CS 383 - Machine Learning

Assignment 7 - Support Vector Machines
Winter 2017

Introduction

In this assignment you're going to use a support vector machine package of your choice to train and test an SVM. In addition, we'll look at how to do multi-class classification using a binary classifier (an SVM).

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)	5pts
Part 2 (Binary SVM)	10pts
Part 3 (Multi-class SVM)	10pts
Report	5pts
TOTAL	30pts

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

Cardiotocography Dataset (CTG.csv)

Download the file CTG.csv from Bblearn. This file contains 2126 instances of 21 feature pertaining to information obtained from Cardiotocography tests. Our task is to determine the fetal state class code given an observation. This code can be one of the 3 values and pertains to the LAST column of the dataset. The second to last column of the dataset can also be used for classification but for our purposes DISCARD it.

Your scripts that use this dataset must be able to run on any dataset where the first two rows contain header information, the 2nd to last column is to be discarded, and the last column contains the target class.
You can read more about the dataset here:

http://archive.ics.uci.edu/ml/datasets/Cardiotocography

1 Theory

1. The Linear Kernel is commonly used if we already are working in high feature space. It is defined as $k(X_i, X_j) = \sum_{d=1}^{D} X_{i,d} X_{j,d}$. If your observations have three features ($D = 3$), what is the function $\phi(u)$ such that $k(X_i, X_j) = \phi(X_i)\phi(X_j)^T$? Show your work (4pts).

2. True or false: A Gaussian Kernel is better than a linear kernel when there are many features but few training samples (1pt).

2 Support Vector Machines

Here we are again tackling the binary classification problem of detecting spam, however now we'll try using a Support Vector Machine to do this.

For simplicity you **ARE** allowed to use a support vector machine library and/or functions. In Matlab these are *fitcsvm* to train and *predict* to predict the class labels.

First 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). 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 from the Spambase set provided on Blackboard. Read about how we'll use this dataset in the Datasets section.
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. Train a SVM on this training data using an SVM library of your choice.
6. Classify/test your SVM using the testing set.
7. Computes the following statistics using the testing data results:
   (a) Precision
   (b) Recall
   (c) F-measure
   (d) Accuracy

Implementation Details

1. Seed the random number generate with zero prior to randomizing the data
2. Use default SVM settings unless you have reason to choose others (in which case document your reasoning, and not just "it did better").

In your report you will need:

1. The requested statistics.

Precision:	0.91946
Recall:	0.89097
F-Measure:	0.90484
Accuracy:	0.92825

Table 1: Evaluation for SVM classifier

3 Multi-Class Support Vector Machines

Many learning algorithms are designed to only perform binary classification out-of-the-box. Support vector machines are one such algorithm. If we want to perform multi-class classification we can either use a *one vs all* or *one vs one* approach. In most cases one-vs-one is a better choice so we'll use it.

Again, for this assignment you **ARE** allowed to use a support vector machine library and/or functions. In Matlab these are *fitcsvm* to train and *predict* to predict the class labels. **HOWEVER** you are only allowed to create binary classifiers (you cannot use any multi-class SVM abilities of the library you use).

Write a script that:

1. Reads in the data from the Cardiotocography set provided on Blackboard. Read about how we'll use this dataset in the Datasets section.
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 and evaluates using a *One vs One* approach:
   (a) Train $\frac{K(K-1)}{2}$ one-vs-one binary classifiers where you only use the training samples from the relevant classes. That is, if you are training a classifier that labels observations as either class i or class k, then only use observations with labels i and j (discard the rest).
   (b) For each test sample, run it through each of the $K(K-1)/2$ classifiers to see which class(es) "beat" the others the most. Choose that class as the your observation's label. Again if there is a tie among several classes, choose at random the predicted label from those classes.
   (c) Since there's more than one class, the concept of "Positive" and "Negative" don't really apply. Therefore just compute the *accuracy* as the percentage of samples classified correctly confident.

Implementation Details

1. Seed the random number generate with zero prior to randomizing the data
2. In the case of ties, just choose a class at random (from the ties).
3. Use default SVM settings unless you have reason to choose others (in which case document your reasoning, and not just "it did better").
4. You **MAY NOT** use a multi-class SVM if one is available to you in the library of your choice (that would defeat the purpose of the assignment).
5. Your code should work for an arbitrary number of classes.

In your report you will need:

1. The accuracy for your two approaches.

Method	Accuracy
One-vs-One	0.915

Table 2: Multi-Class Evaluation

Submission

For your submission, upload to Blackboard a single zip file 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 the theory questions.
2. Part 2
    (a) Requested Statistical Information
3. Part 3:
    (a) Requested statistical information