Lecture 13: Boosting. Computational Learning Theory (COLT)

- Boosting
- Estimating the true error of a hypothesis
- PAC learning
- Other COLT models

Recall: Bias and variance

- For regression problems, the expected error can be decomposed as:

$$\text{Bias}^2 + \text{Variance} + \text{Noise}$$

- Bias is typically caused by the hypothesis class being too simple, and hence not able to represent the true function (*underfitting*)
- Variance is typically caused by the hypothesis class being too large (*overfitting*)
- There is often a trade-off between bias and variance

Measuring bias and variance in practice

- Recall that bias and variance are both defined as expectations:

$$Bias(\mathbf{x}) = E_P[f(\mathbf{x}) - \bar{h}(\mathbf{x})]$$

$$Var(\mathbf{x}) = E_P[(h(\mathbf{x}) - \bar{h}(\mathbf{x}))^2]$$

- To get expected values we *simulated* multiple data sets, by drawing with samples with replacement from the original data set
- This gives a set of hypothesis, whose predictions can be *averaged* together
- This construction is called *bagging* and reduces variance

Ensemble learning in general

- Ensemble learning algorithms work by running a *base learning algorithm* multiple times, then *combining* the predictions of the different hypotheses obtained using some form of voting
- One approach is to construct several classifiers *independently*, then combine their predictions. Examples include:
  - Bagging
  - Randomizing the test selection in decision trees
  - Using a different subset of input features to train different neural nets
- A second approach is to *coordinate* the construction of the hypotheses in the ensemble.

Additive models

- In an ensemble, the output on any instance is computed by averaging the outputs of several hypotheses, possibly with a different weighting.
- Hence, we should choose the individual hypotheses and their weight in such a way as to provide a good fit
- This suggests that instead of constructing the hypotheses independently, we should construct them such that new hypotheses focus on instances that are problematic for existing hypotheses.
- **Boosting** is an algorithm implementing this idea

Main idea of boosting

Component classifiers should concentrate more on difficult examples

- Examine the training set
- Derive some rough ”rule of thumb”
- *Re-weight* the examples of the training set, concentrating on “hard” cases for the previous rule
- Derive a second rule of thumb
- And so on... (repeat this T times)
- *Combine* the rules of thumb into a single, accurate predictor

Questions:

- How do we re-weight the examples?
- How do we combine the rules into a single classifier?

Notation

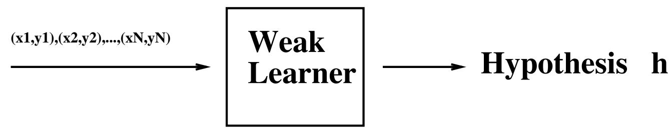
- Assume that examples are drawn independently from some probability distribution P on the set of possible data $\mathcal{D}$
- Notation: $J_P(h)$ is the expected error of h when data is drawn from P:

$$J_P(h) = \sum_{\langle \mathbf{x}, y \rangle} J(h(\mathbf{x}), y) P(\langle \mathbf{x}, y \rangle)$$

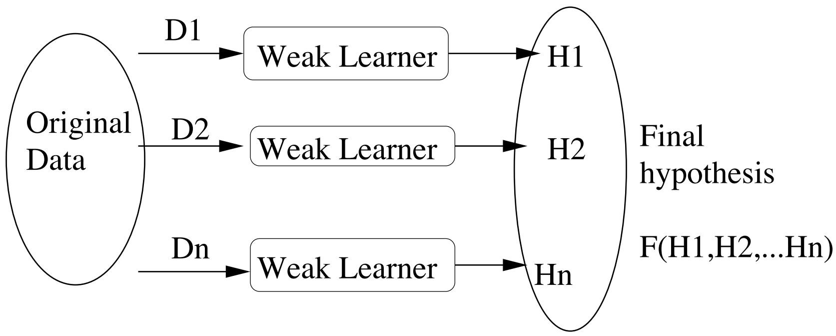
where $J(h(\mathbf{x}), y)$ could be squared error, or $0/1$ loss

Weak learners

- Assume we have some "weak" binary classifiers (e.g., decision stumps: $x_i > t$)
- "Weak" means $J_P(h) < 1/2 - \gamma$ where $\gamma > 0$ (i.e., the true error of the classifier is better than random).



Boosting classifier



AdaBoost (Freund & Schapire, 1995)

1. Input N training examples $\{(\mathbf{x_1}, y_1), \dots (\mathbf{x_N}, y_N)\}$, where $\mathbf{x_i}$ are the inputs and y_i is the desired class label
2. Let $D_1(\mathbf{x_i}) = \frac{1}{N}$ (we start with a uniform distribution)
3. Repeat T times:
   (a) Construct D_{t+1} from D_t (details in a moment)
   (b) Train a new hypothesis h_{t+1} on distribution D_{t+1}
4. Construct the final hypothesis:

$$h_f(\mathbf{x}) = \text{sign}\left(\sum_t \alpha_t h_t(\mathbf{x})\right),$$

Constructing the new distribution

We want data on which we make mistakes to be emphasized:

$$D_{t+1}(\mathbf{x_i}) = \frac{1}{Z_t} D_t(\mathbf{x_i}) \times \begin{cases} \beta_t, & \text{if } h_t(\mathbf{x_i}) = y_i \\ 1, & \text{otherwise} \end{cases} \quad \text{where}$$

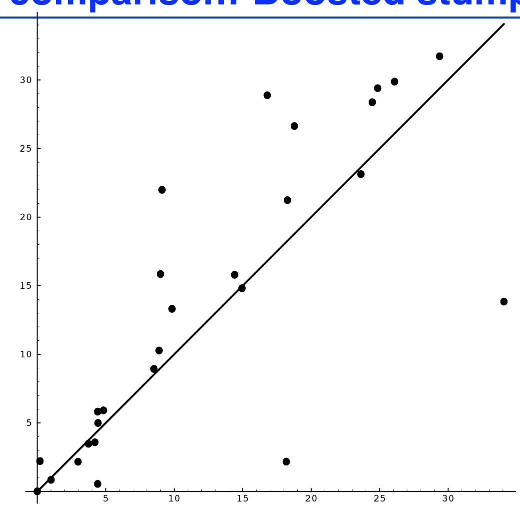
$$\beta_t = \frac{J_{D_t}(h_t)}{1 - J_{D_t}(h_t)}$$

and Z_t is a normalization factor (set such that the probabilities $D_{t+1}(x_i)$ sum to 1).

Construct the final hypothesis:

$$h_f(\mathbf{x}) = \text{sign}\left(\sum_t \alpha_t h_t(\mathbf{x})\right), \text{ where } \alpha_t = \log(1/\beta_t)$$

Empirical comparison: Boosted stumps vs. C4.5



Why does boosting work?

- Weak learners have high bias
- By combining them, we get more expressive classifiers
- Hence, boosting is a *bias-reduction technique*
- What happens as we run boosting longer?

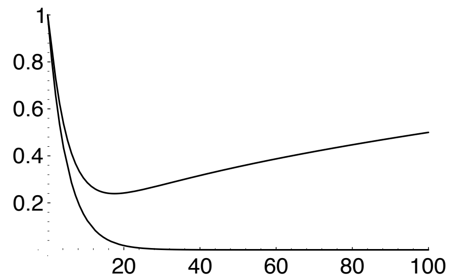
Why does boosting work?

- Weak learners have high bias
- By combining them, we get more expressive classifiers
- Hence, boosting is a *bias-reduction technique*
- What happens as we run boosting longer?

  Intuitively, we get more and more complex hypotheses
- How would you expect bias and variance to evolve over time?

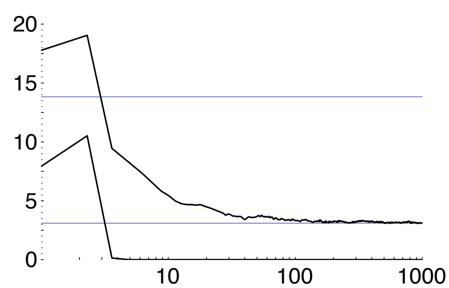
A naive (but reasonable) analysis of generalization error

- Expect the training error to continue to drop (until it reaches 0)
- Expect the test error to *increase* as we get more voters, and h_f becomes too complex.



Actual typical run of AdaBoost

Boosting C4.5 on the letter dataset:



- Test error *does not increase* even after 1000 runs! (more than 2 million decision nodes!)
- Test error *continues to drop* even after training error reaches 0!

These are consistent results through many sets of experiments!

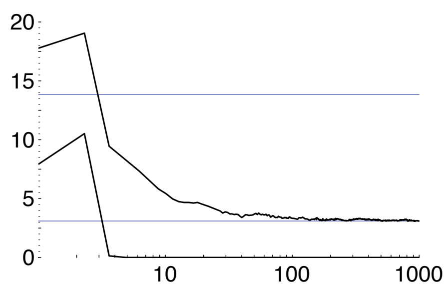
Recall: Classification margin

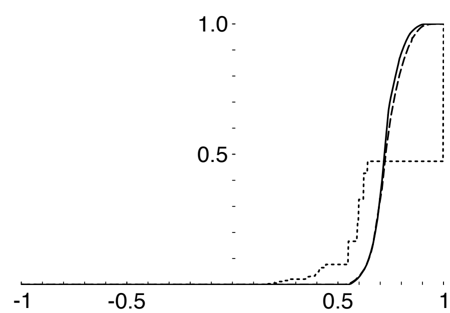
- Boosting constructs hypotheses of the form
  $h_f(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$
- The classification of an example is correct if $\text{sign}(f(\mathbf{x})) = y$
- The **margin** is defined as:

$$\text{margin}(f(\mathbf{x}), y) = y \cdot f(\mathbf{x})$$

- The margin tells us how close the decision boundary is to the data points on each side.
- A higher margin on the training set should yield a lower generalization error
- Intuitively, increasing the margin is similar to lowering the variance

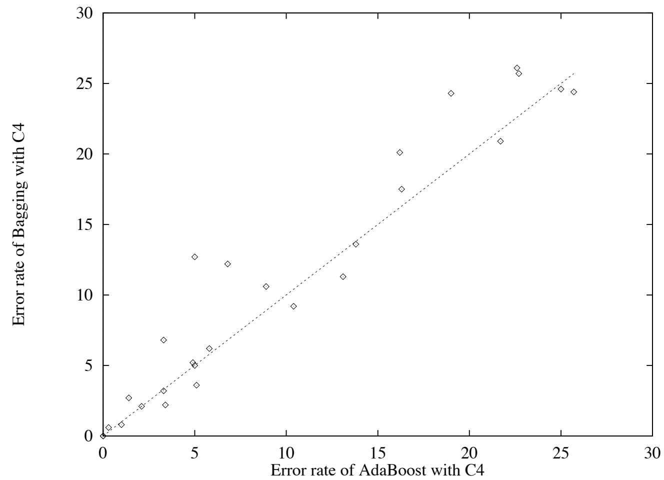
Effect of boosting on the margin





- Between rounds 5 and 10 there is no training error reduction
- But there is a *significant shift* in margin distribution!
- There is a formal proof that boosting increases the margin

Bagging vs. Boosting



Parallel of bagging and boosting

- Bagging is typically faster, but may get a smaller error reduction (not by much)
- Bagging works well with “reasonable” classifiers
- Boosting works with very simple classifiers
  E.g., Boostexter - text classification using decision stumps based on single words
- Boosting may have a problem if a lot of the data is mislabeled, because it will focus on those examples a lot, leading to overfitting.

Summary

- Ensemble methods combine several hypotheses into one prediction
- They work better than the best individual hypothesis from the same class because they reduce bias or variance (or both)
- Bagging is mainly a variance-reduction technique, useful for complex hypotheses
- Main idea is to sample the data repeatedly, train several classifiers and average their predictions.
- Boosting focuses on harder examples, and gives a weighted vote to the hypotheses.
- Boosting works by reducing bias and increasing classification margin.

Binary classification: The golden goal

Given:

- The set of all possible instances X
- A target function (or concept) $f : X \rightarrow \{0, 1\}$
- A set of hypotheses H
- A set of training examples D (containing positive and negative examples of the target function)

$$\langle \mathbf{x_1}, f(\mathbf{x_1}) \rangle, \ldots \langle \mathbf{x_m}, f(\mathbf{x_m}) \rangle$$

Determine:

A hypothesis $h \in H$ such that $h(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x} \in X$.

Approximate Concept Learning

- Requiring a learner to acquire the *right* concept is too strict
- Instead, we will allow the learner to produce a *good approximation* to the actual concept
- For any instance space, there is a non-uniform likelihood of seeing different instances
- We assume that there is a *fixed probability distribution* P on the space of instances X
- The learner is trained and tested on examples whose inputs are drawn *independently and randomly*, according to P.

Recall: Two Notions of Error

Training error of hypothesis h with respect to target concept f:

- How often $h(\mathbf{x}) \neq f(\mathbf{x})$ over the training instances

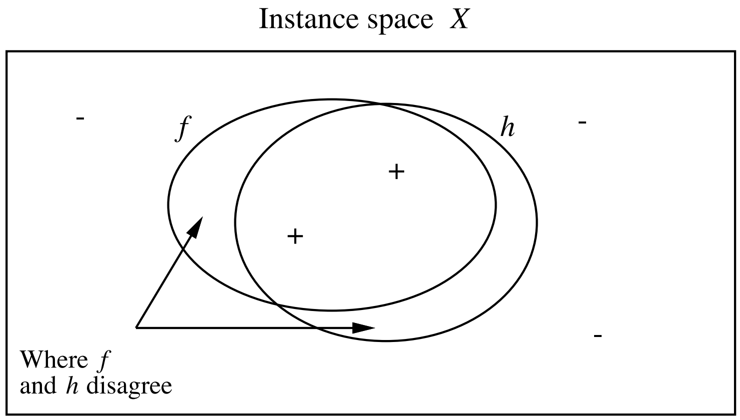
True error of hypothesis h with respect to target concept f:

- How often $h(\mathbf{x}) \neq f(\mathbf{x})$ over future, unseen instances (but drawn according to P)

Questions:

- Can we *bound* the true error of a hypothesis given only its training error?
- How many examples are needed for a good approximation?

True Error of a Hypothesis



True Error Definition

The set of instances on which the target concept and the hypothesis disagree is denoted: $S = \{\mathbf{x} | h(\mathbf{x}) \neq f(\mathbf{x})\}$

The **true error** of h with respect to f is:

$$\sum_{\mathbf{x} \in S} P(\mathbf{x})$$

This is the probability of making an error on an instance randomly drawn from X according to P

Let $\epsilon \in (0, 1)$ be an **error tolerance** parameter. We say that h is a **good approximation** of f (to within ϵ) if and only if the true error of h is less than ϵ.

Example: Rote Learner

- Let $X = \{0,1\}^n$. Let P be the uniform distribution over X.
- Let the concept f be generated by randomly assigning a label to every instance in X.
- Let $D \subset X$ be a set of training instances.
  The hypothesis h is generated by memorizing D and giving a random answer otherwise.
- What is the training error of h?
- What is the true error of h?

Empirical risk minimization

- Suppose we are given a hypothesis class H
- We have a magical learning machine that can sift through H and output the hypothesis with the *smallest training error*, h_{emp}
- This is process is called **empirical risk minimization**
- Is this a good idea?
- What can we say about the error of the other hypotheses in h?

First tool: The union bound

Let $E_1 \ldots E_k$ be k different events (not necessarily independent). Then:

$$P(E_1 \cup \cdots \cup E_k) \leq P(E_1) + \cdots + P(E_k)$$

Second tool: Hoeffding (Chernoff) bound

Let $Z_1 \ldots Z_m$ be m independent identically distributed (iid) binary variables, drawn from a Bernoulli (binomial) distribution:

$$P(Z_i = 1) = \phi \text{ and } P(Z_i = 0) = 1 - \phi$$

Let $\hat{\phi}$ be the mean of these variables:

$$\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} Z_i$$

Let ϵ be a fixed error tolerance parameter. Then:

$$P(|\phi - \hat{\phi}| > \epsilon) \leq 2e^{-2\epsilon^2 m}$$

In other words, if you have lots of examples, the empirical mean is a good estimator of the true probability.

Finite hypothesis space

- Suppose we are considering a finite hypothesis class $H = \{h_1, \ldots h_k\}$ (e.g. conjunctions, decision trees,...)
- Take an arbitrary hypothesis $h_i \in H$
- Suppose we sample data according to our distribution an let $Z_j = 1$ iff $h_i(\mathbf{x_j}) \neq y_j$
- So $e(h_i)$ (the true error of h_i) is the expected value of Z_j
- Let $\hat{e}(h_i) = \frac{1}{m}\sum_{j=1}^{m} Z_j$ (this is the empirical training error of h_i on the data set we have)
- Using the Hoeffding bound, we have:

$$P(|e(h_i) - \hat{e}(h_i)| > \epsilon) \leq 2e^{-2\epsilon^2 m}$$

- So, if we have lots of data, the training error of a hypothesis h_i will be close to its true error with high probability.

What about all hypotheses?

- We showed that the empirical error is "close" to the true error for one hypothesis.
- Let E_i denote the event $|e(h_i) - \hat{e}(h_i)| > \epsilon$
- Can we guarantee this is true for *all* hypothesis?

$$\begin{aligned}
P(\exists h_i \in H, |e(h_i) - \hat{e}(h_i)| > \epsilon) &= P(E_1 \cup \ldots E_k) \\
&\leq \sum_{i=1}^{k} P(E_i) \text{ (union bound)} \\
&\leq \sum_{i=1}^{k} 2e^{-2\epsilon^2 m} \text{ (shown before)} \\
&= 2ke^{-2\epsilon^2 m}
\end{aligned}$$

A uniform convergence bound

- We showed that:

$$P(\exists h_i \in H, |e(h_i) - \hat{e}(h_i)| > \epsilon) \leq 2ke^{-2\epsilon^2 m}$$

- So we have:

$$1 - P(\exists h_i \in H, |e(h_i) - \hat{e}(h_i)| > \epsilon) \geq 1 - 2ke^{-2\epsilon^2 m}$$

  or, in other words:

$$P(\forall h_i \in H, |e(h_i) - \hat{e}(h_i)| < \epsilon) \geq 1 - 2ke^{-2\epsilon^2 m}$$

- This is called a **uniform convergence** result because the bound holds for all hypotheses
- What is this good for?

Sample complexity

- Suppose we want to guarantee that with probability at least $1 - \delta$, the sample (training) error is within ϵ of the true error.
- From our bound, we can set $\delta \geq 2ke^{-2\epsilon^2 m}$
- Solving for m, we get that the number of samples should be:

$$m \geq \frac{1}{2\epsilon^2} \log \frac{2k}{\delta} = \frac{1}{2\epsilon^2} \log \frac{2|H|}{\delta}$$

- So the number of samples needed is *logarithmic* in the size of the hypothesis space

Example: Conjunctions of Boolean Literals

Let H be the space of all pure conjunctive formulae over n Boolean attributes.

Then $|H| = 3^n$ (why?)

From the previous result, we get:

$$m \geq \frac{1}{2\epsilon^2} \log \frac{2|H|}{\delta} = n \frac{1}{2\epsilon^2} \log \frac{6}{\delta}$$

This is linear in n!

Another application: Bounding the true error

$$P(\forall h_i \in H, |e(h_i) - \hat{e}(h_i)| < \epsilon) \geq 1 - 2ke^{-2\epsilon^2 m} = 1 - \delta$$

Suppose we hold m and δ fixed, and we solve for ϵ. Then we get:

$$|e(h_i) - \hat{e}(h_i)| \leq \sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}$$

inside the probability term.

Can we now prove anything about the generalization power of the empirical risk minimization algorithm?

Empirical risk minimization

Let h^* be the best hypothesis in our class (in terms of true error). Based on our uniform convergence assumption, we can bound the true error of h_{emp} as follows:

$$\begin{aligned} e(h_{emp}) &\leq \hat{e}(h_{emp}) + \epsilon \\ &\leq \hat{e}(h^*) + \epsilon \text{ (because } h_{emp} \text{ has better training error} \\ &\quad \text{than any other hypothesis)} \\ &\leq e(h^*) + 2\epsilon \text{ (by using the result on } h^*) \\ &\leq e(h^*) + 2\sqrt{\frac{1}{2m}\log\frac{2|H|}{\delta}} \text{ (from previous slide)} \end{aligned}$$

This bounds how much worse h_{emp} is, wrt the best hypothesis we can hope for!

Bias and variance revisited

We showed that, given m examples, with probability at least $1-\delta$,

$$e(h_{emp}) \leq \left(\min_{h\in H} e(h)\right) + 2\sqrt{\frac{1}{2m}\log\frac{2|H|}{\delta}}$$

Suppose now that we are considering two hypothesis classes $H \subseteq H'$

- The first term would be smaller for H' (we have a larger hypothesis class, hence less “bias”)
- The second term would be larger (the “variance” is increasing)