CS 383 - Machine Learning

Assignment 4 - Classification

Introduction

In this assignment you will implement Naive Bayes and Logistic Regression classifiers for the purpose of binary classification.

You may **not** use any functions from an ML library in your code. And as always your code should work on any dataset that has the same general form as the provided one.

Grading

Although all assignments will be weighed equally in computing your homework grade, below is the grading rubric we will use for this assignment:

Part 1 (Theory)	40pts
Part 2 (Naive Bayes)	25pts
Part 3 (Logistic Regression)	25pts
Report	10pts
TOTAL	100 pts

Datasets

Spambase Dataset (spambase.data) This dataset consists of 4601 instances of data, each with 57 features and a class label designating if the sample is spam or not. The features are *real valued* and are described in much detail here:

https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.names

Data obtained from: https://archive.ics.uci.edu/ml/datasets/Spambase

1 Theory

1. Consider the following set of training examples for an unknown target function: $(x_1, x_2) \rightarrow y$:

Y	x_1	x_2	Count
+	T	T	3
+	T	F	4
+	F	T	4
+	F	F	1
-	T	T	0
-	T	F	1
-	F	T	3
-	F	F	5

   (a) What is the sample entropy, $H(Y)$ from this training data (using log base 2) (5pts)?

   (b) What are the weighted average entropies of the class labels of the subsets created by variables x_1 and x_2 (5pts)?

   (c) Draw the decision tree that would be learned by the ID3 algorithm without pruning from this training data. All leaf nodes should have a single class choice at them. If necessary use the mean class or, in the case of a tie, choose one at random.(10pts)?

2. We decided that maybe we can use the number of characters and the average word length an essay to determine if the student should get an A in a class or not. Below are five samples of this data:

# of Chars	Average Word Length	Give an A
216	5.68	Yes
69	4.78	Yes
302	2.31	No
60	3.16	Yes
393	4.2	No

   (a) What are the class priors, $P(A = Yes), P(A = No)$? (5pts)

   (b) Find the parameters of the Gaussians necessary to do Gaussian Naive Bayes classification on this decision to give an A or not. Standardize the features first over all the data together so that there is no unfair bias towards the features of different scales (5pts).

   (c) Using your response from the prior question, determine if an essay with 242 characters and an average word length of 4.56 should get an A or not. Show the math to support your decision (10pts).

2 Naive Bayes Classifier

For your first programming task, you'll implement, train and test a *Naive Bayes Classifier*.

Download the dataset *spambase.data* from Blackboard. As mentioned in the Datasets area, this dataset contains 4601 rows of data, each with 57 continuous valued features followed by a binary class label (0=not-spam, 1=spam). **Since the features are continuous, we'll use Gaussians to model** $P(x_i|y)$. There is no header information in this file and the data is comma separated. As always, your code should work on any dataset that lacks header information and has several comma-separated continuous-valued features followed by a class id $\in 0, 1$.

Write a script that:

1. Reads in the data.
2. Randomizes the data.
3. Selects the first 2/3 (round up) of the data for training and the remaining for testing
4. Standardizes the data (except for the last column of course) using the training data
5. Divides the training data into two groups: Spam samples, Non-Spam samples.
6. Creates Gaussian models for each feature for each class.
7. Classify each testing sample using these models and choosing the class label based on which class probability is higher.
8. Computes the following statistics using the testing data results:
   (a) Precision
   (b) Recall
   (c) F-measure
   (d) Accuracy (expect around 80%)

Implementation Details

1. Seed the random number generate with zero prior to randomizing the data
2. You may want to consider using the log-exponent trick to avoid underflow issues. Here's a link about it: https://stats.stackexchange.com/questions/105602/example-of-how-the-log-sum-exp-trick-works-in-naive-bayes
3. You also may want to consider removing features with a low standard deviation. They provide little information and can result in extreme spikes in your computation.

3 Logistic Regression

Finally, lets design, implement, train and test a *Logistic Regression Classifier*. For training and testing, we'll use the same dataset as in the previous programming part, and as always, your code should work on any dataset that lacks header information and has several comma-separated continuous-valued features followed by a class id $\in 0, 1$.

Write a script that:

1. Reads in the data.
2. Randomizes the data.
3. Selects the first 2/3 (round up) of the data for training and the remaining for testing
4. Standardizes the data (except for the last column of course) using the training data
5. Trains a logistic classifier.
6. Classify each testing sample using your trained model, choosing an observation to be spam if the output of the model is $\geq 50\%$.
7. Compute the following statistics using the testing data results:
   (a) Precision
   (b) Recall
   (c) F-measure
   (d) Accuracy (expect around 90%)

Implementation Details

1. Seed the random number generate with zero prior to randomizing the data
2. We will let you determine appropriate values for the learning rate, η, the initial parameter values, as well as an appropriate termination criteria.

In your report you will need:

1. The statistics requested for your Logistic Classifier.

Submission

For your submission, upload to Blackboard a single zip file (again no spaces or non-underscore special characters in file or directory names) containing:

1. PDF Writeup
2. Source Code
3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment.

The PDF document should contain the following:

1. Part 1:
   (a) Answers to theory questions
2. Part 2:
   (a) Requested Classification Statistics
3. Part 3:
   (a) Requested Classification Statistics