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

$$f_1, \ldots, f_T \quad \text{classifiers}$$

$$\alpha_1, \ldots, \alpha_T > 0 \quad \text{weights}$$

and returns the ensemble rule

$$h_T(x) =$$

Intuitively, $\alpha_t$ reflects the _______ in $f_t$. 

Let training data \((x_i, y_i), \ldots, (x_n, y_n)\) be fixed. Let \(F\) be a fixed set of classifiers called the \(\underline{\text{---}}\) class.

**Definition**  
A \(\underline{\text{---}}\) for \(F\) is a rule that takes as input a set of weights \(w_1, \ldots, w_n\) \((w_i \geq 0, \sum_{i=1}^{n} w_i = 1)\) and returns a classifier \(f \in F\) such that

is minimized (or at least small)

**Notation**  
A set of weights will be expressed as a vector:

\[
w = (w_1, \ldots, w_n)
\]
Examples of base classes

- Decision trees
  \[ f(x) = \pm \text{sign} \left\{ \sum \chi^{(i)} - c \right\} \]

- Decision (trees of depth \_)

- Radial basis functions
  \[ f(x) = \pm \text{sign} \left\{ \sum k_\sigma(x-x_i) - b \right\} \]
  where \( k_\sigma \) 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.
Given \((x_i, y_i), \ldots, (x_n, y_n)\), \(y_i \in \{-1, +1\}\)

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 \(r_t = \sum_{i=1}^{n} w_i^t \cdot I[\{f^t(x_i) \neq y_i\}]\)

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

- Update

\[ w_{i}^{t+1} = \frac{w_i^t \cdot \exp \left\{ -\alpha_t y_i f^t(x_i) \right\}}{Z_t} \]

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 $w^i$.

Fix $T$ and a base class $F$.

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

  $w_i^{t-1} \leftarrow \frac{w_i^{t-1}}{\exp(\alpha_t)}$  

  $\frac{1}{\exp(\alpha_t)}$

- If $f_t(x_i) = y_i$, then

  $w_i^{t-1} \leftarrow \frac{w_i^{t-1}}{\exp(\alpha_t)}$  

End

Output

$h_T(x) =$
Weak learning

The success of Adaboost is reflected in the following result.

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

Theorem | The training error of Adaboost satisfies

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} I_{h_t(x_i) \neq y_i} \leq \exp \left( -2 \sum_{t=1}^{T} \delta_t^2 \right)$$

In particular, if $\delta_t \geq \delta > 0$ for all $t$, then

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} I_{h_T(x_i) \neq y_i} \leq$$
The assumption $\delta_t \geq \delta > 0 \forall t$ 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 \( \alpha_t = \ldots \).
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

\[
\frac{1}{n} \sum_{i=1}^{n} I \{ h_1(x_i) \neq y_i \} \leq \frac{T}{T} \sum_{t=1}^{T} Z_t
\]

Proof

By unraveling the update rule we find

\[
W_{i+1}^{T+1} = \frac{W_i^T \exp \left( -\alpha_T y_i \tilde{f}_T(x_i) \right)}{Z_T}
\]

\[
= \frac{W_{i-1}^{T-1} \exp \left( -y_i \left[ \alpha_{T-1} \tilde{f}_{T-1}(x_i) + \alpha_T \tilde{f}_T(x_i) \right] \right)}{Z_{T-1} \cdot Z_T}
\]

\[
\vdots
\]

\[
= \frac{1}{n} \cdot \exp \left( -y_i \sum_{t=1}^{T} \alpha_t \tilde{f}_t(x_i) \right)
\]

\[
\frac{Z_1 \cdot Z_2 \cdots Z_T}{Z_1 \cdot Z_2 \cdots Z_T}
\]

\[
= \frac{\exp \left( -y_i \tilde{F}_T(x_i) \right)}{n \sum_{t=1}^{T} Z_t}
\]

where

\[
\tilde{F}_T = \sum_{s=1}^{t} \alpha_s \tilde{f}_s
\]

and

\[
h_t(x) = \text{sign} \{ F_T(x) \}
\]
Now use the bound

\[ \ell_{\hat{h}_T(x_i) \neq y_i} = \ell_{\frac{y_i F_T(x_i)}{\log(1 + \exp(-y_i F_T(x_i)))} < 0} \leq \exp(-y_i F_T(x_i)) \]

Then

\[ 1 = \sum_{i=1}^{n} w_i^{T+1} \]

\[ = \sum_{i=1}^{n} \frac{\exp(-y_i F_T(x_i))}{n \cdot (\prod Z_t)} \]

\[ \geq \frac{1}{(\prod Z_t)} \cdot \frac{1}{n} \sum_{i=1}^{n} \ell_{\hat{h}_T(x_i) \neq y_i} \]

and the lemma follows.
\[ \text{Lemma} \quad Z_t = \sqrt{1 - \frac{4}{\sigma_t^2}} \]

**Proof**

\[ Z_t = \sum_{i=1}^{n} \frac{w_i^t \exp(-\alpha_t y_i f_t(x_i))}{w_{i,t+1}} \]

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

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

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

\[ r_t = \frac{1}{2} - \delta_t \]
\[ \Rightarrow z_t = 2 \sqrt{(\frac{1}{2} - \delta_t)(\frac{1}{2} + \delta_t)} \]

\[ = 2 \sqrt{\frac{1}{4} - \delta_t^2} \]

\[ = \sqrt{1 - 4\delta_t^2} \]

**Lemma**

\[ \sqrt{1 - x} \leq e^{-\frac{1}{2}x} \]

**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 \( 0 \).

Putting it all together, we obtain

\[
\frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{1 - y_i \hat{f}_t(x_i)^2}} \leq \frac{1}{n} \sum_{i=1}^{n} \exp \left(-y_i \hat{f}_t(x_i)\right) \\
= \prod_{t=1}^{T} Z_t \\
= \prod_{t=1}^{T} \sqrt{1 - 4\gamma_t^2} \\
\leq e^{-2 \sum_{t=1}^{T} \gamma_t^2}
\]

Exercise \[ \text{View } Z_t \text{ as a function of } \gamma_t, \text{ and find the value of } \gamma_t \text{ that minimizes } Z_t. \]
Earlier we showed

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

This is a convex, differentiable function of \( \alpha_t \).

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 \( \alpha_t \) is chosen to minimize the corresponding term \( Z_t \) in the bound \( \sum_{t=1}^{T} Z_t \).

That is, the bound is minimized \underline{incrementally} (not \underline{globally}).
We will now generalize Adaboost to an entire class of boosting algorithms. Recall that in bounding the Adaboost training error we used the bound

\[ \frac{1}{\xi u < 0} \leq \frac{1}{\xi u < 0} \leq \]

We will generalize Adaboost by using the bound

\[ \frac{1}{\xi u < 0} \leq \]

where \( \phi(u) \) is called a ______ function.
For computational reasons, the loss is often chosen to be

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

Examples:

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 $f$ is convex, differentiable, and decreasing.
On the $t^\text{th}$ iteration of boosting we have the ensemble

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

and the bound

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{n} \Phi(y_i, F_t(x_i)) \leq \frac{1}{n} \sum_{i=1}^{n} \Phi(y_i, F_t(x_i))$$

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

Boosting may be viewed as \underline{functional gradient descent} applied to the bound.

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

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

\[
\frac{\partial B_t(\alpha, f)}{\partial \alpha} \bigg|_{\alpha_0}
\]
Minimizing this is equivalent to minimizing

\[- \sum_{i=1}^{n} y_i f(x_i) \cdot \frac{\phi'(y_i f_{t-1}(x_i))}{\sum_{j=1}^{n} \phi'(y_j f_{t-1}(x_j))} \]

since \( \phi' < 0 \)

\[w_t \]

\[= \sum_{i=1}^{n} w_t \cdot 1_{\{ f(x_i) \neq y_i \}} - \sum_{i=1}^{n} w_t \cdot 1_{\{ f(x_i) = y_i \}} \]

\[= 2 \left( \sum_{i} w_t \cdot 1_{\{ f(x_i) \neq y_i \}} \right) - 1 \]

To minimize this expression with respect to \( f \),

we can use the

\[\text{Step 2:} \quad \alpha_t := \arg \min_{\alpha} B_t(\alpha, f_t) \]

\[= \arg \min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} \phi\left( y_i f_{t-1}(x_i) + y_i \alpha f_t(x_i) \right) \]
Given \((x_i, y_i), \ldots, (x_n, y_n)\), \(y_i \in \{-1, 1\}\), convex loss \(\phi\)

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

- 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

\[h_T(x) = \text{sign} \left\{ \sum_{t=1}^{T} \alpha_t f_t(x) \right\} = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t f_t(x) \right\}\]
Since $\phi$ is convex, $x$ 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 $W_t$ has a nice formula since

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

$x_t$ 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.

Why did we assume $\phi$ to be decreasing?
A. ensemble, weighted, base  
\[ h_T(x) = \text{sign} \left\{ \sum_{t=1}^{T} \alpha_t f_t(x) \right\}, \text{ confidence} \]

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

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 \((-2\gamma^2 T)\)  
weak learning hypothesis, weak, random guessing, exponentially fast

F. \(\infty\), model selection, overfitting

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

H. convex, differentiable, decreasing; outliers

I. \[ \frac{1}{h} \sum_{i=1}^{n} y_i f(x_i) - \phi'(y_i F_{T-1}(x_i)) \]

J. base learner

K. recursive, Logitboost

L. so that \(\phi' < 0\) (step 1)