CS1675: Homework 8

Due: 11/29/2018, 11:59pm

This assignment is worth 50 points.

In this exercise, you will implement a decision stump (a very basic classifier) and a boosting algorithm. You will also complete an exercise to help review basic probability, in preparation for discussing probabilistic graphical models.

Part I: Decision stumps (15 points)

Implement a set of decision stumps in a function `decision_stump_set`.

Instructions:

- [5 pts] Each decision stump operates on a single feature dimension and uses a threshold over that feature dimension to make positive/negative predictions. This function should iterate over all feature dimensions, and consider 10 approximately equally spaced thresholds for each feature.
- [3 pts] If the feature value for that dimension of some sample is over/under that threshold (using "over" defines one classifier, and using "under" defines another), we classify it as positive (+1), otherwise as negative (-1).
- [5 pts] After iterating over all combinations, the function should pick the best among these Dx10x2 classifiers, i.e. the classifier with highest weighted accuracy.
- [2 pts] Finally, for simplicity, rather than defining a separate function, we will use this one to output the label on the test samples, using the best combination of feature dimension, threshold, and over/under.

Inputs:

- an NxD matrix `X_train` (N training samples, D features),
- an Nx1 vector `y_train` of ground-truth labels for the training set,
- an Nx1 vector `w_train` containing the weights for the N training samples, and
- an MxD matrix `X_test` (M test samples, D features).

Outputs:

- an Nx1 binary vector `correct_train` containing 1 for training samples that are correctly classified by the best decision stump, and 0 for incorrectly classified training samples, and
- an Mx1 vector `y_pred` containing the label predictions on the test set.

Part II: AdaBoost (20 points)

In a function `adaboost`, implement the AdaBoost method defined on pages 658-659 in Bishop (Section 14.3). Use decision stumps as your weak classifiers. If some classifier produces an α value less than 0, set the latter to 0 (which effectively discards this classifier) and exit the iteration loop.

Instructions:

1. [3 pts] Initialize all weights to 1/N. Then iterate:
2. [7 pts] Find the best decision stump, and evaluate the quantities ε and α.
3. [7 pts] Recompute and normalize the weights.
4. [3 pts] Compute the final labels on the test set, using all classifiers (one per iteration).

Inputs:

- `X_train`, `y_train`, `X_test`, and
- a scalar `iters` defining how many iterations of AdaBoost to run (denoted as M in Bishop).

Outputs:

- an `M`x1 vector `y_pred_final`, containing the final labels on the test set, using all `iters` classifiers.

Part III: Testing boosting on Pima Indians (10 pts)

In a script `adaboost_demo.m`, test the performance of your AdaBoost method on the Pima Indians dataset. Use the train/test split code (10-fold cross-validation) from HW4. Convert all 0 labels to -1. Try employing (10, 20, 50) iterations. Compute and report (in `report.pdf/docx`) the accuracy on the test set, using the final test set labels computed above.

Part IV: Probability review (5 points)

In your report file, complete Bishop Exercise 1.3. Show your work.

Submission: Please include the following files:

- `decision_stump_set.m`
- `adaboost.m`
- `adaboost_demo.m`
- `report.pdf/docx`