CS281B/Stat241B (Spring 2006) Statistical Learning Theory | Lecture: 12

Ensemble Methods and Adaboost

Lecturer: Peter Bartlett | *Scribe: Ankit Jain*

1 Ensemble methods

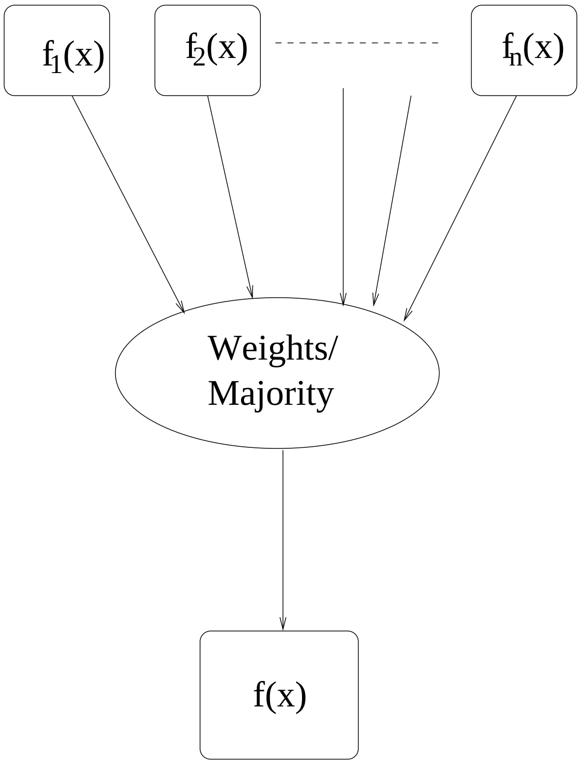

Figure 1: General structure of ensemble methods

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions. Typical example of ensemble methods are boosting, which takes a weighted combination of predictions, and bagging, which takes the majority vote.

2 Example of Classifiers function f_i

(i) ***Decision trees***
In the tree, nodes are labelled with a decision rule such as $x_i \geq \mathcal{Q}$. Leaves are labelled with $y_i \in \pm 1$.

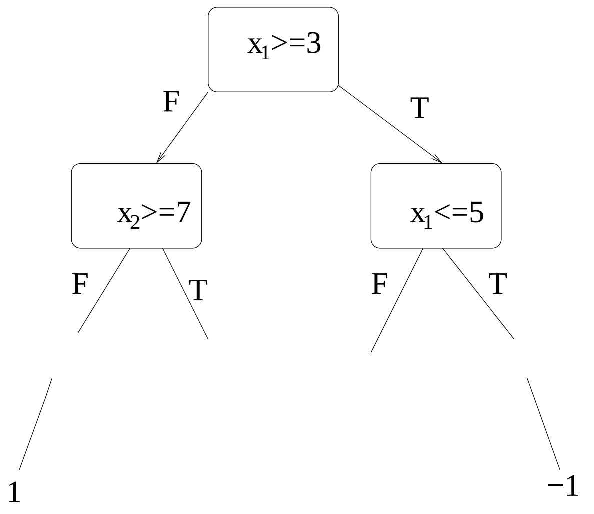

Figure 2: A decision tree

We can also have regression trees i.e. instead of decision stumps we can have more arbitrary decision functions at the nodes. In general, minimizing the empirical risk over the set of decision trees takes time exponential in depth of the tree.

(ii) ***Decision Stumps***
These are trees with 1 node. For example $f(x) = 1_{[x<\theta]}$. For this class we can minimize the empirical risk $\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)$ by sorting along each dimension (which takes time $\mathcal{O}(dn \log n)$ if there are d dimensions).

(iii) ***Linear Threshold Functions***
Here decision boundary is a hyperplane. An example is the function of the form

$$\begin{aligned} f(x) &= \operatorname{sign} \sum_{i=1}^{m} \alpha_i f_i(x_i) \\ &= \operatorname{sign} \sum_{i=1}^{m} \alpha_i \operatorname{sign}\left(\theta_i^T x - \theta_{i_0}\right) \end{aligned}$$

3 Adaboost

The general procedure for Adaboost can be written as follows:

- Maintain weighting D_t over the training data $\{1, \ldots, n\}$.
- Choose f_t sequentially to minimize the D_t-weighted empirical risk.
- Update D_t to D_{t+1} to emphasize the mistakes

In Algorithm 1, z_t is the normalizing factor i.e. z_t is such that $\sum_i D_t(i) = 1$. We choose $z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$. Our choice of z_t is made clear in the next theorem:

Algorithm 1 Adaboost

Initialize $D_0(t) = \frac{1}{n}$ $\quad i = 1, \ldots, n,$ $\quad F_0(x) = 0$ forall x.
for $t = 1, \ldots, T$ **do**
  Choose $f_t \in \mathcal{C}$, where $\mathcal{C}$ is some function class, to minimize: $\epsilon_t = \sum_{i=1}^n D_t(i)\mathbf{1}_{[f_t(x_i)\neq y_i]}$
  $F_t = F_{t-1} + \alpha_t f_t$ where $\alpha_t = 1/2 \log(1-\epsilon_t)/\epsilon_t$.
  $D_{t+1}(i) = \frac{D_t(i)}{z_t}\begin{cases} e^{\alpha_t} & \text{if } f_t(x_i) \neq y_i \\ e^{-\alpha_t} & \text{otherwise} \end{cases}$
end for

Theorem 3.1.

$$\hat{L}(F_T) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{[y_i F_T(x_i)\leq 0]} \leq \prod_{i=1}^T 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

Furthermore:

$$\hat{L}(F_T) \leq (1-4\gamma^2)^{T/2}$$

if $\epsilon_t \leq 1/2 - \gamma$ for all t.

Proof. Note that the last inequality informally means doing better that random guessing.

The idea is similar to Chernoff bounds:

$$y_i F(x_i) \leq 0 \Longrightarrow \exp(-y_i F(x_i)) \geq 1$$

Hence,

$$\begin{aligned}
\hat{L}(F_T) &\leq \frac{1}{n}\sum_{i=1}^n \exp\left(\sum_{i=1}^T -y_i f_t(x_i)\alpha_t\right) \\
&= \frac{1}{n}\sum_{i=1}^n \prod_{t=1}^T \exp\left(-y_i f_t(x_i)\alpha_t\right) \\
&= \frac{1}{n}\sum_{i=1}^n \prod_{t=1}^T \left(z_t \frac{D_{t+1}(i)}{D_t(i)}\right) \\
&= \frac{1}{n}\sum_{i=1}^n \left(\prod_{t=1}^T z_t\right)\frac{D_{T+1}(i)}{D_1(i)}\ .
\end{aligned}$$

In the third step we used:

$$\frac{D_{t+1}(i)}{D_t(i)} = \frac{\exp(-y_i f_i(x_i)\alpha_t)}{z_t}\ .$$

Note that $D_1(i) = \frac{1}{n}$. So if at each step, we choose z_t such that $\sum_i D_{t+1}(i) = 1$, we have

$$\hat{L}(F_T) \leq \prod_{t=1}^T z_t\ .$$

If we choose α_t to minimize the normalization factor we get

$$\alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$$

and

$$z_t = 2\sqrt{(1-\epsilon_t)\epsilon_t}$$

Substituting this we have:

$$\hat{L}(F_T) \leq \prod_{t=1}^{T} 2\sqrt{(1-\epsilon_t)\epsilon_t}$$

The above theorem can be extended to:

Theorem 3.2. For $\bar{F} = \frac{F_T}{\sum_{i=1}^T \alpha_t} = \frac{\sum_{t=1}^T \alpha_t f_t}{\sum_{i=1}^T \alpha_i} \in Co(\mathcal{C})$, we have:

$$\hat{L}(\bar{F}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[y_i \bar{F}(x_i) \leq 0]} \leq \prod_{i=1}^{T} 2\sqrt{\epsilon_t^{1-\gamma}(1-\epsilon_t)^{1+\gamma}}$$

Furthermore if $\epsilon_t \leq 1/2 - \gamma$ for all t, the loss decreases exponentially to zero. That is:
If for all D_t , there exists $f_t \in \mathcal{C}$ with weighted empirical risk less than $1/2-\gamma$, then there exists an $\bar{F} \in co(\mathcal{C})$ with zero empirical risk.

Theorem 3.3. Fix $(x_1, y_1), \ldots, (x_n, y_n)$. If there exists $\bar{F}$ with $y_i \bar{F}(x_i) > \gamma$ for all i then for any probability distribution on $\{1, \ldots, n\}$, there exists $f \in \mathcal{C}$ such that

$$\sum_{i=1}^{n} D(i)\mathbf{1}_{[y_i \neq f(x_i)]} \leq \frac{1-\gamma}{2}$$

For $\bar{F} = \sum \alpha_t f_t$ with $\sum \alpha_t = 1, \alpha_t \geq 0$, choose f randomly with $\mathbb{P}(f = f_t) = \alpha_t$ then:

$$\begin{aligned}
\mathbb{E}\sum_{i=1}^{n} D(i)\mathbf{1}_{[y_i \neq f(x(i)]} &= \sum_t \alpha_t \sum_i D(i)\frac{1-y_i f_t(x_i)}{2} \\
&= \sum_i D(i)\frac{1}{2}\left(1 - \frac{\sum_t \alpha_t f_t(x_i) y_i}{\bar{F}(x_i)}\right) \\
&\leq \sum_i D(i)\frac{1}{2}(1-\gamma) = \frac{1-\gamma}{2}
\end{aligned}$$

So under the random choice of f, $\mathbb{E}\sum D(i)\mathbf{1}_{[y_i \neq f(x_i)]} \leq \frac{1-\gamma}{2}$ and hence there exists an $f \in \mathcal{C}$ with

$$D(i)\mathbf{1}_{[y_i \neq f(x_i)]} \leq \frac{1-\gamma}{2} \ .$$

4 Another interpretation: Gradient Descent

Theorem 4.1. Choosing f_T to minimize $\sum_{i=1}^n D_T(i)\mathbf{1}_{[y_i \neq f_T(x_i)]}$ and $\alpha_T = \frac{1}{2}\ln\frac{1-\epsilon_T}{\epsilon_T}$ is equivalent to choosing α_T, f_T to minimize:

$$\hat{\mathbb{E}}\exp(-YF_T(x)) = \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i(\alpha_T f_T(x_i) + F_{T-1}(x_i)))$$

Proof.

$$\begin{aligned}
\hat{\mathbb{E}}\exp(-YF_T(x)) &= \frac{1}{n}\sum_{i=1}^{n}\left[\left(e^{\alpha_T} - e^{-\alpha_T}\right)\mathbf{1}_{[y_i \neq f_T(x_i)]} + e^{-\alpha_T}\right] e^{-y_i F_{T-1}(x_i)} \\
&= \left(e^{\alpha_T} - e^{-\alpha_T}\right)\left(\prod_{i=1}^{T-1} z_t\right)\sum_{i=1}^{n} D_T(i)\mathbf{1}_{[y_i \neq f_T(x_i)]} + \frac{e^{-\alpha_T}}{n}\sum_{i=1}^{n} e^{-y_i F_{T-1}(x_i)}
\end{aligned}$$

For any $\alpha_T > 0$, the optimal f_T minimizes $\sum_{i=1}^n D_T(i)\mathbf{1}_{[y_i \neq f_T(x_i)]}$. Given f_T, the optimal α_T can be found by solving the first order condition with respect to α_T, i.e.

$$\sum_{i:y_i \neq f_T(x_i)} \frac{1}{n} e^{-y_i F_{T-1}(x_i)} e^{\alpha_T} - \sum_{i:y_i = f_T(x_i)} \frac{1}{n} e^{-y_i F_{T-1}(x_i)} e^{-\alpha_T} = 0$$

$$\Leftrightarrow \left(\epsilon_T e^{\alpha_T} + (1-\epsilon_T) e^{-\alpha_T}\right) \prod_{i=1}^{T-1} z_t = 0 \;\Leftrightarrow\; \alpha_T = \frac{1}{2} \ln \frac{1-\epsilon_T}{\epsilon_T}$$

i.e. we can think of Adaboost as choosing $F \in \text{span}(\mathcal{C})$ to minimize $\frac{1}{n}\sum_{i=1}^n \exp(-y_i F(x_i))$ in a greedy (stepwise) way.

5 A Generalization

We can consider a general optimization problem:

$$\text{minimize } J(F) = \hat{\mathbb{E}}\phi(Y F_t(X)) = \hat{\mathbb{E}}\phi(Y(F_{t-1}(X-1) + \alpha_t f_t(X)))$$

by the gradient descent method. We want to choose a direction $\nu = (\alpha_t f_t(x_1), \ldots, \alpha_t f_t(x_n)) \in \mathbb{R}^n$, to minimize $\nu^T \; \nabla_\nu J(F_{t-1} + \nu)|_{\nu=\mathbf{0}}$. Its easy to see

$$\frac{\partial}{\partial \nu_i} J(F_t)\bigg|_{\nu=\mathbf{0}} = \frac{\partial}{\partial \nu_i} \sum_{i=1}^n \phi(y_i(F_{t-1}(x_i) + \nu_i))\bigg|_{\nu=\mathbf{0}} = \phi'(y_i F_{t-1}(x_i)) y_i.$$

Also:

$$\begin{aligned} & \nu \text{ minimizes } \sum_{i=1}^n \nu_i y_i \phi'(y_i F_{t-1}(x_i)) \\ \Longleftrightarrow \quad & \nu \text{ minimizes } \sum_{i=1}^n (-\nu_i y_i)(-\phi'(y_i F_{t-1}(x_i))) \\ \Longleftrightarrow \quad & \nu \text{ minimizes } \sum_{i=1}^n \mathbf{1}_{[\nu_i \neq y_i]} D_t(i) \text{ for } \nu_i, y_i \in \{\pm 1\} \end{aligned}$$

Hence the general gradient descent algorithm can be written as follows:

Algorithm 2 General gradient Algorithm

Initialize $D_0(t) = \frac{1}{n}$ $\quad i = 1, \ldots, n, \quad F_0(x) = 0$ forall x.
for $t = 1, \ldots, T$ **do**
  Choose $f_t \in \mathcal{C}$, where $\mathcal{C}$ is some function class, to minimize: $\sum_{i=1}^n D_t(i)\mathbf{1}[f_t(x_i) \neq y_i]$
  Choose $\alpha_t \in \mathbb{R}$ to minimize $\hat{E}\phi(Y(\alpha_t f_t(X) + F_{t-1}(X)))$
  $F_t = F_{t-1} + \alpha_t f_t$
  $D_{t+1}(i) = \frac{-\phi'(y_i F_{t-1}(x_i))}{z_t}$
end for
return F_T