. Calculate and report the mean and standard deviation of each attribute in these two subsets. Hint: try to use Matlab's function *find* to split the data.
- (b) Write 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 nonoverlapping 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 $p_{\text{train}} = 0.66$ and report the average length of the training dataset.
- (c) If your code to part b is correct, you should see some variation in the size of the training sets. Write a function *divideset2* that takes the dataset (represented as a matrix) and the probability p_{train}, and returns two nonoverlapping 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 for the same dataset.

Validation schemes

When testing the performance of a learning algorithm the results may be biased by the training/testing data split. To alleviate the problem various random resampling schemes, such as k-fold crossvalidation, radnom subsampling or bootstrap (see lecture notes for Class 2) can be applied to estimate the statistics of interest by averaging the results accross multiple datasets. Please write and submit the following functions that help you to implement these resampling schemes:

- (a) function [train test] = kfold-crossvalidation(data, k, m) that takes the data, k (the number of folds) and m (the target fold) as inputs, and returns training and

  testing data sets, such that the testing set corresponds to m-th fold under the k-th fold crossvalidation scheme. The file should be named *kfold-crossvalidation.m*
- (b) function [train test] = bootstrap1(data) that implements the bootstrap sampling procedure, and returns the training data that are of the same size as the original data and testing data that consists of examples not selected for training. The file should be named *bootstrap1.m*

مسأله ۵ – مسأله ۷ از فصل دوم کتاب بیشاپ

مسأله ۶ – مسأله ۱۲ از فصل دوم کتاب بیشاپ

مسأله ۷ – توزیع پواسن

The Poisson distribution is used to model the number of random arrivals to a system over a fixed period of time. Examples of systems in which events are determined by random arrivals are: arrivals of customers requesting the service, occurence of natural disasters, such as floods, etc. The Poisson distribution is defined as:

$$p(x|\lambda) = \frac{e^{-\lambda}\lambda^x}{x!} \text{ for } x = 0, 1, 2, \cdots.$$

Answer the following questions:

- (a) Using the definition of the Poisson distribution show that the sum of probabilities of all events is 1. (Hint: Look-up and use the definition of e^λ in terms of a sum).
- (b) Derive the mean of the Poisson distribution.
- (c) Assume we have n independent samples of x. What is the ML estimate of the parameter λ.
- (d) The conjugate prior for the Poisson distribution is Gamma distribution. It is defined as:
$$p(\lambda|a, b) = \frac{1}{b^a\Gamma(a)}\lambda^{(a-1)}e^{-\frac{\lambda}{b}}.$$
Show that the posterior density of the parameter λ is again a Gamma distribution.
- (e) Show that the Poission distribution is a member of the exponential family of distributions (will be covered on Monday 01/26/2015). Give $\eta, T(x), Z(\eta)$ and $h(x)$ components.

Now we are ready to do some Matlab experiments:

- (f) plot the probability function for Poisson distributions with parameters $\lambda = 2$ and $\lambda = 6$. Note that the Poisson model is defined over nonenegative integres only.
- (g) Assume the data in 'poisson.txt' that represent the number of incoming phone calls received over a fixed period of time. Compute and report the ML estimate of the parameter λ.
- (h) Assume the prior on λ is given by $\lambda \sim Gamma(a, b)$. Plot the Gamma distribution for the following set of parameters $(a = 1, b = 2)$ and $(a = 3, b = 5)$.
- (i) Plot the posterior density for λ after observing samples in 'poission.txt' and using priors in part (h). What changes in the distribution do you observe?

مسأله ۸ – رگرسیون خطی

In this problem set we use the Boston Housing dataset from the CMU StatLib Library that concerns prices of housing in Boston suburbs. A data sample consists of 13 attribute values (indicating parameters like crime rate, accessibility to major highways etc.) and the median value of housing in thousands we would like to predict. The data are in the file *housing.txt*, the description of the data is in the file *housing_desc.txt* on the course web page.

Part 1. Exploratory data analysis.

Examine the dataset housing.txt using Matlab. Answer the following questions.

- (a) How many binary attributes are in the data set? List the attributes.
- (b) Calculate and report correlations in between the first 13 attributes (columns) and the target attribute (column 14). What are the attribute names with the highest positive and negative correlations to the target attribute?
- (c) Note that the correlation is a linear measure of similarity. Examine scatter plots for attributes and the target attribute using the function you wrote in problem set 1. Which scatter plot looks the most linear, and which looks the most nonlinear? Plot these scatter plots and briefly (in 1-2 sentences) explain your choice.
- (d) Calculate all correlations between the 14 columns (using the corrcoef function). Which two attributes have the largest mutual correlation in the dataset?

Part 2. Linear regression.

Our goal is to predict the median value of housing based on the values of 13 attributes. For your convenience the data has been divided into two datasets: (1) a training dataset *housing_train.txt* you should use in the learning phase, and (2) a testing dataset *housing_test.txt* to be used for testing.

Assume that we choose a linear regression model to predict the target attribute. Using Matlab:

- (a) Write a function *LR_solve* that takes $\mathbf{X}$ and $\mathbf{y}$ components of the data ($\mathbf{X}$ is a matrix of inputs where rows correspond to examples) and returns a vector of coefficients $\mathbf{w}$ with the minimal mean square fit. (Hint: you can use backslash operator '/' to do the least squares regression directly; check Matlab's help).
- (b) Write a function *LR_predict* that takes input components of the test data ($\mathbf{X}$) and a fixed set of weights ($\mathbf{w}$), and computes vector of linear predictions $\mathbf{y}$.
- (c) Write and submit the program *main3_2.m* that loads the train and test set, learns the weights for the training set, and computes the mean squared error of your predictor on both the training and testing data set. See rules for submission of programs on the course webpage.
- (d) in your report please list the resulting weights, and both mean square errors. Compare the errors for the training and testing set. Which one is better?

Part 3. Online gradient descent

The linear regression model can be also learned using the gradient descent method.

(a) Implement an online gradient descent procedure for finding the regression coefficients w. Your program should:

- start with zero weights (all weights set to 0 at the beginning);
- update weights using the annealed learning rate $2/t$, where t denotes the t-th update step. Thus, for the first data point the learning rate is 2, for the second it is $2/2 = 1$, for the 3-rd is $2/3$ and so on;
- repeat the update procedure for 1000 steps reusing the examples in the training data if neccessary (hint: the index of the i-th example in the training set of size n can be obtained by (i mod n) operation);
- return the final set of weights.

(b) Write a program *main3_3.m* that runs the gradient procedure on the data and at the end prints the mean test and train errors. **Your program should normalize the data before running the method.** Run it and report the results. Give the mean errors for both the training and test set. Is the result better or worse than the one obtained by solving the regression problem exactly.

(c) Run the gradient descent on the un-normalized dataset. What happened?

(d) Modify *main3_3.m* from part b. such that it lets you to progressively observe changes in the mean train and test errors. Use functions *init_progress_graph* and *add_to_progress_graph* on the course web page. The *init_progress_graph* initializes the graph structure and *add_to_progress_graph* lets you add new data entries on-fly to the graph. Using the two functions plot the mean squared errors for the training and test test for every 50 iteration steps. Submit the program and include the graph in the report.

(d) Experiment with the gradient descent procedure. Try to use: fixed learning rate (say 0.05, 0.01), or different number of update steps (say 500 and 3000). You may want to change the learning rate schedule as well. Try for example $2/\sqrt{n}$. Report your results and any interesting behaviors you observe.

Part 4. Regression with polynomials.

Assume we are not happy with the predictive accuracy of the linear model and we decide to explore a more complex model for predicting housing values. Assume we decide to use a quadratic polynomial to model the relation between y and $\mathbf{x}$:

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{13} w_i x_i + \sum_{i=1}^{13} \sum_{j=i}^{13} w_{ij} x_i x_j.$$

- (a) Write a function *extendx* that takes an input $\mathbf{x}$ and returns an expanded x that includes all linear and degree two polynomials.
- (b) What happened to the binary attribute after the transformation?

- (c) Write and submit a Matlab program *main3_4.m* that computes the regression coefficients for the extended input and both train and test errors for the result.
- (d) Report both errors in your report and compare them with the results in part 2. What do you see? Which method would you use for the prediction? Why? Please do not turn in the weights for this part in your report.

کدها و نتایج و نمودارها به صورت چاپ شده همراه با توضیحات کافی و پاسخ‌ها به صورت تفصیلی در قالب یک گزارش فنی (Technical Report) ارائه شوند.

موفق باشید