BOOSTING

Boosting is an ______________ method.

Unlike bagging or random forests, boosting determines a ______________ majority vote.

In particular, if the class labels are $y = +1, -1$, then boosting determines

$f_1, \ldots, f_T$ ________ classifiers

$\alpha_1, \ldots, \alpha_T > 0$ weights

and returns the ensemble rule

$$h_T(x) =$$

Intuitively, α_t reflects the ______________ in f_t.

Let training data $(x_1, y_1), \ldots, (x_n, y_n)$ be fixed.
Let $\mathcal{F}$ be a fixed set of classifiers

(B) called the ______ class.

<u>Definition</u> A ______ ________ for $\mathcal{F}$ is a rule that takes as input a set of weights $w_1, \ldots, w_n$ $\left(w_i \geq 0, \ \sum_{i=1}^{n} w_i = 1\right)$ and returns a classifier $f \in \mathcal{F}$ such that

is minimized (or at least small)

<u>Notation</u>

A set of weights will be expressed as a vector:

$$w = (w_1, \ldots, w_n)$$

Examples of base classes

- Decision trees

  (D) +

  −

- Decision ________ (trees of depth __)

$$f(x) = \pm \text{ sign} \left\{ x^{(j)} - c \right\}$$

- Radial basis functions

$$f(x) = \pm \text{ sign} \left\{ k_\sigma (x - x_i) - b \right\}$$

where k_σ is a radially symmetric ________ .

Recall the advantages of ensemble methods:

- increased stability (decision trees)
- combine simple classifiers (stumps, RBFs)

Adaboost

→ the first successful boosting algorithm, introduced by Yoav Freund + Robert Schapire.

Adaboost

Given $(x_1, y_1), \ldots, (x_n, y_n)$, $y_i \in \{-1, +1\}$

Initialize $w_i^1 = \frac{1}{n}$.

For $t = 1, \ldots, T$

- Apply base learner with weights w^t to produce classifier f_t

- Set

$$r_t = \sum_{i=1}^{n} w_i^t \, \mathbb{1}\{f_t(x_i) \neq y_i\}$$

- Set

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - r_t}{r_t}\right)$$

- Update

$$w_i^{t+1} = \frac{w_i^t \cdot \exp\{-\alpha_t y_i f_t(x_i)\}}{Z_t}$$

where Z_t is a normalization constant

End

Output

$$h_T(x) = \text{sign}\left\{\sum_{t=1}^{T} \alpha_t f_t(x)\right\}$$

The Boosting Principle

Choose an initial weight vector w^1.

Fix T and a base class $\mathcal{F}$.

For $t = 1:T$

- Given the weight vector w^t, apply the ______ ________ to generate a classifier f_t
- Determine a confidence $\alpha_t > 0$ in f_t
- If $f_t(x_i) \neq y_i$, then w_i^t

  If $f_t(x_i) = y_i$, then w_i^t

End

Output

$h_T(x) =$

Weak learning

The success of Adaboost is reflected in the following result.

Denote $\gamma_t = \frac{1}{2} - r_t$. We may assume $\gamma_t \geq 0$ (why?)

Theorem | The training error of Adaboost satisfies

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{h_T(x_i) \neq y_i\}} \leq \exp\left(-2 \sum_{t=1}^{T} \gamma_t^2\right)$$

In particular, if $\gamma_t \geq \gamma > 0$ for all t, then

(E) $$\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{h_T(x_i) \neq y_i\}} \leq$$

The assumption $\gamma_t \geq \gamma > 0 \quad \forall t$ is called the _____ _______ _________ and in this case the base learner is called a _____ learner.

In words, the theorem tells us that if our base learner does slightly better than _______ ________, the final ensemble classifier can separate the training data perfectly for T large enough. In fact the training error goes to zero

_________ ______.

Remarks

Ⓕ

- If $r_t = 0$, then $\alpha_t =$

  Does this make sense?

- Setting T is a ______ ________

  problem. If T is too large we

  may experience ____________.

  Cross-validation is a common approach.

- Empirical results suggest that

  Adaboost with decision trees is one

  of the best "off-the-shelf"

  methods for classification.

Proof of Theorem The proof is broken down into some lemmas.

Lemma

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{h_T(x_i)\neq y_i\}} \leq \prod_{t=1}^{T} Z_t$$

Proof By unraveling the update rule we find

$$\begin{aligned}
w_i^{T+1} &= \frac{w_i^{T}\exp\left(-\alpha_T y_i f_T(x_i)\right)}{Z_T} \\
&= \frac{w_i^{T-1}\exp\left(-y_i\left[\alpha_{T-1}f_{T-1}(x_i)+\alpha_T f_T(x_i)\right]\right)}{Z_{T-1}\cdot Z_T} \\
&\;\;\vdots \\
&= \frac{\frac{1}{n}\cdot\exp\left(-y_i\sum_{t=1}^{T}\alpha_t f_t(x_i)\right)}{Z_1\cdot Z_2\cdots Z_T} \\
&= \frac{\exp\left(-y_i F_T(x_i)\right)}{n\prod_{t=1}^{T} Z_t}
\end{aligned}$$

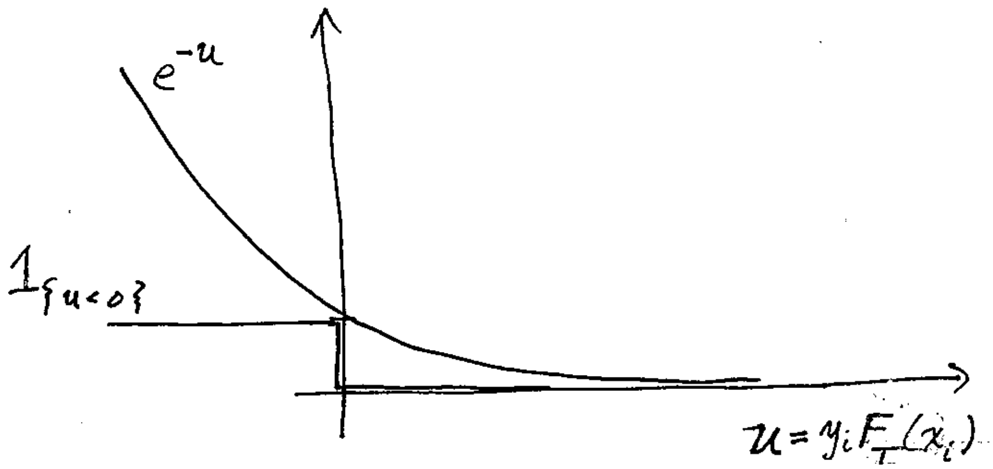
$\boxed{w_i^1}$

where $F_t = \sum_{s=1}^{t}\alpha_s f_s$

and $h_T(x) = \text{sign}\left\{F_T(x)\right\}$

Now use the bound

$$\mathbb{1}_{\{h_T(x_i) \neq y_i\}} = \mathbb{1}_{\{y_i F_T(x_i) < 0\}} \leq \exp\left(-y_i F_T(x_i)\right)$$



Then

$$\begin{aligned} 1 &= \sum_{i=1}^{n} w_i^{T+1} \\ &= \sum_{i=1}^{n} \frac{\exp\left(-y_i F_T(x_i)\right)}{n \cdot (\Pi Z_t)} \\ &\geq \frac{1}{(\Pi Z_t)} \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{h_T(x_i) \neq y_i\}} \end{aligned}$$

and the lemma follows. ∎

Lemma $Z_t = \sqrt{1-4\gamma_t^2}$

Proof $Z_t = \sum_{i=1}^{n} \underbrace{w_i^t \exp\left(-\alpha_t y_i f_t(x_i)\right)}_{w_i^{t+1}}$

$$= \sum_{i: f_t(x_i)=y_i} w_i^t \exp(-\alpha_t) + \sum_{i: f_t(x_i)\neq y_i} w_i^t \exp(\alpha_t)$$

$$= (1-r_t)e^{-\alpha_t} + r_t e^{\alpha_t}$$

Now recall

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-r_t}{r_t}\right) = \ln\sqrt{\frac{1-r_t}{r_t}}$$

Then

$$Z_t = (1-r_t)\sqrt{\frac{r_t}{1-r_t}} + r_t\sqrt{\frac{1-r_t}{r_t}}$$

$$= 2\sqrt{r_t(1-r_t)}$$

Now substitute

$$r_t = \frac{1}{2} - \gamma_t$$

$$\Rightarrow \quad Z_t = 2\sqrt{\left(\tfrac{1}{2}-\gamma_t\right)\left(\tfrac{1}{2}+\gamma_t\right)}$$

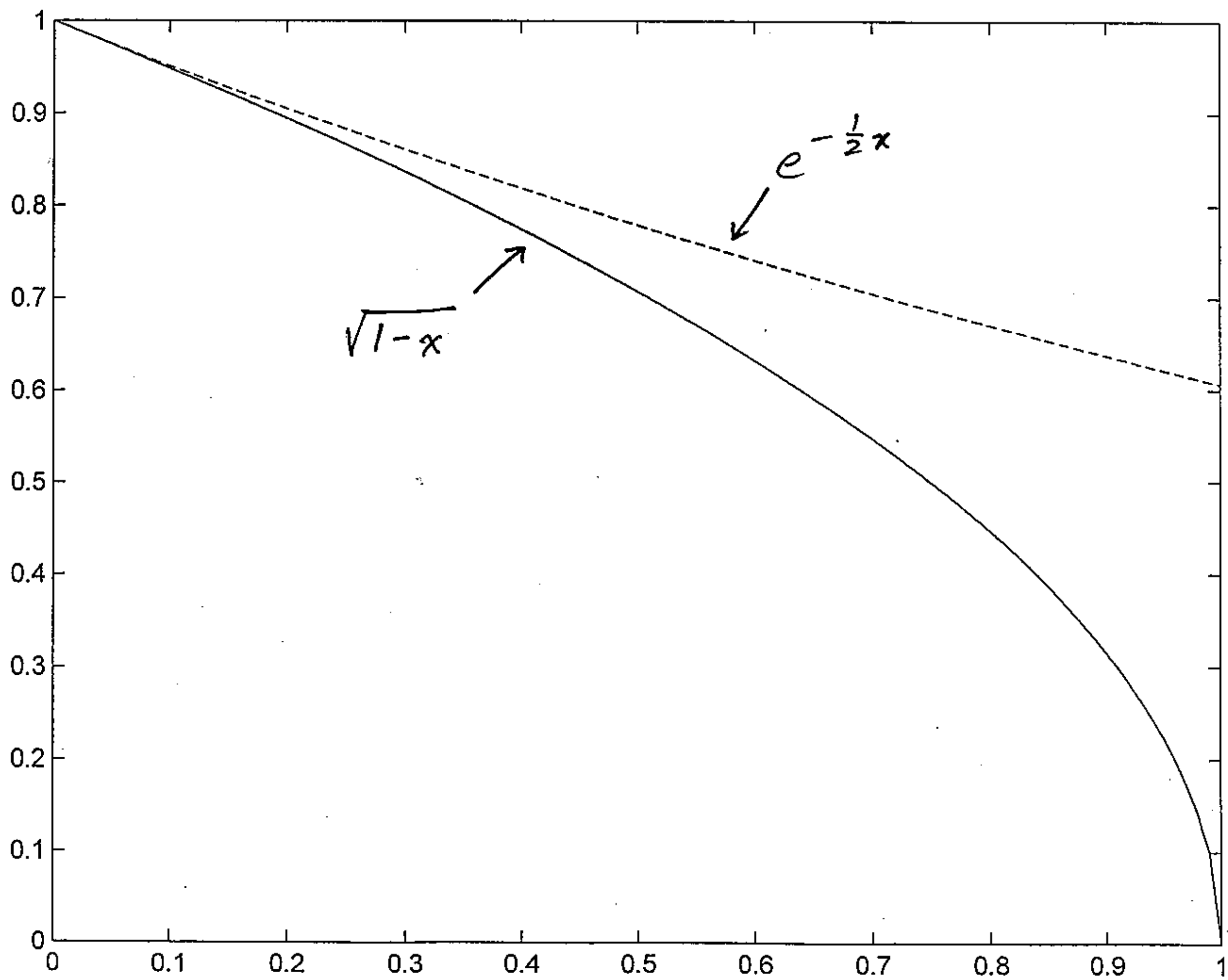
$$= 2\sqrt{\tfrac{1}{4}-\gamma_t^2}$$

$$= \sqrt{1-4\gamma_t^2} \qquad \blacksquare$$

Lemma

$$\sqrt{1-x} \le e^{-\frac{1}{2}x}$$

Proof



Formally, $\sqrt{1-x}$ is concave, $e^{-\frac{1}{2}x}$ is convex, so it suffices to show their slopes (derivatives) are both $= -\frac{1}{2}$ at 0. $\blacksquare$

Putting it all together, we obtain

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} I_{\{h_T(x_i) \neq y_i\}} &\leq \frac{1}{n}\sum_{i=1}^{n} \exp\left(-y_i F_T(x_i)\right) \\
&= \prod_{t=1}^{T} Z_t \\
&= \prod_{t=1}^{T} \sqrt{1-4\gamma_t^2} \\
&\leq e^{-2\sum_{t=1}^{T}\gamma_t^2} \qquad \blacksquare
\end{aligned}
$$

Exercise: View Z_t as a function of α_t, and find the value of α_t that minimizes Z_t.

Solution | Earlier we showed

$$Z_t = (1-r_t)e^{-\alpha_t} + r_t e^{\alpha_t}.$$

This is a convex, differentiable function of α_t.

It is minimized by setting

$$0 = \frac{\partial Z_t}{\partial \alpha_t} = -(1-r_t)e^{-\alpha_t} + r_t e^{\alpha_t}$$

$$\Rightarrow \quad e^{2\alpha_t} = \frac{1-r_t}{r_t}$$

$$\Rightarrow \quad \alpha_t = \frac{1}{2}\ln\left(\frac{1-r_t}{r_t}\right)$$

In conclusion, each α_t is choosen to minimize the corresponding term Z_t in the bound $\prod_{t=1}^{T} Z_t$.

That is, the bound is minimized incrementally (not globally)

Boosting as Functional Gradient Descent

We will now generalize Adaboost to an entire class of boosting algorithms.

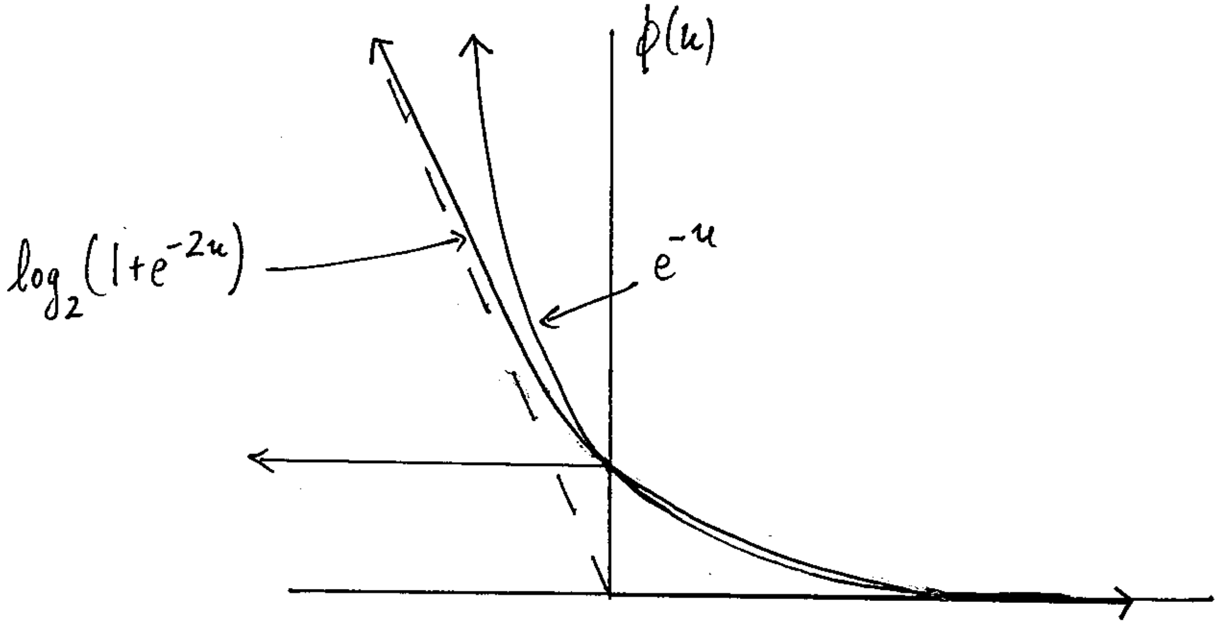
Recall that in bounding the Adaboost training error we used the bound

(G) $$\mathbb{1}_{\{u<0\}} \leq \qquad , \quad u = y_i F_t(x_i)$$

We will generalize Adaboost by using the bound

$$\mathbb{1}_{\{u<0\}} \leq$$

where $\phi(u)$ is called a ______ function.



For computational reasons, the loss is often choosen to be

(H) •

•

•

Examples

- exponential
- logistic
- hinge (not differentiable, decreasing)
- squared error (not decreasing)

Why use different losses?

The logistic loss, for example, doesn't penalize misclassified points as severely, and therefore may be less susceptible to ________.

Let's assume that ϕ is convex, differentiable, and decreasing.

On the t^{th} iteration of boosting we have the ensemble

$$F_t(x) = \sum_{s=1}^{t} \alpha_s f_s(x)$$

and the bound

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{y_i F_t(x_i) < 0\}} \le \frac{1}{n}\sum_{i=1}^{n} \phi(y_i F_t(x_i))$$

View this bound as an objective function to be minimized with respect to F_t.

Boosting may be viewed as <u>functional gradient descent</u> applied to the bound.

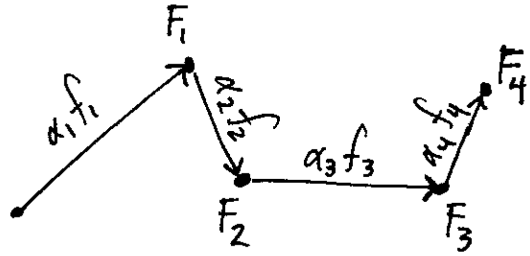
In particular, suppose $\alpha_1, f_1, \ldots, \alpha_{t-1}, f_{t-1}$ are given, and set

$$B_t(\alpha, f) = \frac{1}{n}\sum_{i=1}^{n} \phi\big(y_i F_{t-1}(x_i) + y_i \alpha f(x_i)\big)$$

Then set

1) f_t = function $f \in \mathcal{F}$ (base class) for which the directional derivative of B_t in the direction f is minimized.

2) α_t = stepsize $\alpha > 0$ in the direction f_t for which $B(\alpha, f_t)$ is minimized.



Step 1 | The directional derivative of B_t in the direction f is

(I) $$\left.\frac{\partial B_t(\alpha, f)}{\partial \alpha}\right|_{\alpha=0} =$$

Minimizing this is equivalent to minimizing

$$- \sum_{i=1}^{n} y_i f(x_i) \cdot \underbrace{\frac{\phi'(y_i F_{t-1}(x_i))}{\sum_{j=1}^{n} \phi'(y_j F_{t-1}(x_j))}}_{w_i^t}$$

$\boxed{\text{since } \phi' < 0}$

$$= \sum_{i=1}^{n} w_i^t \mathbb{1}_{\{f(x_i) \neq y_i\}} - \sum_{i=1}^{n} w_i^t \mathbb{1}_{\{f(x_i) = y_i\}}$$

$$= 2\left(\sum_i w_i^t \mathbb{1}_{\{f(x_i) \neq y_i\}} \right) - 1$$

To minimize this expression with respect to f,

J we can use the ______ ________.

Step 2

$$\alpha_t := \arg\min_{\alpha} B_t(\alpha, f_t)$$

$$= \arg\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} \phi\left(y_i F_{t-1}(x_i) + y_i \alpha f_t(x_i) \right)$$

Generalized Boosting Algorithm

Given $(x_1,y_1),\dots,(x_n,y_n)$, $y_i \in \{-1,1\}$, convex loss ϕ

Initialize $w_i^1 = \frac{1}{n}$

For $t = 1,\dots,T$

- Apply base learner with weights w^t to produce classifier f_t
- Set

$$\alpha_t = \arg\min_{\alpha} \frac{1}{n}\sum_{i=1}^{n} \phi\left(y_i F_{t-1}(x_i) + y_i \alpha f_t(x_i)\right)$$

- Update

$$w_i^{t+1} = \frac{\phi'(y_i F_t(x_i))}{\sum_{j=1}^{n} \phi'(y_j F_t(x_j))}$$

End

Output

$$h_T(x) = \operatorname{sign}\{F_T(x)\} = \operatorname{sign}\left\{\sum_{t=1}^{T} \alpha_t f_t(x)\right\}$$

Since ϕ is convex, α is the solution of a convex, univariate optimization problem and can be found efficiently using Newton's method.

When $\phi(u) = e^{-u}$, the algorithm simplifies to Adaboost. In this case

(K) • w_i^t has a nice ______________ formula since

$$\phi'(a+b) = \phi'(a) \cdot \phi'(b)$$

• α_t has a closed form solution.

When $\phi(u) = \log_2(1 + e^{-2u})$, the algorithm is called __________________. For computational efficiency, Friedman, Hastie, and Tibshirani suggest using only one step of Newton's method at each round.

(L) Why did we assume ϕ to be decreasing?

Key

A. ensemble, weighted, base

$h_T(x) = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t f_t(x) \right\}$, confidence

B. base, base learner, $\sum_{i=1}^{n} w_i \mathbb{1}_{\{f(x_i) \neq y_i\}}$

C. base learner, increase, decrease

$h_T(x) = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t f_t(x) \right\}$

D. +: increased stability, performance, −: lose interpretability

stumps, 1, kernel

E. $\exp(-2\gamma^2 T)$ F. weak learning hypothesis,

weak, random guessing, exponentially fast

F. ∞, model selection, overfitting

G. e^{-u}, $\varphi(u)$, loss

H. convex, differentiable, decreasing; outliers

I. $\frac{1}{n} \sum_{i=1}^{n} y_i f(x_i) \cdot \varphi'(y_i F_{t-1}(x_i))$ J. base learner

K. recursive, Logitboost

L. So that $\varphi' < 0$ (Step 1)