CSCI 5525: Machine Learning (Fall'13)

Homework 2, Due 11/01/13

1. **(25 points)** Consider the single layer perceptron with a sigmoid transfer function, i.e., for input $\mathbf{x} \in \mathbb{R}^d$, the predicted output

$$\hat{y}(\mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})} ,$$

and if $y \in \{0, 1\}$ is the true class label, the prediction error is measured by

$$E(\mathbf{w}) = (y - \hat{y}(\mathbf{w}))^2 . \tag{1}$$

   (a) (10 points) Show that $E(\mathbf{w})$ is not necessarily a convex function of $\mathbf{w}$.

   (b) (10 points) Consider replacing the loss function from square loss $(y - \hat{y}(\mathbf{w}))^2$ to Bernoulli relative entropy:

$$E(\mathbf{w}) = y \log \frac{y}{\hat{y}(\mathbf{w})} + (1 - y) \log \frac{(1 - y)}{(1 - \hat{y}(\mathbf{w}))} .$$

   Show that the above choice indeed leads to a convex problem in $\mathbf{w}$.[1]

   (c) (5 points) Show that the modification in (b) above exactly leads to the 2-class logistic regression problem. In particular, show that the objective functions for the modification in (b) and 2-class logistic regression are the same.

2. **(25 points)** Recall that a function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a valid kernel if it is symmetric, i.e., $k(x, x') = k(x', x)$ and positive semi-definite, i.e., $\forall x_1, \ldots, x_n$, the matrix K with $K_{ij} = k(x_i, x_j)$ is positive semi-definite. For the current problem, we assume that the domain $\mathcal{X} = \mathbb{R}$.

   (a) (10 points) Let $k_1, \ldots, k_m$ be a set of valid kernels. Show that for any $w_j \geq 0, j = 1, \ldots, m$, the function $k = \sum_{j=1}^{m} w_j k_j$ is a valid kernel.

   (b) (15 points) Consider the function $k(x, x') = \exp(-(x - x')^2)$ where $x, x' \in \mathbb{R}$. Show that k is a valid kernel.

3. **(20 points)** The Mushroom dataset has 2 classes (edible, poisonous) with 8124 samples, each having 22 nominal features. Feature 11 (stalk-root) has missing values, and will be ignored for the homework (see additional instructions below). Train and evaluate the following classifiers using 10-fold cross-validation:

   (a) (10 points) A decision stump, i.e., 1 layer decision tree, using Information Gain.

   (b) (10 points) A 2 layer decision tree using Gini Index.

   You will have to submit (i) **summary of methods and results** report, and (ii) **code** for each algorithm:

[1] You can use $0 \log 0 = 0$ for simplifying the expression.

(i) **Summary of methods and results:** Briefly describe the approaches in (a) and (b) along with equations for the splitting criterion. Also, for both (a) and (b), report the training set and test set error rates for each fold along with the average error rate and standard deviation across all folds for training and test sets. Finally, include a figure corresponding to the decision tree generated on the entire dataset (without cross-validation) for each case.

(ii) **Code:** For part (a), you will have to submit code for `dstumpIG(filename)` (main file). This main file has **input**: filename for the dataset, and **output**: print to the terminal (stdout) the training set and test set error rates for each fold for 10-fold cross-validation along with the average error rate and standard deviation for training and test sets. The function *must* take the inputs in this order and display the output via the terminal.

The filename will correspond to a plain text file for a dataset, with each line corresponding to a data point: the first entry will be the label and the rest of the entries will be feature values of the data point.

For part (b), you will have to submit code for `dtree2GI(filename)` (main file), with other guidelines staying the same.

For each part, you can submit additional files/functions (as needed) which will be used by the main file. Put comments in your code so that one can follow the key parts and steps in your code.

4. **(30 points)** We consider boosting using different loss functions, and evaluate their performance on the Mushroom dataset using decision stumps.

(a) (10 points) Train and evaluate the adaboost classifier using decision stumps using 10-fold cross-validation.

(b) (10 points) Recall that an additive model constructed using the exponential loss function $L(y, f(x)) = \exp(-yf(x))$ gives Adaboost. Derive the corresponding additive model (known as logitboost) using the logistic loss function $L(y, f(x)) = \log(1+\exp(-yf(x)))$.

(c) (10 points) Train and evaluate the logitboost classifier using decision stumps using 10-fold cross-validation.

Both boosting algorithms should be run with the following number of decision stumps in the additive model: 5, 10, 20, and 40. For each algorithm (adaboost and logitboost), please plot the average test-set error rate over 10 folds along with standard deviation bars with increasing number of decision stumps.

You will have to submit (i) **summary of methods and results** report, and (ii) **code** for each algorithm:

(i) **Summary of methods and results:** Briefly describe the algorithms in (a) and (c) along with necessary update equations. Also, for both (a) and (c), report the training set and test set error rates for each fold along with the average error rate and standard deviation across all folds for training and test sets for each increasing number of decision stumps ($T = 5, 10, 20, 40$).

(ii) **Code:** For part (a), you will have to submit code for `myAdaBoost(filename,` T`)` (main file). This main file has **input**: (1) filename for the dataset and (2) the number T of decision stumps, and **output** print to the terminal (stdout) the training set and test set error rates for each fold in 10-fold cross-validation along with the average error rate and standard deviation across all folds for each increasing number of decision stumps. The function *must* take the inputs in this order and display the output via the terminal.

The filename will correspond to a plain text file for a dataset, with each line corresponding to a data point: the first entry will be the label and the rest of the entries will be feature values of the data point.

For part (c), you will have to submit code for `myLogitBoost(filename, `T`)` (main file), with other guidelines staying the same.

For each part, you can submit additional files/functions (as needed) which will be used by the main file. Put comments in your code so that one can follow the key parts and steps in your code.

Additional instructions: The Mushroom dataset has missing values for feature 11 (stalk-root) which corresponds to column 12 in the dataset. Please preprocess the dataset by removing this feature for both Problems 3 and 4 *in your code.* We will test with the original dataset which will include this feature so please allow your code to preprocess the dataset. Code can only be written in C/C++, Java, Matlab, or Python, no other programming languages will be accepted. All programs must be able to be executed from the terminal command prompt. Please specify instructions on how to run your program in the README file. Information on the size of the datasets, including number of data points and dimensionality of features, as well as number of classes can be readily extracted from the dataset text file.

Evaluation notes: Code will be tested on the provided dataset and possibly other similar (2 class, multivariate, categorical) datasets. Correctness of code, error rates, and standard deviations will be evaluated as well as the discussion of methods used and results.

Instructions

Follow the rules strictly. If we cannot run your functions, you get 0 points.

- **Things to submit**

  1. hw2.pdf: A document which contains the solutions to Problems 1, 2, 3, and 4, including the summary of methods and results.
  2. `dstumpIG` and `dtree2GI`: Code for Problem 3.
  3. `myAdaBoost` and `myLogitBoost`: Code for Problem 4.
  4. README.txt: README file that contains your name, student ID, email, instructions on how to compile (if necessary) and run your code, any assumptions you are making, and any other necessary details.
  5. Any other files, except the data, which are necessary for your program.