6.034f Boosting Notes
Patrick Winston and Luis Ortiz
Draft of November 17, 2009

A *strong* classifier is one that has an error rate close to zero. A *weak* classifier is one that has an error rate just below $\frac{1}{2}$, producing answers just a little better than a coin flip.

Freund and Schapire discovered that you can construct a strong classifier from weak classifiers such that the strong classifier will correctly classify all samples in a sample set. Better still, the strong classifier consists of a sequence of weighted classifiers, determined step by step. In the following, the multiplicative weight found in the first step is α^1 and the weak classifier is $h^1(x_i)$, and in step s, the weight is α^s and the classifier is $h^s(x_i)$:

$$H(x_i) = \text{sign}(\alpha^1 h^1(x_i) + \alpha^2 h^2(x_i) + \cdots + \alpha^s h^s(x_i) \cdots$$

where

$$\text{sign} = \begin{cases} +1 & \text{for positive arguments} \\ -1 & \text{for negative arguments} \end{cases}$$

$$h^s(x_i) = \begin{cases} +1 & \text{for samples the classifier thinks belong to the class} \\ -1 & \text{for samples the classifier thinks do not belong to the class} \end{cases}$$

Freund and Schapire named their method *Adaboost*, an acronym for *ada*ptive *boosting*.

In the first Adaboost step, you find the weak classifier, $h^1(x_i)$, that produces the lowest error rate; then, you find the corresponding multiplier, α^1. In step s, you first find the weak classifier, $h^s(x_i)$, that produces the lowest error rate with the samples reweighted to emphasize previously misclassified samples; then you find the corresponding multiplier, α^s. You continue taking steps until the classifier $H(x_i)$ correctly classifies all samples or you cannot find any weak classifier for the next step.

Several questions emerge:

- How do you compute the error rate[†], E, for the candidates for $h^s(x_i)$?
- How do you compute α^s once you have $h^s(x_i)$?
- How do you reweight the samples to emphasize the misclassified samples for the next step?

To compute the error rate, you assign to each sample, i, at each step, s, an emphasis-determining weight, w_i^s. The weights used in step 1 are all the same:

$$w_i^1 = \frac{1}{\text{number of samples}}$$

Each time you calculate the weights for the next step, you normalize so that the weights still add to 1:

$$\sum_i w_i^s = 1$$

[†] We use E for the error rate to avoid confusion with the base of the natural logarithms, e.

The error rate of a candidate classifier for a particular step is the sum of the weights for the samples that the candidate classifier misclassifies at that step:

$$E^s_{\text{candidate}} = \sum_i \mathrm{w}_i^s \qquad \text{for } i \text{ misclassified by the candidate at step } s$$

E^s is the error rate for the best of the candidate classifiers.

But what about computing α^s and reweighting the samples. With some moderately complex mathematics, Freund and Shipiri determined that computing new weights from old weights using the following formula ensures that the overall error for $\mathrm{H}(x_i)$, as you add classifiers, will stay under an exponential bound, and eventually go to zero. N^s is a normalizing constant[†] for step s that ensures that all the new weights, w_i^{s+1}, add up to 1.

$$\mathrm{w}_i^{s+1} = \begin{cases} \frac{\mathrm{w}_i^s}{N^s} e^{-\alpha^s} & \text{for correctly classified samples} \\ \\ \frac{\mathrm{w}_i^s}{N^s} e^{+\alpha^s} & \text{for misclassified samples} \end{cases}$$

With more math, Freund and Shipiri determined that the exponential bound on the overall error of $\mathrm{H}(x_i)$ is minimized if N^s is minimized. This led them to a formula for α^s:

$$\alpha^s = \frac{1}{2} \ln \frac{1 - E^s}{E^s}$$

At this point, you have all you need to write an Adaboost program:

- You use uniform weights to start.
- For each step, you find the classifier that yields the lowest error rate for the current weights, w_i^s.
- You use that best classifier, $\mathrm{h}^s(x_i)$, to compute the error rate associated with the step, E^s
- You determine the alpha for the step, α^s from the error for the step, E^s.
- With the alpha in hand, you compute the weights for the next step, w_i^{s+1}, from the weights for the current step, w_i^s, taking care to include a normalizing factor, N^s, so that the new weights add up to 1.
- You stop successfully when $\mathrm{H}(x_i)$ correctly classifies all the samples, x_i; you stop unsuccessfully if you reach a point where there is no weak classifier, one with an error rate $< \frac{1}{2}$.

You, however, are not a computer, so calculating those exponentials and logarithms is impractical on an examination. You need to massage the formulas a bit to make them work for you.

First, you plug the formula for α^s into the reweighting formula, producing the following:

$$\mathrm{w}_i^{s+1} = \begin{cases} \frac{\mathrm{w}_i^s}{N^s} \sqrt{\frac{E^s}{1-E^s}} & \text{for correctly classified samples} \\ \\ \frac{\mathrm{w}_i^s}{N^s} \sqrt{\frac{1-E^s}{E^s}} & \text{for misclassified samples} \end{cases}$$

Now, because N^s must be that number that makes the new weights add up to 1, you can write the following:

[†] We use N^s rather than Z^s, used by Freund and Schapire, to avoid confusion with the number 2 when written by hand.

$$N^s = \sqrt{\frac{1-E^s}{E^s}} \sum_{\text{wrong}} w_i^s + \sqrt{\frac{E^s}{1-E^s}} \sum_{\text{right}} w_i^s$$

Note that the sum of the weights over the misclassified samples is the error rate, E^s, and therefore the sum of the rest of the weights must be $1 - E^s$. Accordingly, N^s simplifies:

$$N^s = 2\sqrt{E^s(1-E^s)}$$

Plugging this value into the formula for calculating the new weights yields:

$$w_i^{s+1} = \begin{cases} \frac{w_i^s}{2\sqrt{E^s(1-E^s)}} \sqrt{\frac{E^s}{1-E^s}} = \frac{1}{2}\frac{w_i^s}{(1-E^s)} & \text{for correctly classified samples} \\ \frac{w_i^s}{2\sqrt{E^s(1-E^s)}} \sqrt{\frac{1-E^s}{E^s}} = \frac{1}{2}\frac{w_i^s}{E^s} & \text{for misclassified samples} \end{cases}$$

This makes sense because you know that you need a classifier with an error rate of less than $\frac{1}{2}$ to succeed. Hence, $E^s < \frac{1}{2}$, so the new weights for misclassified samples will go up and the new weights for the correctly classified samples will go down, thus emphasizing the misclassified samples.

You can also usefully note the sums of the new weights:

$$\sum_{\text{wrong}} w_i^{s+1} = \frac{1}{2E^s} \sum_{\text{wrong}} w_i^s = \frac{1}{2}$$

$$\sum_{\text{right}} w_i^{s+1} = \frac{1}{2(1-E^s)} \sum_{\text{right}} w_i^s = \frac{1}{2}$$

You conclude that you can find the new weights by scaling the weights of the misclassified samples up to $\frac{1}{2}$ and by scaling the weights of the correctly classified samples down to $\frac{1}{2}$.

You do not even have to compute any logarithms to get the alphas, because you do not need the alphas to determine the result produced by $H(x_i)$. Here is why: the result is positive if the sum of the alphas for the weak classifiers that return +1 is greater than the sum of the alphas for the weak classifiers that return -1. But the sum of the logarithms of a set of expressions is the logarithm of the product of those expressions. Accordingly, the $H(x_i)$ returns +1 if the following holds:

$$\prod_{h^s(x_i) \text{ that yield } +} \frac{(1-E^s)}{E^s} \geq \prod_{h^s(x_i) \text{ that yield -}} \frac{(1-E^s)}{E^s}$$

Thus, multiplying a few simple fractions usually suffices on an examination. Of course, if all the weak classifiers agree, you need not do any calculation at all to determine the result returned by $H(x_i)$.

6.034 Artificial Intelligence

Fall 2010