MLRG session on ”Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers” by Wyner et. al (2017)

Alex Grabanski

2/6/2017

Generalization Error of AdaBoost

- AdaBoost generalizes surprisingly well in practice, despite the fact that it fits training data perfectly.

Generalization Error of AdaBoost

- AdaBoost generalizes surprisingly well in practice, despite the fact that it fits training data perfectly.
- Somehow seems to reduce both bias *and* variance

Generalization Error of AdaBoost

- AdaBoost generalizes surprisingly well in practice, despite the fact that it fits training data perfectly.
- Somehow seems to reduce both bias *and* variance
- Competing explanations for this behavior existed before this paper (margins theory, statistical theory)

Generalization Error of AdaBoost

- AdaBoost generalizes surprisingly well in practice, despite the fact that it fits training data perfectly.
- Somehow seems to reduce both bias *and* variance
- Competing explanations for this behavior existed before this paper (margins theory, statistical theory)
- Yet, they fail to describe the generalization behavior in various ways

Generalization Error of AdaBoost

- AdaBoost generalizes surprisingly well in practice, despite the fact that it fits training data perfectly.
- Somehow seems to reduce both bias *and* variance
- Competing explanations for this behavior existed before this paper (margins theory, statistical theory)
- Yet, they fail to describe the generalization behavior in various ways
- This paper: Yes, AdaBoost overfits, but the overfitting is very localized.

Generalization Error of AdaBoost

- AdaBoost generalizes surprisingly well in practice, despite the fact that it fits training data perfectly.
- Somehow seems to reduce both bias *and* variance
- Competing explanations for this behavior existed before this paper (margins theory, statistical theory)
- Yet, they fail to describe the generalization behavior in various ways
- This paper: Yes, AdaBoost overfits, but the overfitting is very localized.
- Advanced ideas of ”interpolating classifiers” and ”spiked-smooth decision boundaries”

Margin Theory

- Idea: AdaBoost increases the *confidence* in its predictions over time (boosting rounds)

$$m(x) = y * F(x) = (\sum_{i:f_i(x)=y} \alpha_i) - (\sum_{i:f_i(x)\neq y} \alpha_i)$$

Margin Theory

- Idea: AdaBoost increases the *confidence* in its predictions over time (boosting rounds)
- Measured by the *margins of prediction* on particular examples (below)

$$m(x) = y * F(x) = \Big(\sum_{i:f_i(x)=y} \alpha_i\Big) - \Big(\sum_{i:f_i(x)\neq y} \alpha_i\Big)$$

Margin Theory

- Idea: AdaBoost increases the *confidence* in its predictions over time (boosting rounds)
- Measured by the *margins of prediction* on particular examples (below)
- Higher minimum margin? Can examine fewer classifiers (those with the greatest weight) in the ensemble to get a result.

$$m(x) = y * F(x) = \Big(\sum_{i:f_i(x)=y} \alpha_i\Big) - \Big(\sum_{i:f_i(x)\neq y} \alpha_i\Big)$$

Margin Theory

- Idea: AdaBoost increases the *confidence* in its predictions over time (boosting rounds)
- Measured by the *margins of prediction* on particular examples (below)
- Higher minimum margin? Can examine fewer classifiers (those with the greatest weight) in the ensemble to get a result.
- Direct generalization bounds in terms of the margins (and VC dimension, and sample size) exist, but are not tight.

$$m(x) = y * F(x) = \Big(\sum_{i:f_i(x)=y} \alpha_i\Big) - \Big(\sum_{i:f_i(x)\neq y} \alpha_i\Big)$$

Problems with the Margins Theory

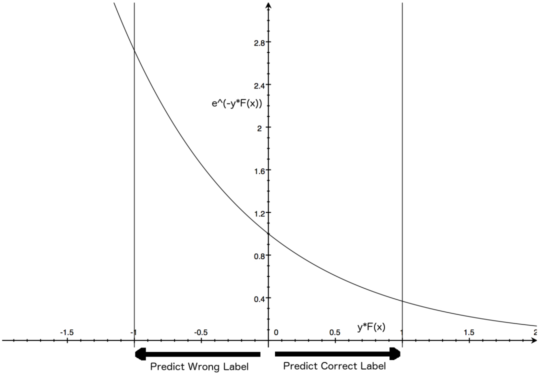
- Explicit maximization of minimum margins (LPBoost, arc-gv) does *not* yield better generalization than AdaBoost in practice!

Problems with the Margins Theory

- Explicit maximization of minimum margins (LPBoost, arc-gv) does *not* yield better generalization than AdaBoost in practice!
- This is *despite* those algorithms achieving tighter generalization error bounds!

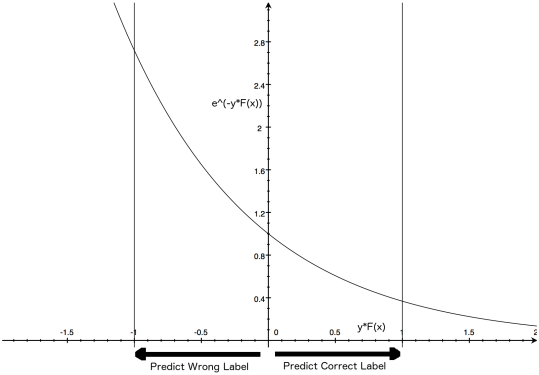
Statistical Theory

- AdaBoost is stagewise minimization of the exponential loss



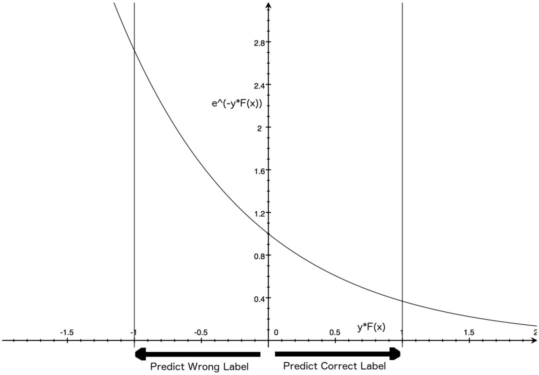
Statistical Theory

- AdaBoost is stagewise minimization of the exponential loss
- Exponential loss is a convex surrogate for optimizing the (intractable) 0-1 loss



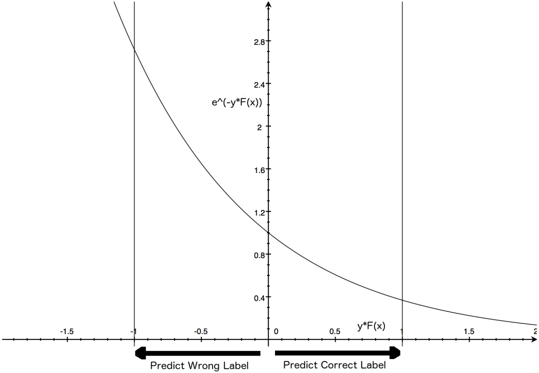
Statistical Theory

- AdaBoost is stagewise minimization of the exponential loss
- Exponential loss is a convex surrogate for optimizing the (intractable) 0-1 loss
- Good generalization in AdaBoost may be achieved by stopping after some finite number of boosting rounds.



Statistical Theory

- AdaBoost is stagewise minimization of the exponential loss
- Exponential loss is a convex surrogate for optimizing the (intractable) 0-1 loss
- Good generalization in AdaBoost may be achieved by stopping after some finite number of boosting rounds.
- Stopping early acts as a form of regularization.



Problems with the Statistical Theory

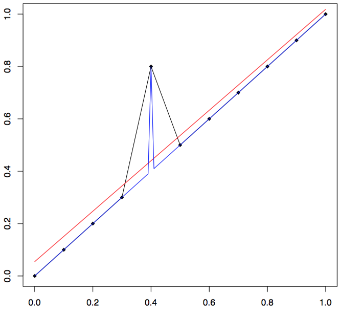
- Cases exist (Evidence Contrary to the Statistical View of Boosting, Mease et. al.) where AdaBoost does not overfit, *even as the number of boosting rounds grows very large*

Problems with the Statistical Theory

- Cases exist (Evidence Contrary to the Statistical View of Boosting, Mease et. al.) where AdaBoost does not overfit, *even as the number of boosting rounds grows very large*
- β-boosting, proposed in (On Boosting and The Exponential Loss, Wyner) does *not* reduce the exponential loss, but is very similar to AdaBoost, and has very similar generalization behavior.

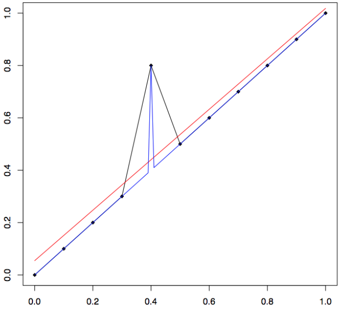
This paper: Interpolating Classifiers

- AdaBoost fits the training data perfectly – overfitting?



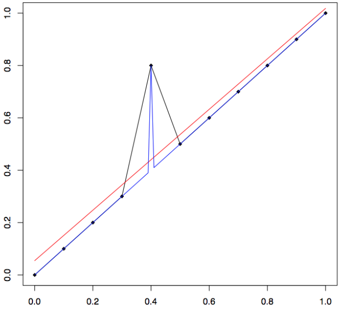
This paper: Interpolating Classifiers

- AdaBoost fits the training data perfectly – overfitting?



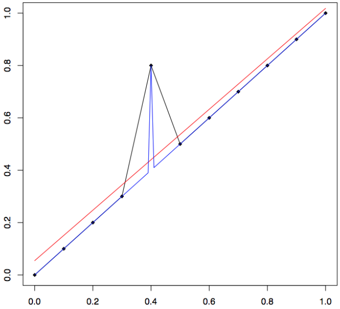
This paper: Interpolating Classifiers

- AdaBoost fits the training data perfectly – overfitting?
- Black and blue ”interpolate” the data (fit the training set perfectly), but blue only overfits *locally*



This paper: Interpolating Classifiers

- AdaBoost fits the training data perfectly – overfitting?
- Black and blue ”interpolate” the data (fit the training set perfectly), but blue only overfits *locally*
- **Big Idea:** Aggregating many different interpolating classifiers *smooths* the non-noisy part of the decision boundary, while keeping *spikes* around the noisy data points. Call this a *Spiked-Smooth* decision boundary.



Similarity of AdaBoost (over Deep Decision Trees) with Random Forests

- Random Forests employ randomization in examples, attribute subsets for splits to generate an ensemble

Algorithm 2: Random Forests Hastie et al. (2009)

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{X}^*$ of size N from the training data
   (b) Grow a decision tree T_b to the data $\mathbf{X}^*$ by doing the following recursively until the minimum node size n_{min} is reached:
      i. Select m of the p variables
      ii. Pick the best variable/split-point from the m variables and partition
2. Output the ensemble $\{T_b\}_b^B$

Let $\hat{C}_b(\mathbf{x}^*)$ be predicted class of tree T_b. Then $\hat{C}_{rf}^B(\mathbf{x}^*) = \text{majority vote}\{\hat{C}_b(\mathbf{x}^*)\}_1^B$.

Similarity of AdaBoost (over Deep Decision Trees) with Random Forests

- Random Forests employ randomization in examples, attribute subsets for splits to generate an ensemble
- Also interpolate the data (with the right n_{min})

Algorithm 2: Random Forests Hastie et al. (2009)

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{X}^*$ of size N from the training data

   (b) Grow a decision tree T_b to the data $\mathbf{X}^*$ by doing the following recursively until the minimum node size n_{min} is reached:

      i. Select m of the p variables

      ii. Pick the best variable/split-point from the m variables and partition

2. Output the ensemble $\{T_b\}_b^B$

Let $\hat{C}_b(\mathbf{x}^*)$ be predicted class of tree T_b. Then $\hat{C}_{rf}^B(\mathbf{x}^*) = \text{majority vote}\{\hat{C}_b(\mathbf{x}^*)\}_1^B$.

Similarity of AdaBoost (over Deep Decision Trees) with Random Forests

- Random Forests employ randomization in examples, attribute subsets for splits to generate an ensemble
- Also interpolate the data (with the right n_{min})
- Example weights approach an invariant distribution in AdaBoost (Random Forests, by Brieman)

Algorithm 2: Random Forests Hastie et al. (2009)

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{X}^*$ of size N from the training data
   (b) Grow a decision tree T_b to the data $\mathbf{X}^*$ by doing the following recursively until the minimum node size n_{min} is reached:
      i. Select m of the p variables
      ii. Pick the best variable/split-point from the m variables and partition
2. Output the ensemble $\{T_b\}_b^B$

Let $\hat{C}_b(\mathbf{x}^*)$ be predicted class of tree T_b. Then $\hat{C}_{rf}^B(\mathbf{x}^*) = \text{majority vote}\{\hat{C}_b(\mathbf{x}^*)\}_1^B$.

Similarity of AdaBoost (over Deep Decision Trees) with Random Forests

- Random Forests employ randomization in examples, attribute subsets for splits to generate an ensemble
- Also interpolate the data (with the right n_{min})
- Example weights approach an invariant distribution in AdaBoost (Random Forests, by Brieman)
- Subsequent classifiers may interpolate, but they also *smooth* the decision boundary.

Algorithm 2: Random Forests Hastie et al. (2009)

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{X}^*$ of size N from the training data
   (b) Grow a decision tree T_b to the data $\mathbf{X}^*$ by doing the following recursively until the minimum node size n_{min} is reached:
      i. Select m of the p variables
      ii. Pick the best variable/split-point from the m variables and partition
2. Output the ensemble $\{T_b\}_b^B$

Let $\hat{C}_b(\mathbf{x}^*)$ be predicted class of tree T_b. Then $\hat{C}_{rf}^B(\mathbf{x}^*) = \text{majority vote}\{\hat{C}_b(\mathbf{x}^*)\}_1^B$.

Boosting round "slabs"

- AdaBoost is AdaBoost with AdaBoost (fixed number of iterations) as a base classifier

$$AdaBoost(B, +\infty) \simeq AdaBoost(AdaBoost(B, L), +\infty)$$

Boosting round "slabs"

- AdaBoost is AdaBoost with AdaBoost (fixed number of iterations) as a base classifier
- Enough iterations, and AdaBoost fits the training data perfectly

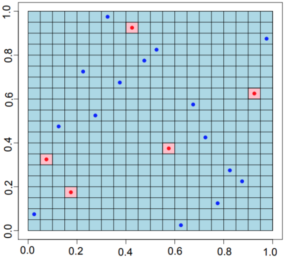
$$AdaBoost(B, +\infty) \simeq AdaBoost(AdaBoost(B, L), +\infty)$$

Boosting round "slabs"

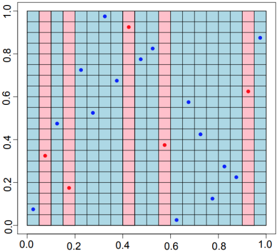
- AdaBoost is AdaBoost with AdaBoost (fixed number of iterations) as a base classifier
- Enough iterations, and AdaBoost fits the training data perfectly
- This only happens, though, if training error rates of base classifiers bounded away from $\frac{1}{2}$

$$AdaBoost(B, +\infty) \simeq AdaBoost(AdaBoost(B, L), +\infty)$$

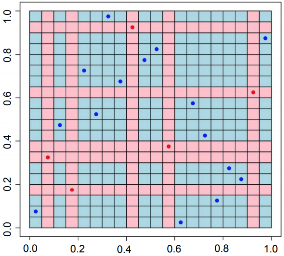
Intuition about base classifiers (from paper)



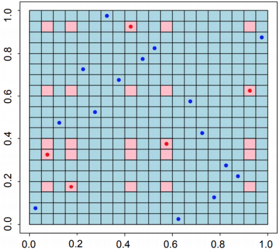
(a) Hypothetical Classifier 1



(b) Hypothetical Classifier 2



(c) Hypothetical Classifier 3



(d) Hypothetical Classifier 4

The paper's synthetic experiments

- Generated data randomly in 2d, but with some noisy data points thrown in.

The paper's synthetic experiments

- Generated data randomly in 2d, but with some noisy data points thrown in.
- Chosen so that Bayes optimal classifier is to classify all points negative (blue)

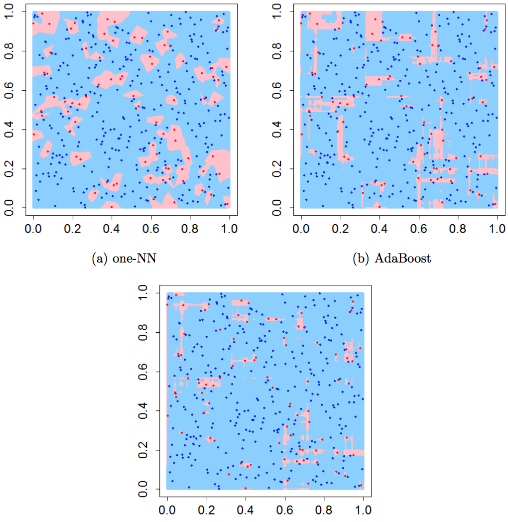
The paper's synthetic experiments

- Generated data randomly in 2d, but with some noisy data points thrown in.
- Chosen so that Bayes optimal classifier is to classify all points negative (blue)
- As conjectured, overfitting seems to be localized, just like with random forests.

The paper's synthetic experiments

- Generated data randomly in 2d, but with some noisy data points thrown in.
- Chosen so that Bayes optimal classifier is to classify all points negative (blue)
- As conjectured, overfitting seems to be localized, just like with random forests.
- Also did other experiments in higher dimensions, similar results

Comparison with Random Forests/1NN (from paper)



(a) one-NN

(b) AdaBoost

(c) Random Forests

Questions/Discussion