Two Views of Boosting: Margin vs. Convex Loss Minimization

Liwei Wang

Peking University
Background

- **Learning and Classification:**
  - Training examples
    \[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\]
    \[x_i \in X, y_i \in Y\]
    i.i.d. from an underlying joint distribution \(P\)
  - Classifier: \(C : X \rightarrow Y\)
  - Generalization Error: \(P(C(x) \neq y)\)
The Boosting algorithm

Input: \( S = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \)
where \( x_i \in X, y_i \in \{-1, 1\} \).
Initialization: \( D_1(i) = 1/n. \)
for \( t = 1 \) to \( T \) do

1. Train base learner using distribution \( D_t. \)
2. Get base classifier \( h_t : X \rightarrow \{-1, 1\}. \)
3. Choose \( \alpha_t. \)
4. Update:

\[
D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},
\]

where \( Z_t \) is a normalization factor chosen so that \( D_{t+1} \) will be a distribution.
end
Output: The final Classifier

\[
H(x) = \text{sgn} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right).
\]
Background

- Empirical observation:
  - AdaBoost often resists to overfitting:
    - The test error of the combined classifier usually keeps decreasing as its size becomes very large, and even after the training error is zero, which seems contradicts the Occam’s razor!
We need a theoretical explanation of the Boosting algorithm:

- Understanding the “mysteries”.
- Develop more efficient algorithms.
A complete theoretical explanation should answer two questions:

- Why AdaBoost often has good performance?
- Why AdaBoost is often (though not always) immune to overfitting?
Outline

- The Margin Explanation
- The Convex Loss Explanation
- Margin vs. Convex Loss
- Open Problems
The Margin Explanation
The Concept of Margin

- Margins in Boosting:
  - The combined (voting) classifier produced by the ensemble learning algorithms could be written as:

\[
 f(x) = \sum_{i} \alpha_i h_i(x), \quad i \in \{1, \ldots, m\}, \quad i \in \{0, \ldots, m\}.
\]
The Concept of Margin

• For binary classification, $y \in \{1, -1\}$. The quantity $y f(x)$ is called the margin of the example $(x, y)$ with respect to the classifier $f$.

• Margin is a confidence measure (like in SVM).

• The minimum margin is the smallest margin over the set of training examples.
• Margin distribution:
Margin Theory

- Margin theory is essentially upper bounds on the generalization error of the voting classifier, in terms of various margin notions.
The Margin Distribution Bound

- **Theorem 1 (Schapire et al. 1998):**
  - For any $\theta$, with probability at least $1 - \delta$, over the random choice of the training set $S$ of $n$ examples, every voting classifier satisfies the following bound:

$$P_D \ yf(x) - 0 \inf_{(0,1]} P_S \ yf(x) \leq O \left( \frac{1}{\sqrt{n}} \log n \log |H| \log \frac{1}{\delta} \right)$$

where $H$ is the set which the base classifiers are chosen from.
Schapire et al. also demonstrated theoretically and empirically that AdaBoost can generate good margin distribution.

The margin distribution keeps improving even after the training error is zero. This accounts for AdaBoost’s resistance to overfitting.
Breiman’s doubt and the Arc-gv algorithm:

- Arc-gv provably generate the largest possible minimum margin among all boosting type algorithms.
Breiman’s Doubt

The minimum margin bound (Breiman 1999):

\[ P_D, yf(x) \leq 0 \quad R \log 2n \quad \log \frac{1}{R} \quad 1 \quad \frac{1}{n} \log |H| \quad . \]

where

\[ R \quad \frac{32 \log 2 |H|}{n^2} \]

and \( \delta \) is the minimum margin.
Breiman’s Doubt

- Breiman’s argument:
  - The minimum margin bound is sharper than the margin distribution bound.

\[
O \left( \frac{\log n}{n} \right) \quad \text{vs.} \quad O\left(\frac{\sqrt{\log n}}{n}\right)
\]

If the bound of Schapire et al. implies that the margin distribution is the key to the generalization error, his bound implies more strongly that the minimum margin governs the generalization error.
Breiman’s Doubt

- Breiman conducted experiments, and found that arc-gv performs **consistently worse** than AdaBoost although it always generates larger minimum margins!
- Arc-gv even generates uniformly better margin distribution than AdaBoost.
- Breiman concluded that neither the margin distribution nor the minimum margin is the right explanation!
**Recent Discovery**

- An important discovery (Reyzin and Schapire 2006):
  - In the margin bounds, the generalization error depends not only on the margin, but also the complexity of the set of base classifiers.

\[
P_D \inf_{yf(x)\geq\delta} P_S \inf_{yf(x)\leq\delta} O \left( \frac{1}{\sqrt{n}} \log n \log |H| \right) \frac{1}{2} \log \frac{1}{\theta}^{1/2}
\]

- To study how margin affects the generalization, one has to keep other factors fixed.
Recent Discovery

• Breiman’s experiment:
  ▪ Base classifiers: Using decision trees of a fixed number of leaves.

• Reyzin and Schapire’s discovery:
  ▪ Trees generated by arc-gv are much deeper than those generated by AdaBoost!
  ▪ Deeper trees are more complex even though the number of leaves are the same!
  ▪ Breiman’s experiment is not a fair comparison.
Recent Discovery

- A fair comparison:
  - Base classifier: decision stump.
  - Results:
    - AdaBoost has better performance than arc-gv.
    - Arc-gv has larger minimum margins than AdaBoost.
    - The margin distribution generated by AdaBoost is “better” than arc-gv.
Two Problems Left

- Has Breiman’s doubt been fully answered?
  - Arc-gv generates larger minimum margin yet has worse performance. Contradict to the (sharper) minimum margin bound!
  - What does it mean a “better” margin distribution?
The EMargin Explanation

- Main results of EMargin:
  - A bound for the generalization error of voting classifiers in terms of a new margin notion—Equilibrium Margin (Emargin). This bound is uniformly sharper than the minimum margin bound.
  - We show that a large Emargin implies a smaller generalization error.
- Bernoulli Relative Entropy:

\[ D(q \parallel p) = q \log \frac{q}{p} - (1 - q) \log \frac{1}{p}, \quad 0 < p, q < 1. \]

- For fixed \( q \), \( D \) is a monotone increasing function of \( p \) for \( q < p < 1 \).

- Inverse Relative Entropy Function:

\[ D^1(q,u) : \quad D(q \parallel D^1(q,u)) \quad u. \quad u \quad 0. \]
The Emargin Bound Theorem:

\[ P_D \ yf(x) = \min_{q \in \{0, \frac{1}{n}, \frac{2}{n}, \ldots, 1\}} \frac{\log |H|}{n} D^1 q, u(\cdot) \]

where \( P_S \ yf(x) = q, \)

\[ u(\cdot) = \frac{1}{n} + \frac{8}{2} \log \frac{2n^2}{\log |H|} \log |H| \log |H| \log n. \]

Let \( q^* \) and \( * \) be the optimal \( q, \) in the Emargin bound

\[ P_D \ yf(x) = \min_{q^* \in \{0, \frac{1}{n}, \frac{2}{n}, \ldots, 1\}} \frac{\log |H|}{n} D^1 q^*, u(\cdot). \]

* is referred to as Emargin.
Explanation of Emargin bound:

- The Emargin bound has a similar flavor to the margin distribution bound. The Emargin and Emargin error depend, in a complicated way on the whole margin distribution.

- The minimum margin is only a special case when the optimal $q^*$ is zero.
Theorem:
- The E-margain bound is uniformly sharper than the minimum margin bound.

Minimum margin is not crucial for the generalization error. Arc-gv does not necessarily have better performance than AdaBoost.

\[
P_D(yf(x) = 0) \leq \frac{\log |H|}{n} D^1 \mathbb{E}[q^*, u(\mathbb{x})].
\]

The E-margain bound implies that it is the E-margin and the E-margin error affect the performance of the classifier.
How do Emargin and Emargin error affect the generalization error?

The Comparison Theorem:
- For two voting classifiers \( f_1, f_2 \), if \( f_1 \) has a larger Emargin and a smaller Emargin error than \( f_2 \), then the Emargin bound of \( f_1 \) is smaller than \( f_2 \).
$P_s(yf(x))$
Further explanation of the Emargin bound:

- By using simple upper bounds of the inverse relative entropy function $D^1(q,u)$, we can recover previous bounds and obtain new bound in simpler forms.

$$\inf_{q} D^1 q, u \hat{(}q) \quad D^1 0, u \hat{(}0) \quad u \hat{(}0) \quad \longrightarrow \text{minimum margin bound}$$

$$\inf_{q} D^1 q, u \hat{(}q) \quad \inf_{q} q \quad \frac{u \hat{(}q)}{2}^{1/2} \quad \longrightarrow \text{margin distribution bound}$$

$$\inf_{q} D^1 q, u \hat{(}q) \quad \inf_{q Cu \hat{(}q)} D^1 q, u \hat{(}q) \quad \inf_{q Cu \hat{(}q)} Cu \hat{(}q) \quad \longrightarrow \text{a new } O\left(\frac{\log n}{n}\right) \text{ bound for the nonzero error case.}$$
Experiments

- Setting:
  - UCI and USPS datasets.
  - Five-fold CV.
  - Binary classification.
  - Finite base classifiers.
  - Comparison of AdaBoost and Arc-gv on their EMargin, EMargin error, test error and minimum margin.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Emargin</th>
<th>Emargin Error</th>
<th>Test Error</th>
<th>Minimum margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast</td>
<td>AdaBoost</td>
<td>0.313</td>
<td>0.803</td>
<td>0.052</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.281</td>
<td>0.909</td>
<td>0.057</td>
<td>0.008</td>
</tr>
<tr>
<td>Diabetes</td>
<td>AdaBoost</td>
<td>0.110</td>
<td>0.748</td>
<td>0.255</td>
<td>-0.064</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.049</td>
<td>0.759</td>
<td>0.256</td>
<td>-0.017</td>
</tr>
<tr>
<td>German</td>
<td>AdaBoost</td>
<td>0.157</td>
<td>0.824</td>
<td>0.258</td>
<td>-0.118</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.034</td>
<td>0.780</td>
<td>0.261</td>
<td>-0.026</td>
</tr>
<tr>
<td>Image</td>
<td>AdaBoost</td>
<td>0.196</td>
<td>0.610</td>
<td>0.023</td>
<td>-0.009</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.195</td>
<td>0.705</td>
<td>0.021</td>
<td>-0.003</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>AdaBoost</td>
<td>0.323</td>
<td>0.800</td>
<td>0.100</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.131</td>
<td>0.577</td>
<td>0.106</td>
<td>0.061</td>
</tr>
<tr>
<td>Letter</td>
<td>AdaBoost</td>
<td>0.078</td>
<td>0.645</td>
<td>0.174</td>
<td>-0.165</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.063</td>
<td>0.958</td>
<td>0.178</td>
<td>-0.034</td>
</tr>
<tr>
<td>Satimage</td>
<td>AdaBoost</td>
<td>0.133</td>
<td>0.521</td>
<td>0.053</td>
<td>-0.054</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.133</td>
<td>0.956</td>
<td>0.057</td>
<td>-0.019</td>
</tr>
<tr>
<td>USPS</td>
<td>AdaBoost</td>
<td>0.108</td>
<td>0.972</td>
<td>0.450</td>
<td>-0.142</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.053</td>
<td>0.990</td>
<td>0.460</td>
<td>-0.024</td>
</tr>
<tr>
<td>Vehicle</td>
<td>AdaBoost</td>
<td>0.105</td>
<td>0.698</td>
<td>0.201</td>
<td>-0.024</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.063</td>
<td>0.720</td>
<td>0.205</td>
<td>-0.009</td>
</tr>
<tr>
<td>Wdbc</td>
<td>AdaBoost</td>
<td>0.350</td>
<td>0.581</td>
<td>0.035</td>
<td>-0.130</td>
</tr>
<tr>
<td></td>
<td>arc-gv</td>
<td>0.350</td>
<td>0.710</td>
<td>0.035</td>
<td>-0.100</td>
</tr>
</tbody>
</table>
Experiments

- Conclusion from the experiments:
  - Usually AdaBoost has a larger EMargin and a smaller EMargin error than arc-gv. This accounts for AdaBoost’s superior performances.
Convex Loss Minimization
Breiman discovered that AdaBoost was a down-the-gradient method for minimizing the exponential loss.

Exp loss: \[ \frac{1}{n} \exp \sum_{i=1}^{n} y_i f(x_i) \quad \frac{1}{n} \exp \sum_{i=1}^{n} y_i^T h_t(x_i) \]

0-1 loss: \[ \frac{1}{n} \sum_{i=1}^{n} I(y_i f(x_i)) \quad \frac{1}{n} \sum_{i=1}^{n} I(y_i^T h_t(x_i)) \]
Convex Loss Minimization

- A natural question:
  - To what extent solving the approximated convex surrogate minimization is equivalent to minimizing the generalization error?
Convex Loss Minimization

- The statistical consequences of minimizing a surrogate:
  - Bayes Consistency:
    - Minimizing the convex loss (boosting) is NOT consistent.
    - With regularization, boosting is consistent:
      - Early stopping
      - L1 regularization
  - Rate of Convergence (dimension independent):
    \[ n^{1/4} \sim n^{1/2} \]
Convex Loss Minimization

- Large margin vs. convex loss minimization:
  - They are complementary explanations.
  - Convex loss minimization:
    - Asymptotic results, compare to the Bayes risk.
    - Depends on precise algorithms
  - Margin:
    - Nonasymptotic uniform bounds, gives confidence interval of the generalization error.
    - Algorithm independent.
Convex Loss Minimization

- Limitation of the convex loss minimization:
  - All based on an important assumption:
    - The linear span of the base classifiers is dense in the space of all measurable functions. Or at least the global minimizer of the convex loss is contained in the linear span.
  - If the base classifiers are decision stumps or other simple models, this assumption does not hold.
Convex Loss Minimization

- What is the consequence of boosting in the misspecified setting:
  - Bayes consistent is impossible.
  - Consistency to the best classifier in the model?
  - Empirically, boosting decision stumps often yields good performance.
Convex Loss Minimization

• A surprising result (Long & Servedio 2008):

There are learning problems such that

- Bayes error is slightly larger than zero.
- The Bayes classifier is within the model class.
- Minimizing the convex loss returns in a classifier whose performance is the same as random guess, even if there are infinitely many training examples!
- Early stopping and L1 regularization do not help!
Convex Loss Minimization

- Convex loss minimization can not explain these results.

- Margin theory can predict the performance by giving the upper bound of the generalization error.
Summary and Future Work
The EMargin bound is an answer to Breiman’s doubt of the margin explanation.

Margin and convex loss minimization are complementary explanations of boosting.

Is it possible to optimize the margin distribution (EMargin)?
Thanks