CSCI 5525: Machine Learning (Fall'15)

Homework 3, Due 11/13/15

1. **(15 points)** Let $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a dataset for a 2-class classification problem, where $y_i \in \{-1, +1\}$. We consider developing boosting algorithms for the problem. At iteration t of boosting, the distribution over the dataset is denoted by w_t, and we assume access to a weak learning algorithm which generates a classifier G_t whose error rate $\epsilon_t = P_{x \sim w_t}[G_t(x) \neq y] \leq (\frac{1}{2} - \frac{\gamma}{2})$, for all t, where $0 < \gamma \leq 1$.

   (a) (10 points) Let $g(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t G_t(x)\right)$. For Adaboost, show that the training error-rate satisfies the following inequality:

   $$\frac{1}{N}\sum_{i=1}^N \mathbb{1}(g(x_i) \neq y_i) \leq \exp\left(-\frac{\gamma^2}{2}T\right) ,$$

   where $\mathbb{1}(\cdot)$ denotes the indicator function, which is 1 if the argument is true, and 0 otherwise.

   (b) (5 points) Assuming that we can always get a weak classifier G_t whose error-rate satisfies $\epsilon_t \leq (\frac{1}{2} - \frac{\gamma}{2})$ for $0 < \gamma \leq 1$, will the training error-rate $\frac{1}{N}\sum_{i=1}^N \mathbb{1}(g(x_i) \neq y_i)$ always become zero as T is increased, i.e., we keep adding more weak classifiers? Clearly explain your answer.

2. **(40 points)** The Ionosphere dataset has 2 classes (good radar returns +1, bad radar returns -1, labels are the last column of the data file.) with 351 samples, each having 34 continuous features. Train and evaluate the following classifiers using 10-fold cross-validation:

   (a) (20 points) Bagged 2-layer decision trees with binary splits using Information Gain with the following number of base classifiers: $B = [5, 10, 15, \ldots, 50]$.

   (b) (20 points) Random forest of 100 2-layer decision trees with binary splits using Information Gain with the size of the random feature set $M = [1, 2, \ldots, 34]$.

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

   (i) **Summary of methods and results:** Briefly describe the algorithms in (a) and (b) along with necessary equations, e.g., for splitting on a feature.
   For (a),

   - Provide a table of the training and test set error rates for each fold and number of base classifiers. This will be a 20×10 table. Row $2i - 1$ and $2i$ are corresponding to ith train and test error respectively and columns pertains to different number of base classifiers, i.e., $B = [5, 10, 15, \ldots, 50]$.

- Also provide the training and test set average error rates and standard deviation across all folds for each number of base classifiers. This can be two rows under the above table, providing mean and standard deviation of each column.
- Finally, include a graph of the training and test set average error rates as the number of base classifiers increase.

For (b),

- Provide a table of the training and test set error rates for each fold and value of M. Similar to the above specification, the table will be 20×34.
- Also provide the training and test set average error rates and standard deviation across all folds for each value of M. This will add two other rows to the above table.
- Finally, include a graph of the training and test set average error rates as the value of M increases.

The graphs *must* have a title, labeled axis, and labeled curves. Finally, thoroughly discuss the results of each method.

(ii) **Code:** For part (a), you will have to submit code for `myBagging2(filename,B, k)` (main file). This main file has **input**:filename for the dataset and a vector B for the number of base classifiers, e.g., $B = [5, 10, 15, \ldots, 50]$, and k as the number of folds in k-fold cross validation and **output**: print to the terminal (stdout) the training and test set error rates and number of base classifiers for each fold of the k-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.

For part (b), you will have to submit code for `myRForest2(filename,M, k)` (main file). This main file has **input**: filename for the dataset and a vector M of the size of the random feature set, e.g., $M = [1, 2, \ldots, 34]$, and k as the number of folds in k-fold cross validation and **output**: print to the terminal (stdout) the training and test set error rates and feature set size for each fold of the k-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. Although in your report you should provide the results for specific value of vectors B and M and number k, your code should be general and accept any input arguments and produce the required results without errors.

Sample input/output in MATLAB for part a:

```
 >>myBagging2("D:\MNIST-13", [10, 50], 2)
Train error for fold 1 with 10 base classifiers:  .281
Test error for fold 1 with 10 base classifiers:  .351
Train error for fold 2 with 10 base classifiers:  .322
Test error for fold 2 with 10 base classifiers:  .371
-----------------------------------------------------
Train error for fold 1 with 50 base classifiers:  .181
Test error for fold 1 with 50 base classifiers:  .251
Train error for fold 2 with 50 base classifiers:  .222
```

```
Test error for fold 2 with 50 base classifiers:  .271
```

Sample input/output test for Python for part a:

```
 $python myBagging2.py /export/MNIST-13 [10,50], 2
Train error for fold 1 with 10 base classifiers:  .281
Test error for fold 1 with 10 base classifiers:  .351
Train error for fold 2 with 10 base classifiers:  .322
Test error for fold 2 with 10 base classifiers:  .371
-----------------------------------------------------
Train error for fold 1 with 50 base classifiers:  .181
Test error for fold 1 with 50 base classifiers:  .251
Train error for fold 2 with 50 base classifiers:  .222
Test error for fold 2 with 50 base classifiers:  .271
```

Sample input/output in MATLAB for part b:

```
 >>myRForest2("D:\MNIST-13", [11, 5], 2)
Train error for fold 1 with 11 random features:  .281
Test error for fold 1 with 11 random features:  .351
Train error for fold 2 with 11 random features:  .322
Test error for fold 2 with 11 random features:  .371
-----------------------------------------------------
Train error for fold 1 with 5 random features:  .381
Test error for fold 1 with 5 random features:  .451
Train error for fold 2 with 5 random features:  .422
Test error for fold 2 with 5 random features:  .471
```

Sample input/output test for Python for part b:

```
 $python myRForest2.py /export/MNIST-13 [11,5], 2
Train error for fold 1 with 11 random features:  .281
Test error for fold 1 with 11 random features:  .351
Train error for fold 2 with 11 random features:  .322
Test error for fold 2 with 11 random features:  .371
-----------------------------------------------------
Train error for fold 1 with 5 random features:  .381
Test error for fold 1 with 5 random features:  .451
Train error for fold 2 with 5 random features:  .422
Test error for fold 2 with 5 random features:  .471
```

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.

3. **(30 points)** Consider a two class classification problem setting with training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $y_i \in \{0, 1\}$ and $\mathbf{x}_i \in \mathbb{R}^d$. Consider a linear activation model $a_i = a(\mathbf{x}_i, \mathbf{w}) = \mathbf{w}^T\mathbf{x}_i$, and

activation functions of the form

$$f_{\text{sigmoid}}(a_i) = \frac{1}{1+\exp(-a_i)} , \quad (1)$$

$$f_{\text{relu}}(a_i) = \max(0, a_i) . \quad (2)$$

(a) (15 points) The square loss on the entire dataset in terms of the activation function f_{sigmoid} applied to all the activations $\mathbf{a} = [a_1 \ \ldots \ a_n]$ is given by

$$L_{sq}^{(\text{sigmoid})}(\mathbf{a}) = \sum_{i=1}^{n} \left(y_i - f_{\text{sigmoid}}(a_i)\right)^2 .$$

Is $L_{sq}^{(\text{sigmoid})}$ a convex function of the activation vector $\mathbf{a}$? Clearly explain/prove your answer with necessary (mathematical) details, including the definition of convexity you are using.

(b) (15 points) The square loss on the entire dataset in terms of the activation function f_{relu} applied to all the activations $\mathbf{a} = [a_1 \ \ldots \ a_n]$ is given by

$$L_{sq}^{(\text{relu})}(\mathbf{a}) = \sum_{i=1}^{n} \left(y_i - f_{\text{relu}}(a_i)\right)^2 .$$

Is $L_{sq}^{(\text{relu})}$ a convex function of the activation vector $\mathbf{a}$? Clearly explain/prove your answer with necessary (mathematical) details, including the definition of convexity you are using.

4. **(15 points)** The problem considers the convolution operation in a convolutional neural network as a matrix operation. Let the input $X \in \mathbb{R}^{n \times m}$ be a $n \times m$ real valued matrix. Let $K \in \mathbb{R}^{3 \times 3}$ be a kernel applied to the input X without any zero-padding or other extension, so that the output $Z = (X * K) \in \mathbb{R}^{n-2 \times m-2}$ is a $(n-2) \times (m-2)$ matrix.

(a) (10 points) Using pseudo-code, clearly describe the convolution operation which takes any input matrix X and kernel matrix K as input, and outputs matrix Z.

(b) (5 points) Let $\text{vec}(X) \in \mathbb{R}^{nm}$ be a vectorized version of matrix X constructed column-wise, i.e., $\text{vec}(X)[1:n] = X[1:n, 1]$, $\text{vec}(X)[n+1:2n] = X[1:n, 2]$, and so on. Let $A \in \mathbb{R}^{(n-2)(m-2) \times (nm)}$ matrix such that $\text{vec}(Z) = A\text{vec}(X)$. Specify the matrix A in terms of the kernel matrix K, i.e., which entries of A will correspond to which entries of K, and which entries are 0.

Additional instructions: All outside material used *must* be cited. Code can only be written in 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.

Evaluation notes: Code will be tested on the provided dataset and possibly other similar (2 class, multivariate, continuous) 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. hw3.pdf: A document which contains the solutions to Problems 1, 2, 3, and 4, including the summary of methods and results.

2. `myBagging2` and `myRForest2`: Code for Problem 1.

3. 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.

4. Any other files, except the data, which are necessary for your program.