Bagging and Boosting: Brief Introduction

Jia Li

Department of Statistics
The Pennsylvania State University

Email: jiali@stat.psu.edu
http://www.stat.psu.edu/~jiali
Overview

- Bagging and boosting are meta-algorithms that pool decisions from multiple classifiers.
- Much information can be found on Wikipedia.
Overview on Bagging

- Invented by Leo Breiman: Bootstrap aggregating.
- Majority vote from classifiers trained on bootstrap samples of the training data.
Bagging

- Generate $B$ bootstrap samples of the training data: random sampling with replacement.
- Train a classifier or a regression function using each bootstrap sample.
- For classification: majority vote on the classification results.
- For regression: average on the predicted values.
- Reduces variation.
- Improves performance for unstable classifiers which vary significantly with small changes in the data set, e.g., CART.
- Found to improve CART a lot, but not the nearest neighbor classifier.
Overview on Boosting

- Iteratively learning weak classifiers
- Final result is the weighted sum of the results of weak classifiers.
- Many different kinds of boosting algorithms: Adaboost (Adaptive boosting) by Y. Freund and R. Schapire is the first.
- Examples of other boosting algorithms:
  - LPBoost: Linear Programming Boosting is a margin-maximizing classification algorithm with boosting.
  - BrownBoost: increase robustness against noisy datasets. Discard points that are repeatedly misclassified.

Jia Li http://www.stat.psu.edu/~jiali
Adaboost for Binary Classification

1. Training data: \((x_i, y_i), i = 1, \ldots, n, x_i \in \mathcal{X}, y_i \in \mathcal{Y} = \{-1, 1\}\).
2. Let \(w_{1,i} = \frac{1}{n}, i = 1, \ldots, n\).
3. For \(t = 1, \ldots, T\):
   3.1 Learn classifier \(h_t : \mathcal{X} \rightarrow \mathcal{Y}\) from a set of weak classifiers called hypotheses, \(\mathcal{H} = \{h(\cdot, \omega) | \omega \in \Omega\}\), that minimizes the error rate with respect to distribution \(w_{t,i}\) over \(x_i\)'s.
   3.2 Let \(r_t = \sum_{i=1}^{n} w_{t,i} I(y_i \neq h_t(x_i))\). If \(r_t > 0.5\), stop.
   3.3 Choose \(\alpha_t \in \mathbb{R}\). Usually set \(\alpha_t = \frac{1}{2} \log \frac{1-r_t}{r_t}\).
   3.4 Update \(w_{t+1,i} = \frac{w_{t,i} e^{-\alpha_t y_i h_t(x_i)}}{Z_t}\), where \(Z_t\) is a normalization factor to ensure \(\sum_i w_{t+1,i} = 1\).
4. Output the final classifier: \(f(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)\).

Note: the update of \(w_{t,i}\) implies incorrectly classified points receive increased weights in the next round of learning.

Jia Li  http://www.stat.psu.edu/~jiali