CS57800 Statistical Machine Learning

Homework 3

Due: Nov 10, 2015 on Tuesday

1 Questions(50pts)

1. (8pts) Let C be a concept space that consists of all polygons with n vertices in the plane. Give the VC dimension of C and prove its correctness.

2. (8pts) Logistic loss function is defined in terms of the true label y and the predicted value $\hat{y} = \omega \cdot x + b$ as follows
$$\ell(y, \hat{y}) = \frac{1}{\log 2} \log(1 + exp(-y\hat{y}))$$
Show that logistic loss is convex for a fixed value of $y \in \pm 1$ and as a function of $\hat{y}$.

3. (8pts) Suppose you run AdaBoost on n training samples and the weighted training error ϵ_t of the t^{th} weak hypothesis is at most γ where $0 < \gamma < 0.5$. Then, find the number of iterations T that you need in order to make the final hypothesis H produce no training error. The number of iterations T should be expreseed in terms of n and γ.

4. (10pts) Suppose you are given 10 two-dimensional points presented in the table below and each point is labeled with either $+$ or $-$. You will use AdaBoost to learn a function mapping points in $\mathbb{R}$ to a boolean value. The space of weak learner considered by the algorithms consists of hypotheses of the form: $x_i > A$ where A is an integer and $i = \{1, 2\}$. Run the AdaBoost algorithm for two rounds using the data points. At each round AdaBoost chooses the weak learner that minimizes the error ϵ. Your answer should consist of:
(1) The weak hypothesis used at each round, and its error.
(2) The distribution D_t over the data points for each round.
(3) The final hypothesis after running two rounds.

index	x_1	x_2	y
1	1	10	-
2	4	4	-
3	8	7	+
4	5	6	-
5	3	16	-
6	7	7	+
7	10	14	+
8	4	2	-
9	4	10	+
10	8	8	-

5. (8pts) If $K_1(\vec{x}, \vec{y})$ and $K_2(\vec{x}, \vec{y})$ are both valid kernel functions, with positive α and β, prove that

$$K(\vec{x}, \vec{y}) = \alpha K_1(\vec{x}, \vec{y}) + \beta K_2(\vec{x}, \vec{y})$$

is also a kernel function.

6. (8pts) Recall that the inequality constraints for the non-separable case in SVM are given by

$$x_i \cdot w + b \geq 1 - \xi_i \text{ for } y_i + 1$$

$$x_i \cdot w + b \leq -1 + \xi_i \text{ for } y_i - 1$$

$$\xi_i \geq \forall i$$

where $x_i (i = 1, \cdots M)$ are the M data points with associated labels $y_i \in \{+1, -1\}$ and associated margins ξ_i. Show that $\sum_{i=1}^{M} \xi_i$ is an upper bound on the training error of the classifier(The error is 1 if a point is misclassified, 0 otherwise).

2 Programming Assignment(50pts)

General Programing Requirements

You should implement your solution using Python or JAVA. You may **not** use anybody else's code; you **must** implement your own version! We will use **MOSS** (Measure Of Software Similarity) from Stanford University to check for plagiarism. You should submit your source code files along with your typed HW report. Any external packages are **not** allowed (e.g. scikit-learn for python) except numpy and scipy, but feel free to use any internal packages (e.g. sys, os, math in python). The TAs should be able to compile and run your code. (And also keep in mind the JAVA version on data.cs.purdue.edu is 1.8 while python is 2.7)

Task Description

In this section, you are required to implement the GD algorithm over the movie review dataset that is the same as one used for HW2.

1. **Feature Representation** As in the previous assignment, your implementation should consider three variations of the feature set as described below

   (1) unigram
   (2) bigram
   (3) use both

2. **Implementation** Implement the GD algorithm using hinge loss by completing the function *GD(maxIterations, regularization, stepSize, lmbd, featureSet)*. The first argument determines the number of training iterations the algorithm will perform over the dataset. The *regularization* argument determines which regularizer the GD algorithm should use, possible values are $l1, l2$. The *stepSize* argument(η) determines the step size taken by GD algorithm. The argument *lmbd*(λ) determines the impact of the regularizer compared to the training loss. The *featureSet* argument determines which feature representation will be used. The argument takes integer values $(1, 2, 3)$ corresponding to three feature representations described above.

3. **Experiments** In addition to the code implementation you should also submit a short report describing the results of your algorithm. For each feature representation, report the accuracy, precision, recall and F1 as described in class, over the training, validation and testing sets used in the previous assignment. Your report should include the values of hyper parameters and an explanation of how was the value picked for each feature representation. You should specifically address the impact your choice of regularizer has on each feature set.

Submission Instructions:

You are required to use LaTeX to type your solutions to questions, and report of your programing as well. Other formats of submission will **not** be accepted. A template named "homework.tex" is also provided for your convenience.

After logging into data.cs.purdue.edu (physically go to the lab or use ssh remotely, you are all granted the accounts to CS data machines during this class), please follow these steps to submit your assignment:

1. Make a directory named *'your Name_your Surname'* and copy all of your files there.

2. While in the upper level directory (if the files are in /homes/dan/dan_goldwasser, go to/homes/dan), execute the following command:

```
turnin -c cs578 -p HW3 *your_folder_name*
```

(e.g. your prof would use: `turnin -c cs578 -p HW3 dan_goldwasser` to submit his work)

Keep in mind that old submissions are overwritten with new ones whenever you execute this command.

3. You can verify the contents of your submission by executing the following command:

   ```
   turnin -v -c cs578 -p HW3
   ```

   Do **not** forget the -v flag here, as otherwise your submission would be replaced with an empty one.

Your submission should include the following files:

1. The source code in python or java (and the executable .jar file for java).
2. Your evaluation & analysis in .pdf format (together with your HW solutions in it).
3. A README file containing your name, instructions to run your code and anything if you would like us to know about your program (like errors, special conditions, etc).