Appendix - AdaBoost link to Gradient Boosting

AdaBoost Algorithm

Recall our AdaBoost Algorithm...

1. Initialize the observation weights $w_i = \frac{1}{N}$, for i = 1, 2, ..., N
2. For m = 1 to M, **do**:
   1. Fit a classifier $G_m(x)$ to the training data using weights w_i.
   2. Compute: $err_m = \dfrac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$
   3. Compute $\alpha_m = log((1 - err_m)/err_m)$
   4. Set $w_i = w_i * exp[\alpha_m * I(y_i \neq G_m(x_i))]$, i = 1, 2, ..., N.
3. Output $G(X) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$

Gradient Boosting Algorithm

Recall our general Gradient boosting algorithm...

1. Initialize $G_0(x)$ (the first tree) $= argmin_\gamma \sum_{i=1}^{N} L(y_i, \phi(x_i; \gamma))$
2. For m = 1 to M, **do**:
    1. Compute the gradient: $r_{im} = -\dfrac{\partial L(y_i, G_{m-1}(x_i))}{\partial G_{m-1}(x_i)}$
    2. Use the weak leaner (here a tree) to compute γ_m which minimizes:
    $$\sum_{i=1}^{N} L(r_{im}, \phi(x_i; \gamma))$$
    3. Update $G_m(X) = G_{m-1}(X) + v\phi(X; \gamma_m)$
3. Return $G(X) = G_M(x)$

Gradient Boosting - Step 2.2

- Use the weak leaner (here a tree) to compute γ_m which minimizes:

$$\sum_{i=1}^{N} L(r_{im}, \phi(x_i; \gamma))$$

- With **AdaBoost**, we use exponential loss

$$exp(-y_i * \hat{y}_i)$$

- At Step 2.2 (above), then, we find minimize the following:

$$argmin_{\gamma} \sum_{i=1}^{N} exp[-y_i * \hat{y}_i]$$

Gradient Boosting - Simplifications/Notes

- Recall that at stage m, $\hat{y}_i = f_m(x_i)$, and $f_m(x_i) = f_{m-1}(x_i) + \alpha_m \phi(x_i, \gamma_m)$
- So, for Step 2.2, we can rewrite our minimization as follows:

$$argmin_{\gamma,\alpha} \sum_{i=1}^{N} exp[-y_i * (f_{m-1}(x_i) + \alpha\phi(x_i, \gamma))]$$

- Some notes before diving in:
  - For this derivation, we are assuming **binary classification**, where we are fitting to $y_i \epsilon$-1, $+1$ (e.g. the negative cases are given by -1, and the positive cases a $+1$). This will simplify the math.
  - For the remainder of this derivation, we're going to drop γ within ϕ (let's just minimize our loss over α), and denote our loss as follows:

$$L_m(\phi) = \sum_{i=1}^{N} exp[-y_i * (f_{m-1}(x_i) + \alpha\phi(x_i))]$$

  - We're basically going to take α from this loss and work through to $\alpha_m = log((1 - err_m)/err_m)$ that we saw in the **AdaBoost** algorithm

Gradient Boosting to AdaBoost

- Let's minimize our loss:

$$L_m(\phi) = \sum_{i=1}^{N} exp[-y_i * (f_{m-1}(x_i) + \alpha\phi(x_i))]$$

- Before we get there, let's somehow work in those weights (w_i) that we have in AdaBoost (this is going to be a long, messy derivation, but we'll put it all back together at some point)

Gradient Boosting to AdaBoost I

1. $L_m(\phi) = \sum_{i=1}^{N} exp[-y_i * (f_{m-1}(x_i) + \alpha\phi(x_i))]$
2. $L_m(\phi) = \sum_{i=1}^{N} exp[-y_i * f_{m-1}(x_1)] * exp[-\alpha y_i \phi(x_i)]$
3.
   1. Define $w_{i,m} = exp[-y_i * f_{m-1}(x_i)]$, and then plug that in:
   2. $L_m(\phi) = \sum_{i=1}^{N} w_{i,m} * exp[-\alpha y_i \phi(x_i)]$
4.
   1. If $y_i = \phi(x_i)$, then $y_i * \phi(x_i) = 1$, and $exp[-\alpha y_i \phi(x_i)] = exp[-\alpha]$
   2. If $y_i \neq \phi(x_i)$, then $y_i * \phi(x_i) = -1$, and $exp[-\alpha y_i \phi(x_i)] = exp[\alpha]$
   3. Use that result to obtain the following: $L_m(\phi) = \sum_{y_i = \phi(x_i)} w_{i,m} * e^{-\alpha} + \sum_{y_i \neq \phi(x_i)} w_{i,m} * e^{\alpha}$

Gradient Boosting to AdaBoost II

5. Since those α's aren't depending on i, we can move them outside the summation:
$L_m(\phi) = e^{-\alpha} \sum_{y_i=\phi(x_i)} w_{i,m} + e^{\alpha} \sum_{y_i \neq \phi(x_i)} w_{i,m}$
6. Instead of taking the sum over just those observations where $y_i = \phi(x_i)$ or $y_i \neq \phi(x_i)$, we can take it over all observations and move that condition inside the summation:

$$L_m(\phi) = e^{-\alpha} \sum_{i=1}^{N} w_{i,m} I(y_i = \phi(x_i)) + e^{\alpha} \sum_{i=1}^{N} w_{i,m} I(y_1 \neq \phi(x_i))$$

Note: The I is the indicator function.

7. Next, we do a little math (see the change in the first term):

$$L_m(\phi) = e^{-\alpha} \sum_{i=1}^{N} w_{i,m}(1 - I(y_i \neq \phi(x_i))) + e^{\alpha} \sum_{i=1}^{N} w_{i,m} I(y_1 \neq \phi(x_i))$$

Gradient Boosting to AdaBoost IV

Some more manipulation...

8. $L_m(\phi) = e^{-\alpha} \sum_{i=1}^{N} w_{i,m} - e^{-\alpha} \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i)) + e^{\alpha} \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))$
9. $L_m(\phi) = e^{-\alpha} \sum_{i=1}^{N} w_{i,m} + (e^{\alpha} - e^{-\alpha}) \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))$
10. Okay, so now we have our loss in a format where we can take the derivative with respect to α, set it equal to 0, and solve.

Gradient Boosting to AdaBoost V

11. First let's take the derivative $\frac{\partial L_m(\phi)}{\partial \alpha}$, and set it equal to 0:

$$0 = -e^{-\alpha} \sum_{i=1}^{N} w_{i,m} + (e^{\alpha} + e^{-\alpha}) \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))$$

12. Multiply through on the right side:

$$0 = -e^{-\alpha} \sum_{i=1}^{N} w_{i,m} + e^{\alpha} \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i)) + e^{-\alpha} \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))$$

13. Move the $-e^{-\alpha}\sum_{i=1}^{N} w_{i,m}$ to the left side, and then divide each side by it: $1 = \dfrac{e^{\alpha}\sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i)) + e^{-\alpha}\sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))}{-e^{-\alpha}\sum_{i=1}^{N} w_{i,m}}$

14. Divide every term on the right side by $e^{-\alpha}$:

$$1 = \frac{e^{2\alpha} \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i)) + \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))}{\sum_{i=1}^{N} w_{i,m}}$$

15. Multiply through by the $\sum_{i=1}^{N} w_{i,m}$ on the bottom:

$$\sum_{i=1}^{N} w_{i,m} = e^{2\alpha} \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i)) + \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))$$

Gradient Boosting to AdaBoost VII

16. Subtract the $\sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))$ from the right, and then divide by it to isolate $e^{2\alpha}$:

$$e^{2\alpha} = \frac{\sum_{i=1}^{N} w_{i,m} - \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))}{\sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))}$$

17. Take the log of everything (with base e), and simplifiy:

$$\alpha = \frac{1}{2} \log \left(\frac{\sum_{i=1}^{N} w_{i,m} - \sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))}{\sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))} \right)$$

Gradient Boosting to AdaBoost VIII

18. Because math (I'm tired of this derivation and you should be, too):

$$\alpha = \frac{1}{2}\log\left(\frac{\sum_{i=1}^{N} w_{i,m}}{\sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))} - 1\right)$$

19. Denote err_m as **AdaBoost** does (then see what we do in the next slide):

$$err_m = \frac{\sum_{i=1}^{N} w_{i,m} I(y_i \neq \phi(x_i))}{\sum_{i=1}^{N} w_{i,m}}$$

Gradient Boosting to AdaBoost IX

20. Using that definition of err_m in 19, we can re-write 18 as:

$$\alpha = \frac{1}{2} \log \left(\frac{1}{err_m} - \frac{err_m}{err_m} \right)$$

21. We've made it!! (Rework 20. . . ..):

$$\alpha = \frac{1}{2} \log \left(\frac{1 - err_m}{err_m} \right)$$

Note: Since the $\frac{1}{2}$ is a constant, it's not important in this case.