RECITATION 10: ENSEMBLE METHODS

10-301/10-601 Introduction to Machine Learning (Summer 2023)
http://www.cs.cmu.edu/~hchai2/courses/10601
Released: August 3rd, 2023
Quiz Date: August 8th, 2023
TAs: Alex, Andrew, Sofia, Tara, and Neural the Narwhal

1 Ensemble Methods

The idea of ensemble methods is to build a model for prediction by combining the strengths of a group of simpler models. We'll cover two examples of ensemble methods: random forests and AdaBoost.

1.1 Random Forests

1. What are some downsides of decision trees, and how can we explain this in the context of the bias-variance tradeoff?

Random Forests = Sample Bagging + Split-Feature Randomization

2. What is **sample bagging**?

3. What is **split-feature randomization**?

4. How do these techniques affect the bias and variance of an individual tree?

5. How do these techniques affect the bias and variance of an ensemble of trees?

6. For each data point $\mathbf{x}^{(i)}$, define $t^{(-i)}$ to be the set of decision trees that $\mathbf{x}^{(i)}$ was not used to train. Use each tree in $t^{(-i)}$ to make a prediction for $\mathbf{x}^{(i)}$, and use these predictions to make an aggregated prediction $\overline{t^{(-i)}}(\mathbf{x}^{(i)})$ (i.e. for classification take the majority vote). Then, we can define the *out-of-bag* error as follows:

$$E_{OOB} = \frac{1}{N}\sum_{i=1}^{N} 1\left(\overline{t^{(-i)}}(\mathbf{x}^{(i)}) \neq y^{(i)}\right)$$

Why can we use E_{OOB} for hyperparameter optimization even though it was calculated using training points we used to learn the decision trees with?

7. **Random Forest Example:** Suppose we train a random forest with two decision trees on the following dataset, using the provided bootstrap samples. Assume that for ties, we predict $Y = 1$.

All	X_0	X_1	X_2	X_3	Y
1	1	0	0	0	1
2	0	0	1	0	1
3	0	0	0	1	1
4	0	0	0	0	0
5	0	1	0	1	1

Sample 1	X_0	X_1	X_2	X_3	Y
1	1	0	0	0	1
4	0	0	0	0	0
5	0	1	0	1	1

Sample 2	X_0	X_1	X_2	X_3	Y
3	0	0	0	1	1
4	0	0	0	0	0
5	0	1	0	1	1

(a) Suppose we train our first tree on Sample 1 and the split feature randomization chooses $\{X_1, X_2\}$ for the feature candidates at the root. What feature will we split on at the root?

(b) Suppose we then recurse on the left child (with feature value 0) of the root and split feature randomization chooses $\{X_0, X_2\}$ for the feature indices. What feature will we split on?

(c) Suppose we train our second tree on Sample 2 and the split feature randomization chooses $\{X_2, X_3\}$ for the feature candidates at the root. What feature will we split on at the root?

(d) What is the training error of the ensemble?

(e) What is the out of bag error of the ensemble?

1.2 AdaBoost

1.2.1 AdaBoost Definitions

- T: The number of iterations used to train AdaBoost.
- N: The number of training samples.
- $S = \{(x^{(1)}, y^{(1)}), \cdots, (x^{(N)}, y^{(N)})\}$: The training samples with binary labels ($y^{(i)} \in \{-1, +1\}$).
- $\omega_t^{(i)}$: The weight assigned to training example i at time t. Note that $\sum_i \omega_t^{(i)} = 1$.
- h_t: The weak learner constructed at time t (a function $X \to \{-1, +1\}$).
- ϵ_t: The weighted (by ω_t) error of h_t.
- $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$: The normalization factor for the distribution update at time t.
- $\alpha_t = \frac{1}{2}\ln((1-\epsilon_t)/\epsilon_t)$: The weight assigned to the learner h_t in the composite hypothesis.
- $H_t(x) = \left(\sum_{t'=1}^{t} \alpha_{t'} h_{t'}(x)\right) / \left(\sum_{t'=1}^{t} \alpha_{t'}\right)$: The majority vote of the weak learners, rescaled based on the total weights.
- $g_t(x) = \text{sign}(H_t(x))$: The voting classifier decision function.

1.2.2 AdaBoost Weighting

AdaBoost relies on building an ensemble of weak learners, assigning them weights based on their errors during training.

1. Assume we are in the binary classification setting. What happens to the weight $\alpha_t = \dfrac{1}{2}\ln\left(\dfrac{1-\epsilon_t}{\epsilon_t}\right)$ of classifier h_t if its error $\epsilon_t > 0.5$? Why is this useful?

   Note that if we can find weak learners h_t with $\epsilon_t < 0.5$ for all t, training error will decrease exponentially fast in the total number of iterations T.

2. AdaBoost also assigns weights $\omega_t^{(i)}$ for each data point. Explain in broad terms how the weights assigned to examples get updated in each iteration.

1.2.3 The Margin

In the following question, we will examine the generalization error of AdaBoost using a concept known as the *classification margin*.

For a binary classification task, assume that we use a probabilistic classifier that provides a probability distribution over the possible labels (i.e. $p(y|x)$ for $y \in \{+1, -1\}$). The classifier output is the label with highest probability. We define the *classification margin* for an input as the signed difference between the probability assigned to the correct label and the incorrect label $p_{correct} - p_{incorrect}$, which takes on values in the range $[-1, 1]$.

1. Let $\text{margin}_t(x, y)$ represent the margin for our AdaBoost classifier at iteration t on the sample (x, y). Write a single inequality in terms of $\text{margin}_t(x, y)$ that is true if and only if the classifier makes a mistake on the input (x, y) (i.e., provide a bound on the margin in the case the classifier is incorrect). Assume the classifier makes a mistake on ties.

2. For a given input and label $(x^{(i)}, y^{(i)})$, write $\text{margin}_t(x^{(i)}, y^{(i)})$ in terms of $x^{(i)}, y^{(i)}$, and f_t.

1.2.4 Weak Learners

We always talk using AdaBoost with "weak" learners; why can't we ensemble together "stronger" learners? Let's take a look at bounds on the test error of AdaBoost, fixing the number of samples N and number of training iterations T, but allowing variation in the hypothesis class of weak learners $\mathcal{H}$.

Let d be the VC-dimension of the hypothesis class. Consider the following bounds on the error of the ensemble H_T with respect to d:

$$\text{Bound 1 (PAC Learning)}: \quad \text{True Error} \leq \text{Train Error} + O\left(\sqrt{T \log T}\sqrt{d}\sqrt{\frac{\log N}{N}}\right)$$

$$\text{Bound 2 (Margin Analysis)}: \quad \text{True Error} \leq \hat{P}_S\left[\text{margin}_T \leq \theta\right] + O\left(\frac{1}{\theta}\sqrt{d}\sqrt{\frac{\log^2 N}{N}}\right)$$

1. What happens to our bounds on true error if we increase the VC dimension of the weak learner hypothesis space?

2. What concept does this connection between classifier complexity and error relate to?