10-601: Homework 4

Due: 18 October 2014 11:59pm (Autolab)

TAs: Qihui Li, Siping Ji

Name: _____

Andrew ID: _____

Please answer to the point, and do not spend time/space giving irrelevant details. Please state any additional assumptions you make while answering the questions. For Questions in this assignment, you need to submit your answers in a single PDF file on autolab, either a scanned handwritten version or a LaTeXpdf file. Please make sure you write legibly for grading. For Implementations, submit your m-files on autolab.

You can work in groups. However, no written notes can be shared, or taken during group discussions. You may ask clarifying questions on Piazza. However, under no circumstances should you reveal any part of the answer publicly on Piazza or any other public website. The intention of this policy is to facilitate learning, not circumvent it. Any incidents of plagiarism will be handled in accordance with CMU's Policy on Academic Integrity.

⋆: Code of Conduct Declaration

- Did you receive any help whatsoever from anyone in solving this assignment? Yes / No.
- If you answered *yes*, give full details: _____ (e.g. *Jane explained to me what is asked in Question 3.4*)
- Did you give any help whatsoever to anyone in solving this assignment? Yes / No.
- If you answered *yes*, give full details: _____ (e.g. *I pointed Joe to section 2.3 to help him with Question 2*).

1 SVM (20 points)(TA:- Siping Ji)

In this section you will implement the polynomial and gaussian kernel, and plug in the kernel with our provided SVM implementation and test it on the dataset.

1.1 The Dataset

A dangerous mutant of bird flu viruses was discovered and a widespread pandemic is imminent. A vaccine was developed for improving immunity against this potent virus. However, this new vaccine is very expensive and only effective for a subset of patients. Expression of two genes X,Y are shown as predictors of the vaccine's effectiveness. You are asked to train a SVM to predict the vaccine's effectiveness on patients using gene expression measurement. Download the dataset from the course website. The .mat file contains two matrices: train and test. There are 160 training examples and 40 testing examples. The dataset used in this section has two dimensional features (it's better for you to first visualize the data and see which SVM kernel makes sense). The task is to use SVM with different kernels to perform binary classification.

1.2 Training SVM

Given a set of training points $x_1, x_2, ..., x_m$ and a set of labels $y_1, ..., y_m$ we want to maximize the margin between decision boundary and support vectors. By incorporating slack variable, the model is more flexible for handling non-separable cases. The constrained optimization problem can be formalized as following:

$$min \qquad C\sum_{n=1}^{N}\xi_n + \frac{1}{2}||\mathbf{w}||^2 \tag{1}$$

$$s.t. \qquad y(x_n)t_n \geq 1 - \xi_n, \qquad n = 1, ..., N \tag{2}$$

$$\xi_n \geq 0, \qquad n = 1, ..., N \tag{3}$$

where C ≥ 0 is a parameter that controls the trade-off between the slack variable penalty and the margin.

1.3 Kernel Trick

Kernel functions can be used in SVM to classify linearly inseparable data in implicit hight dimensional feature space without making explicit feature mapping. Here are several kernel functions that are commonly used.

$$Polynomial: \qquad (x_i \cdot x_j + 1)^d \tag{4}$$

$$Gaussian: \qquad exp(-\gamma||x_i - x_j||^2) \tag{5}$$

(20 points) Implement the kernel functions. Here we already provide you with a svm implementation where you can easily plug in the kernel function. Please implement the polynomial in **polynomial_kernel.m** and gaussian kernel in **gaussian_kernel.m** and have them tested on the dataset we provide to you. **Note:** Under folder svm, function **svm_train.m** and **svm_classify.m**

is provided for training and testing phase of svm. A **svm_runner.m** is also provided to you as an example for accessing these interfaces. **If you are using matlab and encounter the problem about qp solver, please use the commented statement in line 31(which is the qp solver in matlab) instead of line 34 (which is the qp solver in octave).**

2 Compare Classifiers (80 points)(TA:- Qihui Li)

In this part we'll implement several measures for evaluating the performance of classifier. Now that we have learned and implemented several supervised machine learning algorithms, we may wonder how well a certain classifier performs and how they perform against each other regarding a certain problem. We already know we can compare their classification accuracy on a held out test set. But is it a robust measure? Would the result be different if our selection of held out test set is different?

You have three tasks in this section.

1. Implement a function for calculating *Confidence Interval* .
2. Implement functions performing test on *Held-out Set* and test on *Cross Validation* and obtain confidence interval under these 2 tests with the SVM algorithm you've implemented on the vaccine dataset in section 1.
3. Use *t-test* for comparing classifiers and compare Logistic Regression with Neural Network performances on the MNIST dataset from home work 3.

2.1 Confidence Interval

Beside the accuracy, now you also have to output a confidence interval for the accuracy. Here is a detailed tutorial of *ConfidenceInterval*. You should compute confidence interval based on the formula given below:

$$[Accuracy_s(h) - Z_n\sqrt{Accuracy_s(h)(1 - Accuracy_s(h)/n}, Accuracy_s(h) + Z_n\sqrt{Accuracy_s(h)(1 - Accuracy_s(h)/n}$$

where $Accuracy_s(h)$ is the accuracy on sample set from hypothesis h, in this case the sample set is the test set. Z_N is the appropriate number of standard deviation corresponding to the interval level N in Normal distribution which can be looked up in a table, and n is the size of testing set. A Z_N (z-score) look up table is provided *here*.

(10 points)Implement the `ConstructInterval` functions. For this task you need to implement function [Accuracy, lowerInterval, upperInterval] = ConstructInterval(Ypredict,Ytest,confLevel). You only need to compute confidence interval of confidence level 99% and 95%.

2.2 Test on Held-out set

One strong assumption for constructing confidence intervals is that the classifier should be independent from the testing set. (Otherwise it will almost always optimistically biased.)
One way to enforce independence is to draw a partition from training data as testing set and "hold out" this partition during training, so that it is not seen by the classifier learner. However, holding out too much data will weaken the accuracy of the classifier (there is less data in training), while drawing too little data will weaken the accuracy of the test (and lead to a wider confidence interval).

(0 points)Implement the `PartitionHeldOut` functions. For this task you need to implement function `testInstanceLabel = PartitionHeldOut(size, k)` where size is the number of instances in the set and k is the number of partitions you wish to do. Your output testInstanceLabel is a binary vector where its elements is either 0 or 1, indicating which instances are used as testing set. The size of it is the number of instances in the set. In PartitionHeldOut function, you partition the set into k subsets of equal size randomly. You select one of them as your testing set by labelling the corresponding elements of the subset in testInstanceLabel as 1 and the rest as training set by labelling the rest elements in testInstanceLabel as 0.

(20 points)Implement the `TrainHeldOut functions`. For this task you need to implement function `Ypredict = TrainHeldOut(Xtrain, Ytrain, testInstanceLabel)`. In TrainHeldOut function, testInstanceLabel is a binary vector to indicate the training set and testing set. You should use Naive Bayses classifier you implemented in home work 2 for this. Your output Ypredict will only contain test part of instances.

(10 points) Question Combine the train and test sets of the vaccine dataset in section 1as new training dataset. The new training dataset should have 200 instances in total. Using no kernel (or $\phi(x_n) = x_n$)), train a SVM using C = 0.5. Calculate 95% and 99% confidence interval on number of partition 2 and 10 on new training dataset. Describe in one sentence what you observe in the result.

2.3 Test by Cross-Validation

In order to get more testing data, one way is to do cross-validation. For a 10-fold cross- validation, you train the model 10 times, each time using a different 1/10 data as the testing set. In the end, every data point has one prediction (since it was used exactly once as a test case), and we can get the prediction accuracy on the whole training set to construct a confidence interval. Note that for each model, though the classifier is independent from the testing data, those 10 models are actually dependent (due to overlap with the training set). However, in practice, this usually a good approximation.

(0 points)Implement the `PartitionCrossSet` functions. For this task you need to implement function `YcrossSetLabel = PartitionCrossSet(size,k)`. In PartitionCrossSet, you randomly assign instances to one of k sets, and every set should have (as close as possible to) the same number of instances. crossSetLabel is a vector contains number from 1 to k, which indicates the set the instances belong to.

(20 points)Implement the `TrainCrossSet` functions. For this task you need to implement function `Ypredict = TrainCrossSet(Xtrain, Ytrain, crossSetLabel)`.You should use Naive Bayses classifier you implemented in home work 2 for this. You can call TrainHeldOut from your TrainCrossSet. In TrainCrossSet, crossSetLabel indicates how you segment the data to k sets, you should train model k times and every time one partition will be the testing set. Therefore, the size of Ypredict should be the same as Ytrain.

(20 points) Question Combine the train and test sets of the vaccine dataset in section 1 as new training dataset. The new training dataset should have 200 instances in total.Using no kernel (or $\phi(x_n) = x_n$)), train a SVM using different values of C = 0, 0.1, 0.3, 0.5, 1, 2, 5, 8, 10. Use 4-fold cross validation and report the train and test errors. Produce a plot of 2 curves: training and testing error. The x-axis shows different values of C and the y-axis shows the error rate.
Which value of C yields the best training and testing error rate? Which value of C should you use?

2.4 P Values and t-tests

P Values and t-tests are statistical measures for determining the significance of an an event. A detailed tutorial can be found *here*.
To systematically evaluate two classifiers without possible random bias in comparison, we need to look at whether the performance of one classifier is significantly better than the other by statistical measures.
The method is described below.

Divide the testing set into k disjoint subsets. For each subset T_i, we compute the difference of error rates from two classifiers.

$$Y_i = error_{t_i}(h1) - error_{t_i}(h2)$$

Each Y_i is a random variable, so the average $\bar{Y} = \frac{1}{k}\sum_{i=1}^{k} Y_i$ is also a random variable. When k $\geq$ 30, the random variable Y approximates a normal distribution. But if k $<$ 30, it can be approximated as t-distribution with k-1 degrees of freedom.

In the next step, we set a null-hypothesis that $E[\bar{Y}] = 0$. If we can reject this null-hypothesis by two-tail test, it means one classifier is better than the other (either h1 or h2 is better). Or, by one-tail test, we can test if the a specific h is better than the other.

For example, when a given k, and null hypothesis $E[\bar{Y}] = 0$, t-value will be

$$\frac{\bar{y}\sqrt{k}}{S_k}$$

$$S_k^2 = \frac{1}{k-1}\sum_{i=1}^{k}(y_i - \bar{y})^2$$

Then we can see whether we can reject the null hypothesis under different significance levels.

(10 points) Question Use the best set of parameters you've found for Logistic Regression and NN on the MNIST data set from home work 3. Compare Logistic Regression and NN, report their accuracy and p-value under one-tailed test and two-tailed test with k=10. Reason about what you found in two sentence.
Note: You should use the matlab function *ttest* directly.

3 Deliverables

Submit your codes in three parts via AutoLab. You can use the your codes from previous assignments as classifiers. You are NOT allowed to use any off-the-shelf optimizer. You should upload

your codes (including all your function files) along with a report, which should solve the questions above.
You should tar gzip the following items into hw4part1.tgz and submit to the Homework 4 - Construct Interval under Autolab:

- ConstructInterval.m
- and all other auxiliary functions you have written

You should tar gzip the following items into hw4part2.tgz and submit to the Homework 4 - Cross-Set + Held-Out under Autolab:

- TrainHeldOut.m
- TrainCrossSet.m
- and all other auxiliary functions you have written

You should tar gzip the following items into hw4part3.tgz and submit to the Homework 4 - SVM under Autolab:

- polynomial_kernel.m
- gaussian_kernel.m
- and all other auxiliary functions you have written
- report.pdf

Please submit a file called `hw4part3.tgz`. Please place all your code in a folder `hw4part3` and then tar the folder using the following instructions :
Tar gzip the files directly using `tar -cvf hw4part3.tgz *.m report.pdf`. Do NOT put the above files in a folder and then tar gzip the folder. You do not need to upload the saved predicted labels (i.e. the .mat files). Please make sure your code is working fine under Octave before you submit.
Note: You should not include *svm_train.m svm_classify.m svm_runner.m Partition_CrossSet.m Partition_HeldOut.m nb_train.m or nb_test.m* in your homework submissions though you do need to call these functions in the files you submit.