CSCI 5525: Machine Learning (Fall'16)

Homework 3, Due 11/11/16

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. **(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.

3. **(15 points)** Recall that gradient boosting can work with any loss function $L(y, F(\mathbf{x}))$, where the additive model $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$. Consider a 2-class classification setting, where the loss for gradient boosting is the logistic loss, i.e.,

$$L(y, F(\mathbf{x})) = \sum_{i=1}^{n} -y_i F(\mathbf{x}_i) + \log(1 + \exp(y_i F(\mathbf{x}_i))$$

   (i) (10 points) Assuming the current function to be $F_{m-1}(\mathbf{x})$, what is the gradient $g_m(\mathbf{x}_i)$ computed at $F(\mathbf{x}) = F_{m-1}(\mathbf{x})$ at point $\mathbf{x}_i$? What is the optimization problem which needs to be solved to obtain $h(\mathbf{x}, \mathbf{a}_m)$?

   (ii) (5 points) What is the optimization problem which needs to be solved to obtain the step size ρ_m for the additive model $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$? Does the problem have a closed form solution? Clearly explain your answer.

Programming assignments: The next problem involves programming. We will be using the Boston housing dataset. The dataset has 506 points, 13 features, and 1 target (response) variable. You can find more information about the dataset here:
https://archive.ics.uci.edu/ml/datasets/Housing

While the original dataset is for a regression problem, we will create two classification datasets for the homework. Note that you only need to work with the `target` t to create these classification dataset, the `data` X should not be changed.

a. `Boston50`: Let τ_{50} be the median (50th percentile) over all t (response) values. Create a 2-class classification problem such that $y = 1$ if $t \geq \tau_{50}$ and $y = 0$ if $t < \tau_{50}$. By construction, note that the class priors will be $p(y = 1) \approx \frac{1}{2}, p(y = 0) \approx \frac{1}{2}$.

b. `Boston75`: Let τ_{75} be the 75th percentile over all t (response) values. Create a 2-class classification problem such that $y = 1$ if $t \geq \tau_{75}$ and $y = 0$ if $t < \tau_{75}$. By construction, note that the class priors will be $p(y = 1) \approx \frac{1}{4}, p(y = 0) \approx \frac{3}{4}$.

***The original Boston dataset is provided on Moodle course page in csv format. The last column shows target values, which is continuous (house price). All other columns are data values. Each row is a input point. Same as HW1, your functions (main) will take the original Boston dataset as input and output results for both `Boston50` and `Boston75` in one run.**

4. **(40 points)** Train and evaluate the following classifiers using 10-fold cross-validation:

   (a) (20 points) Adaboost using 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 $M = [1, 2, \ldots, 13]$. Note that for $M = 13$, we get a bagged trees.

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 $2i$ should correspond to the ith fold 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 plot 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 20×13.
- 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 plot 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, briefly discuss the results of each method.

***Guidelines on feature split:** In the `Boston` dataset, there are in total 13 features, and only the fourth feature (column) has binary values whereas all others have continuous values (non-categorical). During feature selection, we will use **binary split** for each feature with continuous values. To find the best split, enumerate the $[10, 20, ..., 90]^{th}$ percentiles for each continuous feature for data points belonging to the current tree-node. The percentiles will be treated as possible split points for each feature. The best-split maximizes the information gain for this feature. For example, if your feature set has 13 features to choose from, then for each feature with continuous values, evaluate all possible split-values (as given by the 9 percentiles), and choose the best split point for each feature with its information gain recorded (**within feature**). Next, choose the best feature from your feature set that overall maximizes the information gain (**across features**). Note that the split-values based on percentiles cannot be predetermined. They need to be computed on-the-fly using the data points you have at the current tree node.

(ii) **Code:** For part (a), you will have to submit code for `myABoost(filename,B, k)` (main file). This main file has **input**: a filename (including extension and absolute path) containing 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 `myRForest(filename,M, k)` (main file). This main file has **input**: a filename (including extension and absolute path) containing the dataset, a vector M of the size of the random feature set, e.g., $M = [1, 2, \ldots, 13]$, 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.

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.

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: Code can only be written in Python or Matlab; no other programming languages will be accepted. One should be able to execute all programs from matlab/python command prompt or the code can be a jupyter notebook. Your code must run on a CSE lab machine (e.g., csel-kh1260-01.cselabs.umn.edu). 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.

Each function must take the inputs in the order specified in the problem and display the textual output via the terminal. The input data file for your function must be exactly the same as the original downloaded file, which will be used as input for grading.

For each part, you can submit additional files/functions (as needed) which will be used by the main file. In your code, you cannot use machine learning libraries such as those available from scikit-learn for learning the models or for cross-validation. However, you may use libraries for basic matrix computations. Put comments in your code so that one can follow the key parts and steps in your code.

Extra Credit problem:

EC1 **(20 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) (10 points) Let $\text{vec}(X) \in \mathbb{R}^{nm}$ be a vectorized version of matrix X constructed columnwise, 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.

Instructions

Follow the rules strictly. If we cannot run your code, you will not get any credit.

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. `myABoost` and `myRForest`: Code for Problem 4.
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.