Boosting with AdaBoost

Cornell CS 3/5780 · Spring 2026

Ensemble Methods

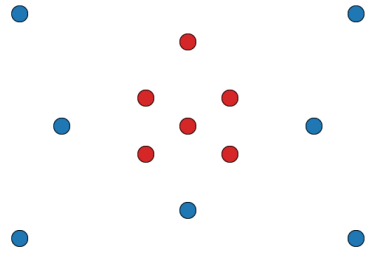
- Train multiple models/hypotheses and combine them
- Ensemble methods are plug-and-play (can be used with any base algorithm for learning)
- Last lecture: bagging, which takes high variance/low bias methods and ensembles them to reduce variance (fixes overfitting)

$$H_{\text{Bagged}}(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m} h_i(\mathbf{x})$$

  - Sample m points with replacement uniformly from training data
  - Random forest: bagged decision trees with max depth m and $k < d$ subsampled features (often $k = \sqrt{(d)}$)
- Today: boosting, which takes high bias/low variance methods and ensembles them to obtain low bias model (fixes underfitting)

Boosting Setting

- Training data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with $y_i \in \{+1, -1\}$ (binary classification)
- **Weak learner**: a high bias (simple) classification algorithm that does (slightly) better than chance
  - Examples: depth 1 decision tree, nearly random classifier
- Question: how can you combine depth 1 DT to get zero training error for the following?



Boosting Algorithm Idea

- How can we ensemble weak learners to get an algorithm with 0 training error?
- Idea: **sequentially** weight and sample points from training data (more weight on points with errors)
- Ensemble will have the form

$$H_{\text{Boosted}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

1. Initialize equal weights w_1
2. For $t = 1$ to T:
   1. Weight datapoints in D according to w_t to get D_t
   2. Use weak learning algorithm on D_t to obtain classifier h_t
   3. Add h_t to ensemble and update w_{t+1} based on errors
3. Return ensembled classifier

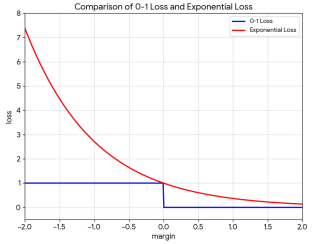
Adaboost Algorithm

1. set $w_1[i] = \frac{1}{n}$ for all i *(initialize uniformly)*
2. set $H_0 = 0$ *(initialize ensemble to 0)*
3. For $t = 1$ to T:
    1. Weight points in D according to w_t to get D_t
    2. Obtain h_t via weak learning algorithm on D_t
    3. Compute error and ensemble weight
    $$\epsilon_t = \sum_{i=1}^{n} w_t[i]\mathbf{1}\{h_t(\mathbf{x}_i) \neq y_i\}, \quad \alpha_t = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$
    4. Update ensemble $H_t = H_{t-1} + \alpha_t h_t$
    5. Update weights $w_{t+1}[i] = \frac{w_t[i]\exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^{n} w_t[j]\exp(-\alpha_t y_j h_t(\mathbf{x}_j))}$

Boosting Theorem

- **Weak Learner Assumption**: For all distributions over points in D, the weak learning algorithm can produce a hypothesis whose weighted classification error is less than $1/2 - \gamma$.
- **Boosting Theorem:** If the weak learner assumption holds with some margin γ, the boosted classifier will have **0 training error** after
$$T = O\left(\frac{\log(n)}{\gamma^2}\right)$$
- Implications of this theorem:
    - Each weak learner is high bias but low variance classifier
    - We only combine $O(\log(n))$ of these weak learners
    - So Boosted classifier will **not** have too high of a variance

Proof Step 1: Bounding Error by Exponential Loss



- We will bound the training error
$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{H_T(\mathbf{x}_i) \neq y_i\}$$
- Define the **exponential loss**:
$$\Phi_T = \frac{1}{n}\sum_{i}\exp(-y_i H_T(x_i))$$
- Notice that if $y_i H_T(\mathbf{x}_i) \leq 0$, then $\exp(-y_i H_T(x_i)) \geq 1$, so Φ_T *upper bounds* the training error
- We will show that the exponential loss decreases by a fixed multiplicative factor at each boosting round.

Proof Step 2: Loss Shrinks Multiplicatively

- Recall the update rule
$$H_{t+1}(x) = H_t(x) + \alpha_{t+1}h_{t+1}(x)$$
- Therefore,
$$\Phi_{t+1} = \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i H_{t+1}(\mathbf{x}_i)) = \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i H_t(\mathbf{x}_i))\exp(-\alpha_{t+1}y_i h_{t+1}(\mathbf{x}_i))$$
$$= \Phi_t \sum_{i=1}^{n}\frac{\exp(-y_i H_t(\mathbf{x}_i))}{n\Phi_t}\exp(-\alpha_{t+1}y_i h_{t+1}(\mathbf{x}_i))$$
- Define weights: $p_t[i] = \exp(-y_i H_t(\mathbf{x}_i))/\left(\sum_{j=1}^{n}\exp(-y_j H_t(\mathbf{x}_j))\right)$
    - Can show by induction that $p_t[i] = w_t[i]$ algorithm weights
- Define multiplier $Z_{t+1} = \sum_{i=1}^{n} w_t[i]\exp(-\alpha_{t+1}y_i h_{t+1}(\mathbf{x}_i))$
- Final recursive update is
$$\Phi_{t+1} = \Phi_t Z_{t+1}$$

Proof Step 3: Using Weak Learner Assumption

- Now we just need to show that $Z_t = \sum_{i=1}^n w_{t-1}[i] \exp(-\alpha_t y_i h_t(\mathbf{x}_i)) < 1$
- Since both $h_t(\mathbf{x}_i), y_i \in \{-1, +1\}$, the product $y_i h_t(\mathbf{x}_i)$ is always ± 1:

$$Z_t = \sum_{i:, h_t(\mathbf{x}i) = y_i} w_t[i] e^{-\alpha_t} + \sum_{i:, h_t(\mathbf{x}_i) \neq y_i} w_t[i] e^{\alpha_t}$$

- Recall that

$$\epsilon_t = \sum_{i=1}^n w_t[i] \mathbf{1}\{h_t(\mathbf{x}_i) \neq y_i\}, \quad \alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

- Therefore,

$$Z_t = (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t} = 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

- Under the **weak learner assumption**, $\epsilon_t \leq \frac{1}{2} - \gamma$ so

$$Z_t \leq \sqrt{1 - 4\gamma^2}$$

Proof Step 4: Final Bound

- Putting it all together, we show the exponential loss decreases quickly

$$\Phi_{t+1} = \Phi_t Z_{t+1} \leq \Phi_t \sqrt{1-4\gamma^2} \implies \Phi_T \leq (\sqrt{1-4\gamma^2})^T \Phi_0$$

- Simplify using $1 - x \leq e^{-x}$ and $\Phi_0 = 1$:

$$\text{Training error} \;\leq\; \Phi_T \;\leq\; e^{-2\gamma^2 T}$$

- Lastly, notice that by 0-1 loss, if the training error is less than $\frac{1}{n}$, it must be exactly zero.

$$e^{-2\gamma^2 T} < \frac{1}{n} \iff T > \frac{\log(n)}{2\gamma^2}$$