Likelihood Machine Learning Seminar II

Lecture 6: Random Forest & Boosting Tree

Likelihood Lab

XingYu Fu

1. Introduction to Ensemble Learning

Ensemble learning is a inference technique that combines multiple outputs from different predictive machine learning algorithms to make a better comprehensive prediction. Generally, ensemble learning method can lower the predictive variance and increase the predictive accuracy compared to single machine learning algorithm. There are mainly three categories of ensemble learning methods:

- **Bagging Framework:** Sample multiple subsets from the original training data with replacement (*Bootstrapping*) and train machine learning models on each subset *in parallel*. The final prediction is the average or majority voting of each sub-model's output. Below is the pseudo-code of bagging given by ZhiHua Zhou.

```
Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
       Base learning algorithm $\mathcal{L}$;
       Number of learning rounds $T$.
Process:
  for $t = 1, \cdots, T$:
      $\mathcal{D}_t = Bootstrap(\mathcal{D})$;     % Generate a bootstrap sample from $\mathcal{D}$
      $h_t = \mathcal{L}(\mathcal{D}_t)$     % Train a base learner $h_t$ from the bootstrap sample
  end.
Output: $H(\boldsymbol{x}) = \text{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{T} \mathbb{1}(y = h_t(\boldsymbol{x}))$     % the value of $\mathbb{1}(a)$ is 1 if $a$ is $true$ and 0 otherwise
```

- **Boosting Framework:** Train a number of weak learning algorithms *sequentially*, each of which pays special attention to the *mistakes* that are made by previous learners. The final prediction is a weighted average of all outputs from all kept weak learners, where the weights are directly related to the performances of the models. The below pseudo-code is the famous boosting algorithm AdaBoosting.

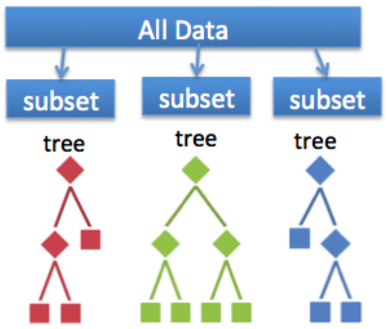
```
Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
       Base learning algorithm $\mathcal{L}$;
       Number of learning rounds $T$.
Process:
  $D_1(i) = 1/m$.      % Initialize the weight distribution
  for $t = 1, \cdots, T$:
      $h_t = \mathcal{L}(\mathcal{D}, D_t)$;      % Train a base learner $h_t$ from $\mathcal{D}$ using distribution $D_t$
      $\epsilon_t = \Pr_{i \sim D_i}[h_t(\boldsymbol{x}_i \neq y_i)]$;      % Measure the error of $h_t$
      $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$;      % Determine the weight of $h_t$
      $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) \neq y_i \end{cases}$
              $= \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$      % Update the distribution, where $Z_t$ is a normalization
                                                % factor which enables $D_{t+1}$ to be a distribution
  end.
Output: $H(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x})) = \text{sign} \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$
```

- **Stacking Framework:** Train a number of first-level learners from the training data by applying different learning algorithms and then combine the predicted results from them by a second-level learners, which is called as *meta-learner*.

```
Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
       First-level learning algorithms $\mathcal{L}_1, \cdots, \mathcal{L}_T$;
       Second-level learning algorithm $\mathcal{L}$.
Process:
  for $t = 1, \cdots, T$:
      $h_t = \mathcal{L}_t(\mathcal{D})$      % Train a first-level individual learner $h_t$ by applying the first-level
  end;                          % learning algorithm $\mathcal{L}_t$ to the original data set $\mathcal{D}$
  $\mathcal{D}' = \emptyset$;      % Generate a new data set
  for $i = 1, \cdots, m$:
      for $t = 1, \cdots, T$:
          $z_{it} = h_t(\boldsymbol{x}_i)$      % Use $h_t$ to classify the training example $\boldsymbol{x}_i$
      end;
      $\mathcal{D}' = \mathcal{D}' \cup \{((z_{i1}, z_{i2}, \cdots, z_{iT}), y_i)\}$
  end;
  $h' = \mathcal{L}(\mathcal{D}')$.      % Train the second-level learner $h'$ by applying the second-level
                          % learning algorithm $\mathcal{L}$ to the new data set $\mathcal{D}'$
Output: $H(\boldsymbol{x}) = h'(h_1(\boldsymbol{x}), \cdots, h_T(\boldsymbol{x}))$
```

2. Random Forest

Now we see a more concrete examples: Random Forest. It is a bagging ensemble of decision trees, each of which focuses on different features to avoid trees that are too similar. The technique of selecting different features for different tress is also called as *column sampling*.



3. Boosting Tree

Apply the boosting framework on decision tree, you obtain the so-called boosting tree algorithm.