Beyond Binary Classification

- Before we’ve talked about combining weak predictor (boosting)
Beyond Binary Classification

- Before we’ve talked about combining weak predictor (boosting)
  - What if you have strong predictors?
Beyond Binary Classification

- Before we’ve talked about combining weak predictor (boosting)
  - What if you have strong predictors?
- How do you make inherently binary algorithms multiclass?
- How do you answer questions like ranking?
General Online Setting

- For $t = 1$ to $T$:
  - Get instance $x_t \in X$
  - Predict $\hat{y}_t \in Y$
  - Get true label $y_t \in Y$
  - Incur loss $L(\hat{y}_t, y_t)$

- Classification: $Y = \{0, 1\}$, $L(y, y') = |y' - y|$

- Regression: $Y \subset \mathbb{R}$, $L(y, y') = (y' - y)^2$
General Online Setting

- For \( t = 1 \) to \( T \):
  - Get instance \( x_t \in X \)
  - Predict \( \hat{y}_t \in Y \)
  - Get true label \( y_t \in Y \)
  - Incur loss \( L(\hat{y}_t, y_t) \)

- Classification: \( Y = \{0, 1\} \), \( L(y, y') = |y' - y| \)

- Regression: \( Y \subset \mathbb{R} \), \( L(y, y') = (y' - y)^2 \)

- **Objective**: Minimize total loss \( \sum_t L(\hat{y}_t, y_t) \)
Prediction with Expert Advice

- For $t = 1$ to $T$:
  - Get instance $x_t \in X$ and advice $a_t, i \in Y, i \in [1, N]$
  - Predict $\hat{y}_t \in Y$
  - Get true label $y_t \in Y$
  - Incur loss $L(\hat{y}_t, y_t)$
Prediction with Expert Advice

- For \( t = 1 \) to \( T \):
  - Get instance \( x_t \in X \) and advice \( a_t, i \in Y, i \in [1, N] \)
  - Predict \( \hat{y}_t \in Y \)
  - Get true label \( y_t \in Y \)
  - Incur loss \( L(\hat{y}_t, y_t) \)

- **Objective**: Minimize regret, i.e., difference of total loss vs. best expert

\[
\text{Regret}(T) = \sum_{t} L(\hat{y}_t, y_t) - \min_i \sum_{t} L(a_{t,i}, y_t)
\]  

(1)
Mistake Bound Model

- Define the maximum number of mistakes a learning algorithm $L$ makes to learn a concept $c$ over any set of examples (until it’s perfect).

$$M_L(c) = \max_{x_1, \ldots, x_T} |\text{mistakes}(L, c)|$$

(2)

- For any concept class $C$, this is the max over concepts $c$.

$$M_L(C) = \max_{c \in C} M_L(c)$$

(3)
Mistake Bound Model

- Define the maximum number of mistakes a learning algorithm $L$ makes to learn a concept $c$ over any set of examples (until it’s perfect).

\[ M_L(c) = \max_{x_1, \ldots, x_T} |\text{mistakes}(L, c)| \quad (2) \]

- For any concept class $C$, this is the max over concepts $c$.

\[ M_L(C) = \max_{c \in C} M_L(c) \quad (3) \]

- In the expert advice case, assumes some expert matches the concept (realizable)
Halving Algorithm

\[ H_1 \leftarrow H; \]

\begin{algorithm}
for \( t \leftarrow 1 \ldots T \) do
  Receive \( x_t \);
  \( \hat{y}_t \leftarrow \text{Majority}(H_t, \tilde{a}_t, x_t); \)
  Receive \( y_t \);
  if \( \hat{y}_t \neq y_t \) then
    \[ H_{t+1} \leftarrow \{ a \in H_t : a(x_t) = y_t \}; \]
return \( H_{T+1} \)
\end{algorithm}

Algorithm 1: The Halving Algorithm (Mitchell, 1997)
Halving Algorithm Bound (Littlestone, 1998)

- For a finite hypothesis set

\[ M_{\text{Halving}}(H) \leq \log |H| \]  

(4)

- After each mistake, the hypothesis set is reduced by at least by half
Halving Algorithm Bound (Littlestone, 1998)

- For a finite hypothesis set
  \[ M_{\text{Halving}}(H) \leq \lg |H| \]  
  \[ M_{\text{Halving}}(H) \leq \frac{1}{\log_2 |H|} \]  

- After each mistake, the hypothesis set is reduced by at least by half

- Consider the optimal mistake bound \( \text{opt}(H) \). Then
  \[ \text{VC}(H) \leq \text{opt}(H) \leq M_{\text{Halving}}(H) \leq \lg |H| \]  

- For a fully shattered set, form a binary tree of mistakes with height \( \text{VC}(H) \)
Halving Algorithm Bound (Littlestone, 1998)

- For a finite hypothesis set
  \[
  M_{\text{Halving}}(H) \leq \lg |H| \tag{4}
  \]
- After each mistake, the hypothesis set is reduced by at least by half
- Consider the optimal mistake bound \( \text{opt}(H) \). Then
  \[
  \text{VC}(H) \leq \text{opt}(H) \leq M_{\text{Halving}}(H) \leq \lg |H| \tag{5}
  \]
- For a fully shattered set, form a binary tree of mistakes with height \( \text{VC}(H) \)
- What about non-realizable case?
Experts

Weighted Majority (Littlestone and Warmuth, 1998)

```
for i ← 1...N do
    w_{1,i} ← 1;
for t ← 1...T do
    Receive x_t;
    \hat{y}_t ← 1 \left[ \sum_{a_{t,i}=1} w_t ≥ \sum_{a_{t,i}=0} w_t \right];
    Receive y_t;
    if \hat{y}_t ≠ y_t then
        for i ← 1...N do
            if a_{t,i} \neq y_t then
                w_{t+1,i} ← β w_{t,i};
            else
                w_{t+1,i} ← w_{t,i}
    return w_{T+1}
```

- Weights for every expert
- Classifications in favor of side with higher total weight (y ∈ {0, 1})
- Experts that are wrong get their weights decreased (β ∈ [0, 1])
- If you’re right, you stay unchanged
Weighted Majority (Littlestone and Warmuth, 1998)

\[
\text{for } i \leftarrow 1 \ldots N \text{ do} \\
\quad w_{1,i} \leftarrow 1; \\
\text{for } t \leftarrow 1 \ldots T \text{ do} \\
\quad \text{Receive } x_t; \\
\quad \hat{y}_t \leftarrow 1 \left[ \sum_{a_{t,i}=1} w_t \geq \sum_{a_{t,i}=0} w_t \right]; \\
\quad \text{Receive } y_t; \\
\quad \text{if } \hat{y}_t \neq y_t \text{ then} \\
\quad \quad \text{for } i \leftarrow 1 \ldots N \text{ do} \\
\quad \quad \quad \text{if } a_{t,i} \neq y_t \text{ then} \\
\quad \quad \quad \quad w_{t+1,i} \leftarrow \beta w_{t,i}; \\
\quad \quad \quad \text{else} \\
\quad \quad \quad \quad w_{t+1,i} \leftarrow w_{t,i} \\
\text{return } w_{T+1}
\]

- Weights for every expert
- Classifications in favor of side with higher total weight ($y \in \{0, 1\}$)
- Experts that are wrong get their weights decreased ($\beta \in [0, 1]$)
- If you’re right, you stay unchanged
Experts

Weighted Majority (Littlestone and Warmuth, 1998)

\[
\begin{align*}
&\text{for } i \leftarrow 1 \ldots N \text{ do} \\
&\quad w_{1,i} \leftarrow 1; \\
&\text{for } t \leftarrow 1 \ldots T \text{ do} \\
&\quad \text{Receive } x_t; \\
&\quad \hat{y}_t \leftarrow 1 \left[ \sum_{a_{t,i}=1} w_t \geq \sum_{a_{t,i}=0} w_t \right]; \\
&\quad \text{Receive } y_t; \\
&\quad \text{if } \hat{y}_t \neq y_t \text{ then} \\
&\quad\quad \text{for } i \leftarrow 1 \ldots N \text{ do} \\
&\quad\quad\quad \text{if } a_{t,i} \neq y_t \text{ then} \\
&\quad\quad\quad\quad w_{t+1,i} \leftarrow \beta w_{t,i}; \\
&\quad\quad\quad \text{else} \\
&\quad\quad\quad\quad w_{t+1,i} \leftarrow w_{t,i} \\
&\text{return } w_{T+1}
\end{align*}
\]

- Weights for every expert
- Classifications in favor of side with higher total weight ($y \in \{0, 1\}$)
- Experts that are wrong get their weights decreased ($\beta \in [0, 1]$)
- If you’re right, you stay unchanged
Experts

Weighted Majority (Littlestone and Warmuth, 1998)

\[
\text{for } i \leftarrow 1 \ldots N \text{ do} \\
\quad w_{1,i} \leftarrow 1; \\
\text{for } t \leftarrow 1 \ldots T \text{ do} \\
\quad \text{Receive } x_t; \\
\quad \hat{y}_t \leftarrow 1 \left[ \sum_{a_{t,i} = 1} w_t \geq \sum_{a_{t,i} = 0} w_t \right]; \\
\quad \text{Receive } y_t; \\
\quad \text{if } \hat{y}_t \neq y_t \text{ then} \\
\quad \quad \text{for } i \leftarrow 1 \ldots N \text{ do} \\
\quad \quad \quad \text{if } a_{t,i} \neq y_t \text{ then} \\
\quad \quad \quad \quad w_{t+1,i} \leftarrow \beta w_{t,i}; \\
\quad \quad \quad \text{else} \\
\quad \quad \quad \quad w_{t+1,i} \leftarrow w_{t,i} \\
\text{return } w_{T+1}
\]

- Weights for every expert
- Classifications in favor of side with higher total weight ($y \in \{0, 1\}$)
- Experts that are wrong get their weights decreased ($\beta \in [0, 1]$)
- If you’re right, you stay unchanged
Weighted Majority

- Let $m_t$ be the number of mistakes made by WM until time $t$
- Let $m^*_t$ be the best expert’s mistakes until time $t$
- $N$ is the number of experts

$$m_t \leq \frac{\log N + m^*_t \log \frac{1}{\beta}}{\log \frac{2}{1+\beta}}$$ (6)

- Thus, mistake bound is $O(\log N)$ plus the best expert
- Halving algorithm $\beta = 0$
Proof: Potential Function

- Potential function is the sum of all weights

\[ \Phi_t \equiv \sum_i w_{t,i} \quad (7) \]

- We’ll create sandwich of upper and lower bounds
Proof: Potential Function

- Potential function is the sum of all weights

\[ \Phi_t \equiv \sum_i w_{t,i} \]  

(7)

- We’ll create sandwich of upper and lower bounds
- For any expert \( i \), we have lower bound

\[ \Phi_t \geq w_{t,i} = \beta^{m_t,i} \]  

(8)
Proof: Potential Function

- Potential function is the sum of all weights

\[ \Phi_t \equiv \sum_i w_{t,i} \]  \hspace{1cm} (7)

- We’ll create sandwich of upper and lower bounds
- For any expert \( i \), we have lower bound

\[ \Phi_t \geq w_{t,i} = \beta_{m_t,i} \]  \hspace{1cm} (8)

Weights are nonnegative, so \( \sum_i w_{t,i} \geq w_{t,i} \)
Proof: Potential Function

- Potential function is the sum of all weights

\[ \Phi_t \equiv \sum_i w_{t,i} \]  

(7)

- We’ll create sandwich of upper and lower bounds

- For any expert \( i \), we have lower bound

\[ \Phi_t \geq w_{t,i} = \beta^{m_{t,i}} \]  

(8)

Each error multiplicatively reduces weight by \( \beta \)
Proof: Potential Function (Upper Bound)

- If an algorithm makes an error at round $t$

$$\Phi_{t+1} \leq \frac{\Phi_t}{2} + \frac{\beta \Phi_t}{2}$$  (9)
Proof: Potential Function (Upper Bound)

- If an algorithm makes an error at round $t$

\[ \Phi_{t+1} \leq \frac{\Phi_t}{2} + \frac{\beta \Phi_t}{2} \]  \hspace{1cm} (9)

Half (at most) of the experts by weight were right
Proof: Potential Function (Upper Bound)

- If an algorithm makes an error at round $t$

$$\Phi_{t+1} \leq \frac{\Phi_t}{2} + \frac{\beta \Phi_t}{2}$$  \hspace{1cm} (9)

Half (at least) of the experts by weight were wrong
Proof: Potential Function (Upper Bound)

- If an algorithm makes an error at round \( t \)

\[
\Phi_{t+1} \leq \frac{\Phi_t}{2} + \frac{\beta \Phi_t}{2} = \left[ \frac{1 + \beta}{2} \right] \Phi_t
\]  

(9)
Proof: Potential Function (Upper Bound)

- If an algorithm makes an error at round $t$
  \[
  \Phi_{t+1} \leq \frac{\Phi_t}{2} + \frac{\beta \Phi_t}{2} = \left[ \frac{1 + \beta}{2} \right] \Phi_t
  \] (9)

- Initially potential function sums all weights, which start at 1
  \[
  \Phi_1 = N
  \] (10)
Proof: Potential Function (Upper Bound)

- If an algorithm makes an error at round $t$
  \[ \Phi_{t+1} \leq \frac{\Phi_t}{2} + \frac{\beta \Phi_t}{2} = \left[ \frac{1 + \beta}{2} \right] \Phi_t \]  
  \[ (9) \]

- Initially potential function sums all weights, which start at 1
  \[ \Phi_1 = N \]  
  \[ (10) \]

- After $m_T$ mistakes after $T$ rounds
  \[ \Phi_T \leq \left[ \frac{1 + \beta}{2} \right]^{m_T} N \]  
  \[ (11) \]
Weighted Majority Proof

- Put the two inequalities together, using the best expert

\[ \beta^m \leq \Phi_T \leq \left[ \frac{1 + \beta}{2} \right]^m N \]  

(12)
Weighted Majority Proof

- Put the two inequalities together, using the best expert

\[
\beta m^*_T \leq \Phi_T \leq \left[ \frac{1 + \beta}{2} \right]^{m_T} N
\]  

(12)

- Take the log of both sides

\[
m^*_T \log \beta \leq \log N + m_T \log \left[ \frac{1 + \beta}{2} \right]
\]  

(13)
Weighted Majority Proof

- Put the two inequalities together, using the best expert

$$\beta m^*_T \leq \Phi_T \leq \left[ \frac{1 + \beta}{2} \right]^{m_T} N$$  \hspace{1cm} (12)

- Take the log of both sides

$$m^*_T \log \beta \leq \log N + m_T \log \left[ \frac{1 + \beta}{2} \right]$$  \hspace{1cm} (13)

- Solve for $m_T$

$$m_T \leq \frac{\log N + m^*_T \log \frac{1}{\beta}}{\log \left[ \frac{2}{1 + \beta} \right]}$$  \hspace{1cm} (14)
Weighted Majority Recap

- Simple algorithm
- No harsh assumptions (non-realizable)
- Depends on best learner
- Downside: Takes a long time to do well in worst case (but okay in practice)
- Solution: Randomization