Árboles (CART), Random Forests, Boosting y Kernels

Universidad de los Andes y Quantil

Junio 2020

Contenido

Árboles de clasificación y regresión (CART)

- En este caso el mejor clasificador (regresor) se expresa de la forma:

$$\widehat{f}(x) = \sum_{m=1}^{M} c_m I\{x \in R_m\} \tag{1}$$

donde R_m son diferentes regiones en las cuales la función se va aproximar por una contante c_m.

- Algunas regiones son difíciles de describir (panel izquierdo arriba siguiente gráfica).
- Una alternativa es encontrar regiones haciendo separaciones binarias secuenciales.

Árboles de clasificación y regresión (CART)

- En este caso el mejor clasificador (regresor) se expresa de la forma:

$$\widehat{f}(x) = \sum_{m=1}^{M} c_m I\{x \in R_m\} \tag{1}$$

donde R_m son diferentes regiones en las cuales la función se va aproximar por una contante c_m.

- Algunas regiones son difíciles de describir (panel izquierdo arriba siguiente gráfica).
- Una alternativa es encontrar regiones haciendo separaciones binarias secuenciales.

Árboles de clasificación y regresión (CART)

- En este caso el mejor clasificador (regresor) se expresa de la forma:

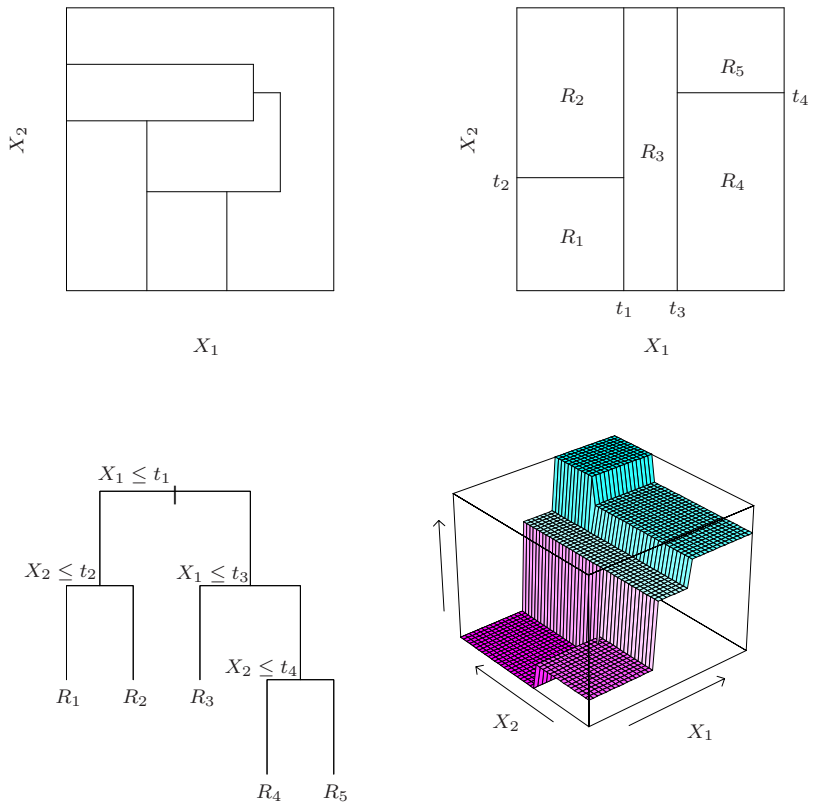
$$\widehat{f}(x) = \sum_{m=1}^{M} c_m I\{x \in R_m\} \tag{1}$$

donde R_m son diferentes regiones en las cuales la función se va aproximar por una contante c_m.

- Algunas regiones son difíciles de describir (panel izquierdo arriba siguiente gráfica).
- Una alternativa es encontrar regiones haciendo separaciones binarias secuenciales.

Árboles de clasificación y regresión (CART)

- Los paneles 2,3,4 representan regiones obtenidas mediante separaciones binarias secuenciales.



Contenido

Árboles de regresión

- Consideramos primero el caso en el que la variable objetivo es continua (problema de regresión).
- Si la función de pérdida es cuadrática las mejores constantes en cada región son el promedio.

$$\widehat{c}_m = media\{y_i \,|x_i \in R_m\} \tag{2}$$

Árboles de regresión

- Consideramos primero el caso en el que la variable objetivo es continua (problema de regresión).
- Si la función de pérdida es cuadrática las mejores constantes en cada región son el promedio.

$$\widehat{c}_m = media\{y_i \,|x_i \in R_m\} \tag{2}$$

Árboles de regresión

- Para definir las regiones hacemos lo siguiente. Sea j una variable de separación y s el punto de corte. Definimos las dos regiones:

$$R_1(j, s) = \{X \,|x_j \leq s\} \tag{3}$$

$$R_2(j, s) = \{X \,|x_j > s\} \tag{4}$$

Árboles de regresión

- Ahora resolvemos el siguiente problema:

$$min_{j,s}(min_{c_1} \sum_{x \in R_1(j,s)} (y_j - c_1)^2 + min_{c_2} \sum_{x \in R_2(j,s)} (y_j - c_2)^2) \tag{5}$$

- Para cada elección de j, s, los valores c_1, c_2 que minimizan el error son los promedios en cada región $R_1(j, s), R_2(j, s)$, respectivamente.
- Una vez resuelto este problema, con j, s, c_1, c_2, se repite el proceso en cada region $R_1(j, s), R_2(j, s)$.
- Ahora la pregunta fundamental es qué tanto crecemos el árbol? El tamaño del árbol es una medida de la complejidad del mismo.

Árboles de regresión

- Ahora resolvemos el siguiente problema:

$$min_{j,s}(min_{c_1} \sum_{x \in R_1(j,s)} (y_j - c_1)^2 + min_{c_2} \sum_{x \in R_2(j,s)} (y_j - c_2)^2) \tag{5}$$

- Para cada elección de j, s, los valores c_1, c_2 que minimizan el error son los promedios en cada región $R_1(j,s), R_2(j,s)$, respectivamente.
- Una vez resuelto este problema, con j, s, c_1, c_2, se repite el proceso en cada region $R_1(j,s), R_2(j,s)$.
- Ahora la pregunta fundamental es qué tanto crecemos el árbol? El tamaño del árbol es una medida de la complejidad del mismo.

Árboles de regresión

- Ahora resolvemos el siguiente problema:

$$min_{j,s}(min_{c_1} \sum_{x \in R_1(j,s)} (y_j - c_1)^2 + min_{c_2} \sum_{x \in R_2(j,s)} (y_j - c_2)^2) \tag{5}$$

- Para cada elección de j, s, los valores c_1, c_2 que minimizan el error son los promedios en cada región $R_1(j, s), R_2(j, s)$, respectivamente.
- Una vez resuelto este problema, con j, s, c_1, c_2, se repite el proceso en cada region $R_1(j, s), R_2(j, s)$.
- Ahora la pregunta fundamental es qué tanto crecemos el árbol? El tamaño del árbol es una medida de la complejidad del mismo.

Árboles de regresión

- Ahora resolvemos el siguiente problema:

$$min_{j,s}(min_{c_1} \sum_{x \in R_1(j,s)} (y_j - c_1)^2 + min_{c_2} \sum_{x \in R_2(j,s)} (y_j - c_2)^2) \tag{5}$$

- Para cada elección de j, s, los valores c_1, c_2 que minimizan el error son los promedios en cada región $R_1(j, s), R_2(j, s)$, respectivamente.
- Una vez resuelto este problema, con j, s, c_1, c_2, se repite el proceso en cada region $R_1(j, s), R_2(j, s)$.
- Ahora la pregunta fundamental es qué tanto crecemos el árbol? El tamaño del árbol es una medida de la complejidad del mismo.

Árboles de regresión

- Una alternativa sería solo permitir ramificaciones cuando la reducción en el error supera cierto umbral. Sin embargo esta estrategia puede no ser muy buena en la medida que ramificaciones posteriores pueden reducir sustancialmente el error.
- Otra alternativa es crecer el árbol hasta que cada nodo terminal tenga menos de cierto umbral de ejemplos de la muestra.
- Podar:
  1. Cortar ramas y cambiar hojas con las observaciones de todos los nodos terminales que se cortan.
  2. El tamaño del árbol se determina podándolo con el objetivo de minimizar una medida del costo de la complejidad (*cost complexity pruning*).

Árboles de regresión

- Una alternativa sería solo permitir ramificaciones cuando la reducción en el error supera cierto umbral. Sin embargo esta estrategia puede no ser muy buena en la medida que ramificaciones posteriores pueden reducir sustancialmente el error.
- Otra alternativa es crecer el árbol hasta que cada nodo terminal tenga menos de cierto umbral de ejemplos de la muestra.
- Podar:
  1. Cortar ramas y cambiar hojas con las observaciones de todos los nodos terminales que se cortan.
  2. El tamaño del árbol se determina podándolo con el objetivo de minimizar una medida del costo de la complejidad (*cost complexity pruning*).

Árboles de regresión

- Una alternativa sería solo permitir ramificaciones cuando la reducción en el error supera cierto umbral. Sin embargo esta estrategia puede no ser muy buena en la medida que ramificaciones posteriores pueden reducir sustancialmente el error.
- Otra alternativa es crecer el árbol hasta que cada nodo terminal tenga menos de cierto umbral de ejemplos de la muestra.
- Podar:
  1. Cortar ramas y cambiar hojas con las observaciones de todos los nodos terminales que se cortan.
  2. El tamaño del árbol se determina podándolo con el objetivo de minimizar una medida del costo de la complejidad (*cost complexity pruning*).

Árboles de regresión

- Una alternativa sería solo permitir ramificaciones cuando la reducción en el error supera cierto umbral. Sin embargo esta estrategia puede no ser muy buena en la medida que ramificaciones posteriores pueden reducir sustancialmente el error.
- Otra alternativa es crecer el árbol hasta que cada nodo terminal tenga menos de cierto umbral de ejemplos de la muestra.
- Podar:
  1. Cortar ramas y cambiar hojas con las observaciones de todos los nodos terminales que se cortan.
  2. El tamaño del árbol se determina podándolo con el objetivo de minimizar una medida del costo de la complejidad (*cost complexity pruning*).

Árboles de regresión

- Sea T un árbol, indexemos por m los nodos terminales del árbol y sea $|T|$ el número de nodos terminales en T.
- Sea N_m el número de observaciones tal que $x \in R_m$.
- $\widehat{c}_m$ el promedio de las variables objetivo en la región R_m.
- Sea $Q_m(T)$ el error promedio cuadrático entre la variable objetivo y el promedio en la región.
- El costo de la complejidad se define como:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{6}$$

Árboles de regresión

- Sea T un árbol, indexemos por m los nodos terminales del árbol y sea $|T|$ el número de nodos terminales en T.
- Sea N_m el número de observaciones tal que $x \in R_m$.
- $\widehat{c}_m$ el promedio de las variables objetivo en la región R_m.
- Sea $Q_m(T)$ el error promedio cuadrático entre la variable objetivo y el promedio en la región.
- El costo de la complejidad se define como:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{6}$$

Árboles de regresión

- Sea T un árbol, indexemos por m los nodos terminales del árbol y sea $|T|$ el número de nodos terminales en T.
- Sea N_m el número de observaciones tal que $x \in R_m$.
- $\widehat{c}_m$ el promedio de las variables objetivo en la región R_m.
- Sea $Q_m(T)$ el error promedio cuadrático entre la variable objetivo y el promedio en la región.
- El costo de la complejidad se define como:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{6}$$

Árboles de regresión

- Sea T un árbol, indexemos por m los nodos terminales del árbol y sea $|T|$ el número de nodos terminales en T.
- Sea N_m el número de observaciones tal que $x \in R_m$.
- $\widehat{c}_m$ el promedio de las variables objetivo en la región R_m.
- Sea $Q_m(T)$ el error promedio cuadrático entre la variable objetivo y el promedio en la región.
- El costo de la complejidad se define como:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{6}$$

Árboles de regresión

- Sea T un árbol, indexemos por m los nodos terminales del árbol y sea $|T|$ el número de nodos terminales en T.
- Sea N_m el número de observaciones tal que $x \in R_m$.
- $\widehat{c}_m$ el promedio de las variables objetivo en la región R_m.
- Sea $Q_m(T)$ el error promedio cuadrático entre la variable objetivo y el promedio en la región.
- El costo de la complejidad se define como:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{6}$$

Árboles de regresión

- El problema que queremos resolver es:

$$min_{T' \subset T} C_\alpha(T')$$

donde T' es cualquier sub árbol que se puede obtener de T podándolo.

- Valores altos de α resulta en árboles más pequeños. Si es cero el resultado es el árbol T.
- El parámetro α se puede determinar usando validación cruzada. Por ejemplo el valor de α que reduzca el error promedio, M-fold, de validación cruzada.

Árboles de regresión

- El problema que queremos resolver es:

$$min_{T' \subset T} C_\alpha(T')$$

  donde T' es cualquier sub árbol que se puede obtener de T podándolo.
- Valores altos de α resulta en árboles más pequeños. Si es cero el resultado es el árbol T.
- El parámetro α se puede determinar usando validación cruzada. Por ejemplo el valor de α que reduzca el error promedio, M-fold, de validación cruzada.

Árboles de regresión

- El problema que queremos resolver es:

$$min_{T' \subset T} C_\alpha(T')$$

  donde T' es cualquier sub árbol que se puede obtener de T podándolo.
- Valores altos de α resulta en árboles más pequeños. Si es cero el resultado es el árbol T.
- El parámetro α se puede determinar usando validación cruzada. Por ejemplo el valor de α que reduzca el error promedio, M-fold, de validación cruzada.

Contenido

- La única modificación necesaria es definir la función de error en cada ramificación: $Q_m(T)$ (i.e. medidad de impureza).
- Sean $R_1, ..., R_M$ un conjunto de regiones.
- Sea $p(m, k)$ La cantidad de observaciones de la clase k en la región m como proporción de la cantidad de observaciones en esa región.
- Sea $k(m)$ la clase mayoritaria en la región m.
- Definimos el error de clasificación como $L(m) = 1 - p(m, k(m))$.
- Ahora para crecer y podar el árbol se sigue el mismo algoritmo de árboles de regresión.

- La única modificación necesaria es definir la función de error en cada ramificación: $Q_m(T)$ (i.e. medidad de impureza).
- Sean $R_1, ..., R_M$ un conjunto de regiones.
- Sea $p(m, k)$ La cantidad de observaciones de la clase k en la región m como proporción de la cantidad de observaciones en esa región.
- Sea $k(m)$ la clase mayoritaria en la región m.
- Definimos el error de clasificación como $L(m) = 1 - p(m, k(m))$.
- Ahora para crecer y podar el árbol se sigue el mismo algoritmo de árboles de regresión.

- La única modificación necesaria es definir la función de error en cada ramificación: $Q_m(T)$ (i.e. medidad de impureza).
- Sean $R_1, ..., R_M$ un conjunto de regiones.
- Sea $p(m, k)$ La cantidad de observaciones de la clase k en la región m como proporción de la cantidad de observaciones en esa región.
- Sea $k(m)$ la clase mayoritaria en la región m.
- Definimos el error de clasificación como $L(m) = 1 - p(m, k(m))$.
- Ahora para crecer y podar el árbol se sigue el mismo algoritmo de árboles de regresión.

- La única modificación necesaria es definir la función de error en cada ramificación: $Q_m(T)$ (i.e. medidad de impureza).
- Sean $R_1, ..., R_M$ un conjunto de regiones.
- Sea $p(m, k)$ La cantidad de observaciones de la clase k en la región m como proporción de la cantidad de observaciones en esa región.
- Sea $k(m)$ la clase mayoritaria en la región m.
- Definimos el error de clasificación como $L(m) = 1 - p(m, k(m))$.
- Ahora para crecer y podar el árbol se sigue el mismo algoritmo de árboles de regresión.

- La única modificación necesaria es definir la función de error en cada ramificación: $Q_m(T)$ (i.e. medidad de impureza).
- Sean $R_1, ..., R_M$ un conjunto de regiones.
- Sea $p(m, k)$ La cantidad de observaciones de la clase k en la región m como proporción de la cantidad de observaciones en esa región.
- Sea $k(m)$ la clase mayoritaria en la región m.
- Definimos el error de clasificación como $L(m) = 1 - p(m, k(m))$.
- Ahora para crecer y podar el árbol se sigue el mismo algoritmo de árboles de regresión.

- La única modificación necesaria es definir la función de error en cada ramificación: $Q_m(T)$ (i.e. medidad de impureza).
- Sean $R_1, ..., R_M$ un conjunto de regiones.
- Sea $p(m, k)$ La cantidad de observaciones de la clase k en la región m como proporción de la cantidad de observaciones en esa región.
- Sea $k(m)$ la clase mayoritaria en la región m.
- Definimos el error de clasificación como $L(m) = 1 - p(m, k(m))$.
- Ahora para crecer y podar el árbol se sigue el mismo algoritmo de árboles de regresión.

Árboles de clasificación

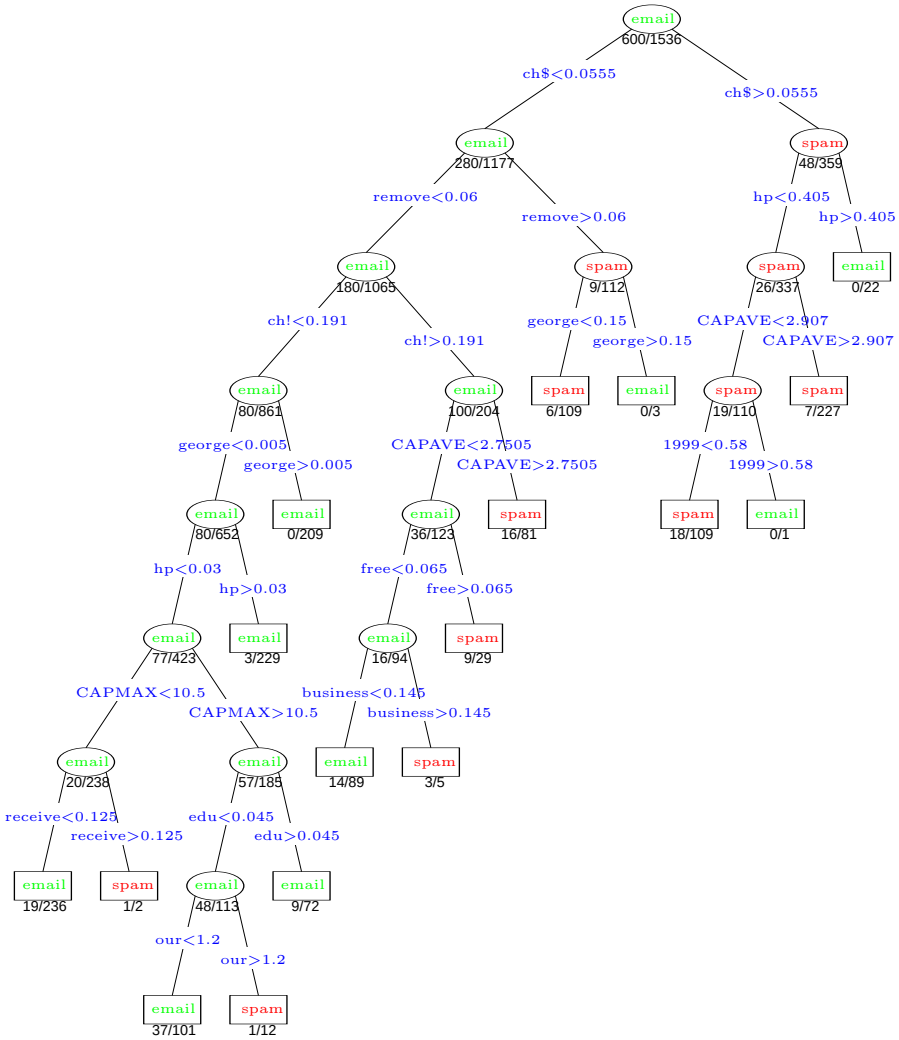


FIGURE 9.5. *The pruned tree for the* `spam` *example. The split variables are shown in blue on the branches, and the classification is shown in every node. The numbers under the terminal nodes indicate misclassification rates on the test data.*

Árboles de clasificación

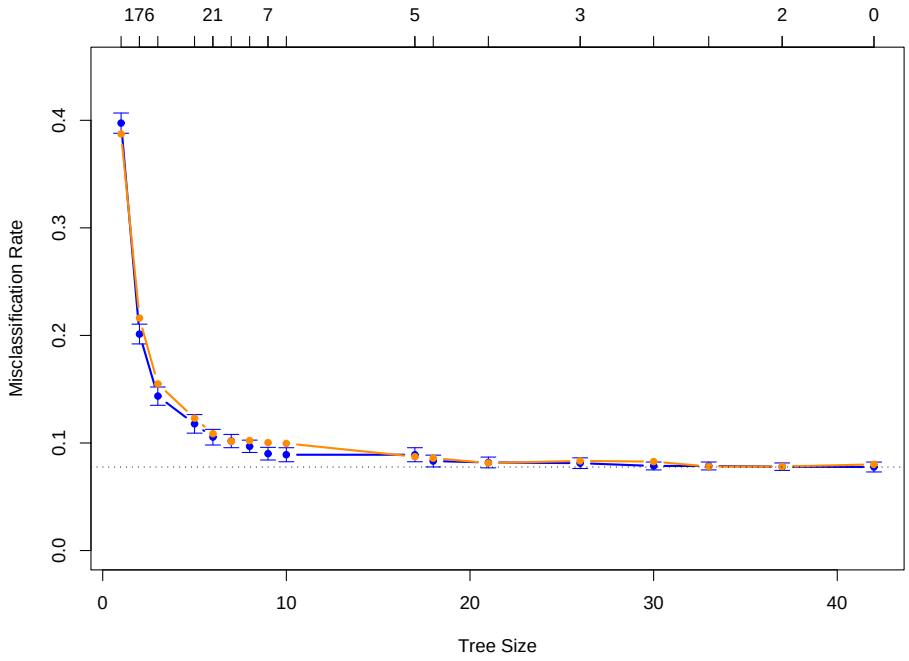


FIGURE 9.4. *Results for* `spam` *example. The blue curve is the* 10*-fold cross-validation estimate of misclassification rate as a function of tree size, with standard error bars. The minimum occurs at a tree size with about* 17 *terminal nodes (using the "one-standard--error" rule). The orange curve is the test error, which tracks the CV error quite closely. The cross-validation is indexed by values of* α*, shown above. The tree sizes shown below refer to* $|T_\alpha|$*, the size of the original tree indexed by* α*.*

Contenido

Random Forests

- Es una técnica para reducir varianza construyendo una gran cantidad de árboles no correlacionados para después promediarlos.

Random Forests: Algoritmo

- Para $b = 1, ..., B$
- Muestrear Z^* del mismo tamaño de la muestra.
- Construir un árbol T_b usando el siguiente procedimiento hasta que se alcance un número de datos sea menor que n_{min} en cada nodo.
  1. Elegir aleatoriamente m variables de las p.
  2. Elegir la mejor división en el nodo entre las m.
  3. Repetir los dos pasos anteriores hasta alcanzar el mínimo en cada hoja.
- Promedia los árboles en el caso de un problema de regresión o usar voto mayoritario (cada árbol contribuye con un voto) en el caso de clasificación.

Random Forests: Propiedades

- Si cada árbol se crece bastante es posible reducir el sesgo pero la varianza aumenta. Al promediar se reduce la varianza.
- Como los árboles se contruyen usando el mismo procedimiento, el sesgo del promedio es similar al sesgo de cada árbol. Potencialmente la ganancia está en en la reducción de varianza.
- La reducción de la varianza se obtiene eficientemente al promediar árboles no correlacionados.
- La elección aleatoria en cada división garantiza que unas pocas variables no dominen la regresión.

Random Forests: Propiedades

- Si cada árbol se crece bastante es posible reducir el sesgo pero la varianza aumenta. Al promediar se reduce la varianza.
- Como los árboles se contruyen usando el mismo procedimiento, el sesgo del promedio es similar al sesgo de cada árbol. Potencialmente la ganancia está en en la reducción de varianza.
- La reducción de la varianza se obtiene eficientemente al promediar árboles no correlacionados.
- La elección aleatoria en cada división garantiza que unas pocas variables no dominen la regresión.

Random Forests: Propiedades

- Si cada árbol se crece bastante es posible reducir el sesgo pero la varianza aumenta. Al promediar se reduce la varianza.
- Como los árboles se contruyen usando el mismo procedimiento, el sesgo del promedio es similar al sesgo de cada árbol. Potencialmente la ganancia está en en la reducción de varianza.
- La reducción de la varianza se obtiene eficientemente al promediar árboles no correlacionados.
- La elección aleatoria en cada división garantiza que unas pocas variables no dominen la regresión.

Random Forests: Propiedades

- Si cada árbol se crece bastante es posible reducir el sesgo pero la varianza aumenta. Al promediar se reduce la varianza.
- Como los árboles se contruyen usando el mismo procedimiento, el sesgo del promedio es similar al sesgo de cada árbol. Potencialmente la ganancia está en en la reducción de varianza.
- La reducción de la varianza se obtiene eficientemente al promediar árboles no correlacionados.
- La elección aleatoria en cada división garantiza que unas pocas variables no dominen la regresión.

Random Forests: Rendimiento

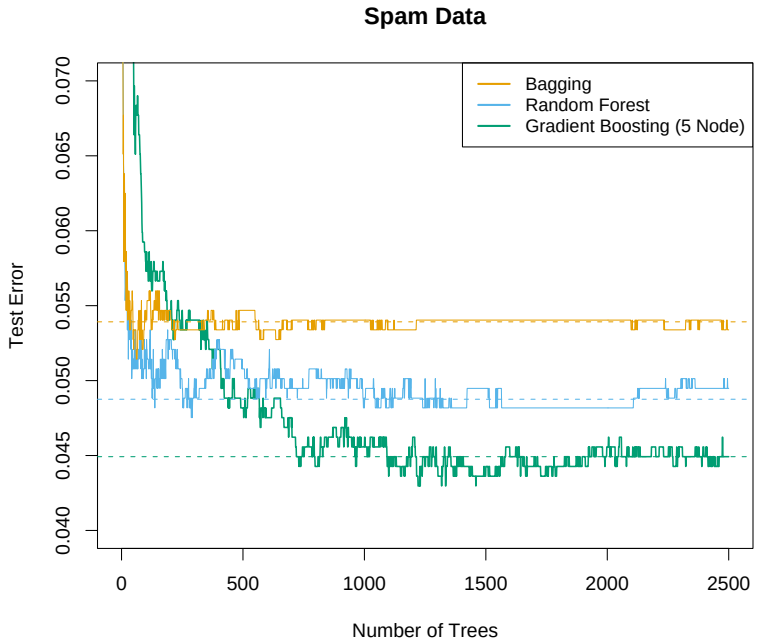


FIGURE 15.1. *Bagging, random forest, and gradient boosting, applied to the spam data. For boosting, 5-node trees were used, and the number of trees were chosen by 10-fold cross-validation (2500 trees). Each "step" in the figure corresponds to a change in a single misclassification (in a test set of 1536).*

Random Forests: Rendimiento

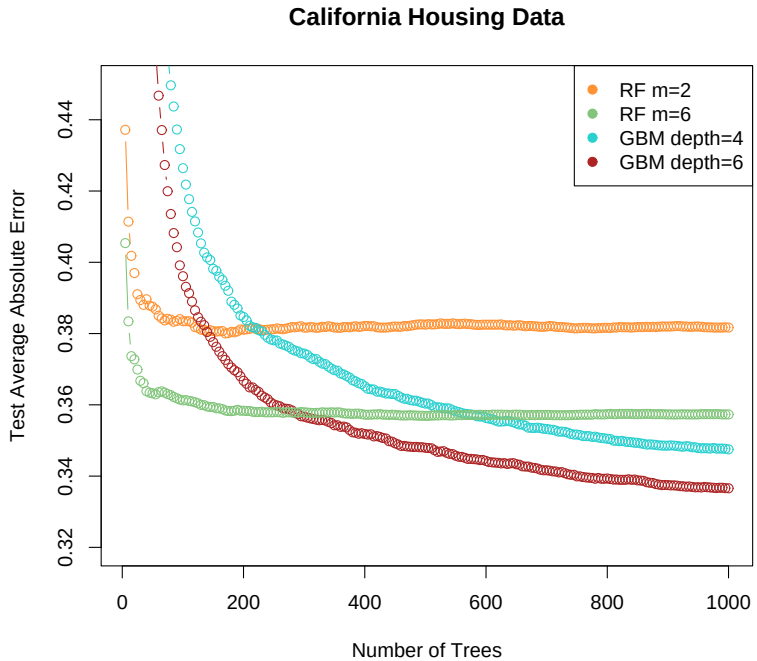


FIGURE 15.3. *Random forests compared to gradient boosting on the California housing data. The curves represent mean absolute error on the test data as a function of the number of trees in the models. Two random forests are shown, with* $m = 2$ *and* $m = 6$*. The two gradient boosted models use a shrinkage parameter* $\nu = 0.05$ *in (10.41), and have interaction depths of* 4 *and* 6*. The boosted models outperform random forests.*

Random Forests: Recomendaciones

- Para clasificación $m = \sqrt{p}$ y $n_{min} = 1$.
- Para regresión $m = \frac{p}{3}$ y $n_{min} = 5$.
- En la práctica deben ser calibrados. Por ejemplo, en el ejemplo de California, funciona mejor otros valores.

Random Forests: Importancia relativa

- Dado un nodo t del árbol, sea $\nu(t)$ la variable que se usa en ese nodo para dividir en el árbol. Sea $L(\nu(t))$ la pérdida en ese nodo si no se hiciera la división y $L_1(t)$ y $L_2(t)$ la pérdida en cada una de las divisiones que se realizan.
- Definamos la importancia de usar la variable l en el árbol T como:

$$I_l^2(T) = \sum_t i_t^2 I(v(t) = l)$$

donde $i_t^2 = (L(\nu(t)) - L(1) - L(2))^2$

- Para un bosque aleatorio se define como:

$$I_l^2 = \frac{1}{B} \sum_b I_l^2(T_b)$$

- Se normaliza la importancia más alta en 100.

Ejemplo: Importancia relativa

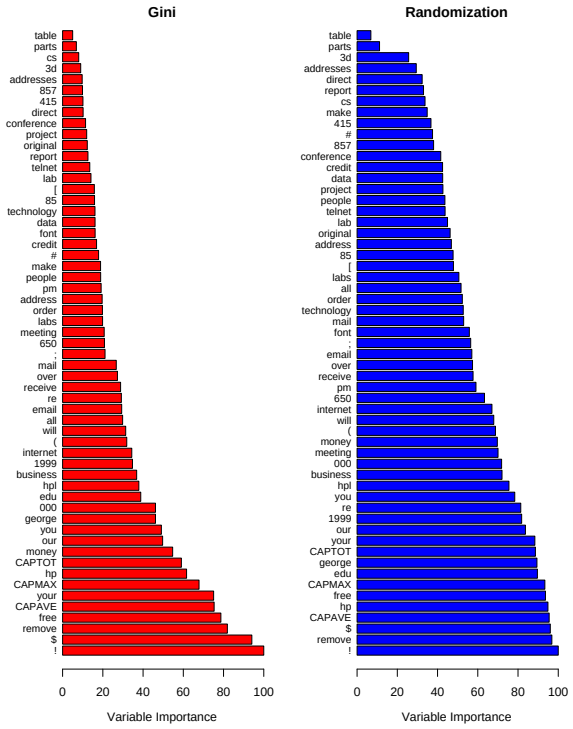


FIGURE 15.5. *Variable importance plots for a classification random forest grown on the* `spam` *data. The left plot bases the importance on the Gini splitting index, as in gradient boosting. The rankings compare well with the rankings produced by gradient boosting (Figure 10.6 on page 316). The right plot uses* OOB *randomization to compute variable importances, and tends to spread the importances more uniformly.*

Contenido

Estimación de densidades

- Supongamos que $x_1, ..., x_N$ es una muestra de datos tomada con una distribución con densidad $f_X(x)$.
- Un primer estimador local es:

$$f_X(x_0) = \frac{num\{N(x_0)\}}{N\lambda} \tag{7}$$

donde $N(x_0)$ es una vecindad de tamaño λ.

- Una versión suavizada es:

$$f_X(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \tag{8}$$

donde por ejemplo $K_\lambda(x_0, x_i) = \psi_\lambda(x - x_0)$ y ψ_λ es la densidad Gaussiana $(0, \lambda^2)$.

Estimación de densidades

- Supongamos que $x_1, ..., x_N$ es una muestra de datos tomada con una distribución con densidad $f_X(x)$.
- Un primer estimador local es:

$$f_X(x_0) = \frac{num\{N(x_0)\}}{N\lambda} \tag{7}$$

donde $N(x_0)$ es una vecindad de tamaño λ.

- Una versión suavizada es:

$$f_X(x_0) = \frac{1}{N\lambda}\sum_{i=1}^{N} K_\lambda(x_0, x_i) \tag{8}$$

donde por ejemplo $K_\lambda(x_0, x_i) = \psi_\lambda(x - x_0)$ y ψ_λ es la densidad Gaussiana $(0, \lambda^2)$.

Estimación de densidades

- Supongamos que $x_1, ..., x_N$ es una muestra de datos tomada con una distribución con densidad $f_X(x)$.
- Un primer estimador local es:

$$f_X(x_0) = \frac{num\{N(x_0)\}}{N\lambda} \tag{7}$$

donde $N(x_0)$ es una vecindad de tamaño λ.

- Una versión suavizada es:

$$f_X(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \tag{8}$$

donde por ejemplo $K_\lambda(x_0, x_i) = \psi_\lambda(x - x_0)$ y ψ_λ es la densidad Gaussiana $(0, \lambda^2)$.

Estimación de densidades

- En forma reducida:

$$f_X(x) = \frac{1}{N\lambda} \sum_{i=1}^{N} \psi_\lambda(x - x_i) \tag{9}$$

- En p dimensiones:

$$f_X(x) = \frac{1}{N\lambda(2\lambda^2\pi)^{\frac{p}{2}}} \sum_{i=1}^{N} \exp(-\frac{1}{2}(\frac{x - x_i}{\lambda})^2) \tag{10}$$

Estimación de densidades

- En forma reducida:

$$f_X(x) = \frac{1}{N\lambda} \sum_{i=1}^{N} \psi_\lambda(x - x_i) \tag{9}$$

- En p dimensiones:

$$f_X(x) = \frac{1}{N\lambda(2\lambda^2\pi)^{\frac{p}{2}}} \sum_{i=1}^{N} \exp(-\frac{1}{2}(\frac{x - x_i}{\lambda})^2) \tag{10}$$

Estimación de densidades

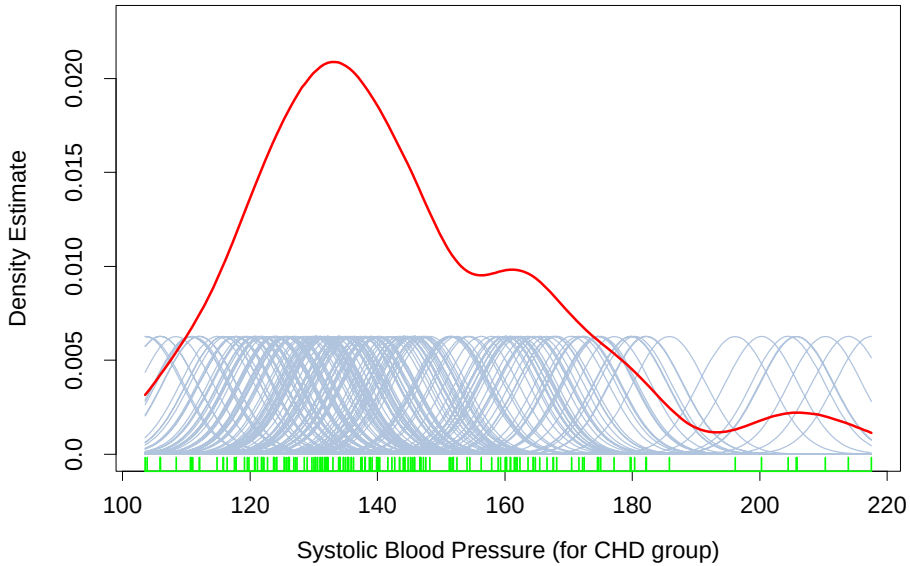


FIGURE 6.13. *A kernel density estimate for systolic blood pressure (for the CHD group). The density estimate at each point is the average contribution from each of the kernels at that point. We have scaled the kernels down by a factor of 10 to make the graph readable.*

Contenido

AdaBoost

- Viene de Adaptive Boosting.
- Supongamos que tenemos una muestra $\tau_n = \{(x_1, y_1), ..., (x_n, y_n)\}$ donde $y_i \in \{-1, 1\}$.
- Defina unos pesos iniciales para cada observación i: $D_1(i) = \frac{1}{n}$. D_t siempre será una distribución sobre las n observaciones.

AdaBoost: Algoritmo

- Para cada $t = 1, ..., T$
- Construir un clasificador (puede ser débil) h_t que minimice la función de pérdida:
  1. Defina el error e_t como:

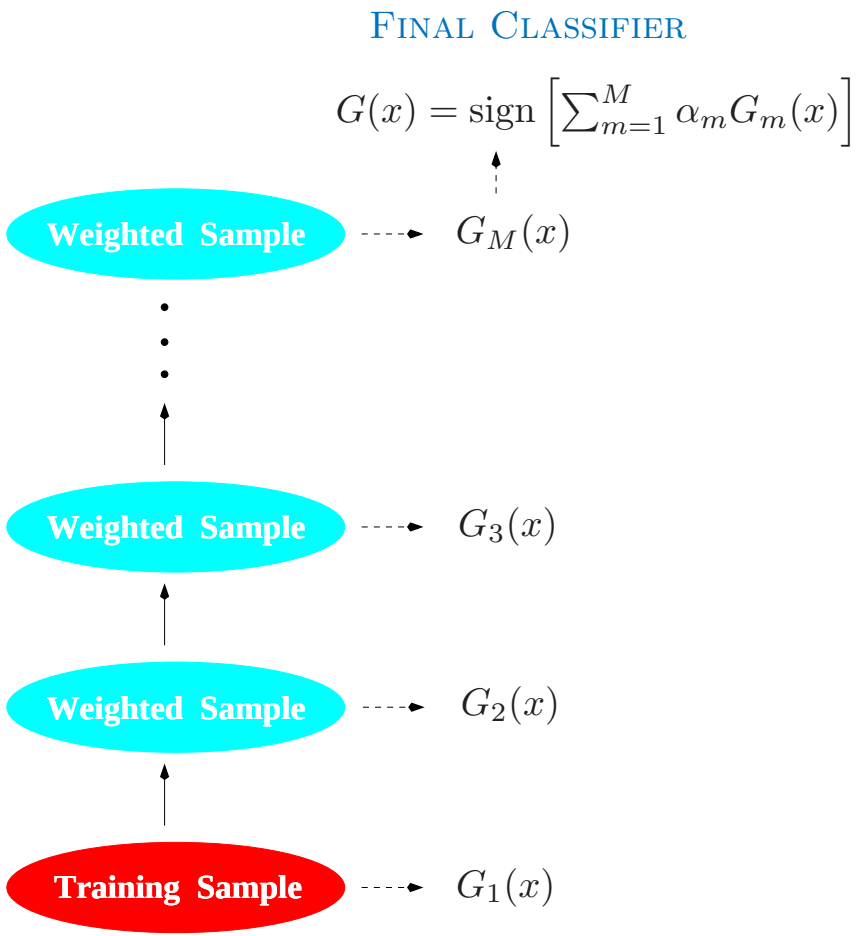
$$e_t = \sum_{i=1}^{n} D_t(i) I(y_i \neq h_t(x_i)) \tag{11}$$

  2. Calcular $\alpha_t = \frac{1}{2} \log(\frac{1-e_t}{e_t})$
  3. Modificar los pesos:

$$D_{t+1}(i) \rightarrow \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

  donde $Z_t = \sum_{i=1}^{n} D_t(i)$
- $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

AdaBoost: Algoritmo



Boosting: Es un clasificador muy potente

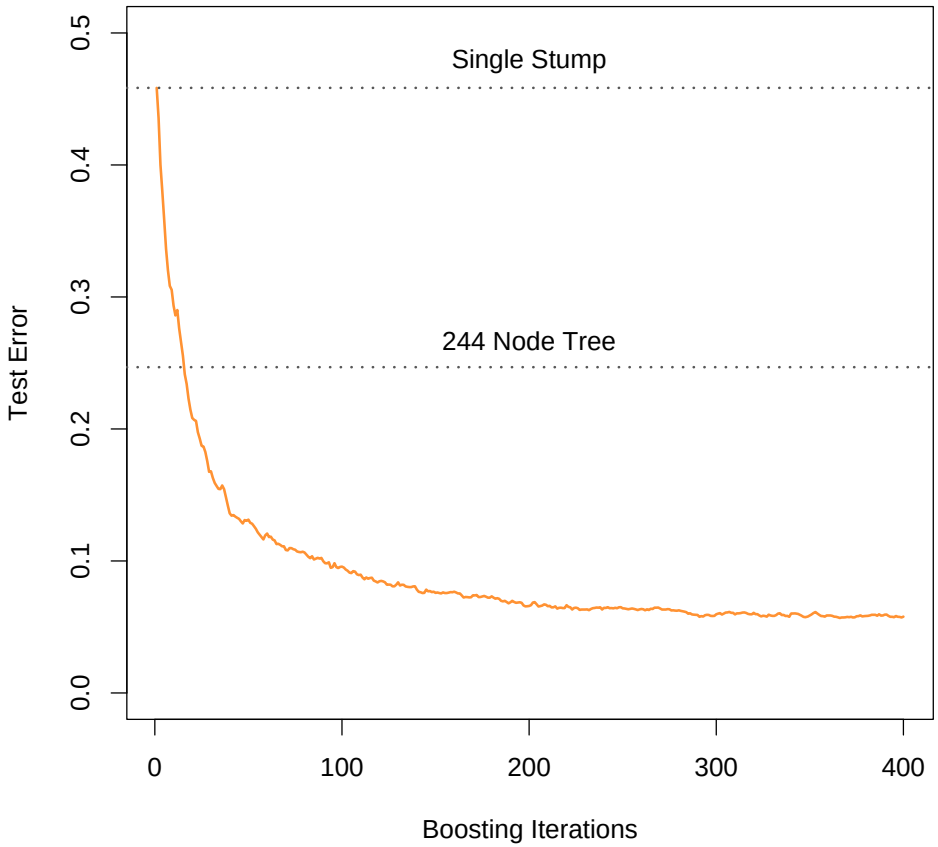


FIGURE 10.2. *Simulated data (10.2): test error rate for boosting with stumps, as a function of the number of iterations. Also shown are the test error rate for a single stump, and a 244-node classification tree.*

Boosting: Observaciones

- Si elegimos un ejemplo de τ de forma aleatoria usando la distribución D_t la probabilidad de que h_t lo clasifique de forma errada es e_t.
- e_t se mide usando la misma distribución D_t con respecto a la cual se entrena la hipótesis h_t.
- No es absolutamente necesario que h_t minimice e_t solamente que sea marginalmente mejor que un clasificador aleatorio (i.e., la hipótesis de clasificadores débiles significa que existe un $\gamma > 0$ tal que $e_t < \frac{1}{2} - \gamma$).
- $\alpha_t > 0$ si $e_t < \frac{1}{2}$ y es mayor entre menor sea el error.
- Si calculamos el error de h_t usando la distribución D_{t+1} este es idéntico a $\frac{1}{2}$.
- Más adelante mostramos que el error de entrenamiento cae exponencialmente como función del número de clasificadores débiles que se usen para combinar.

Boosting: Observaciones

- Por el momento no hay garantía de que el clasificador tenga un buen error de generalización.
- El error de entrenamiento $h(x)$ lo denotamos por $\widehat{err}(h)$ y es simplemente la frecuencia de ejemplos mal clasificados por h sin ningun tipo de ponderación.

Boosting en acción: clasificadores débiles

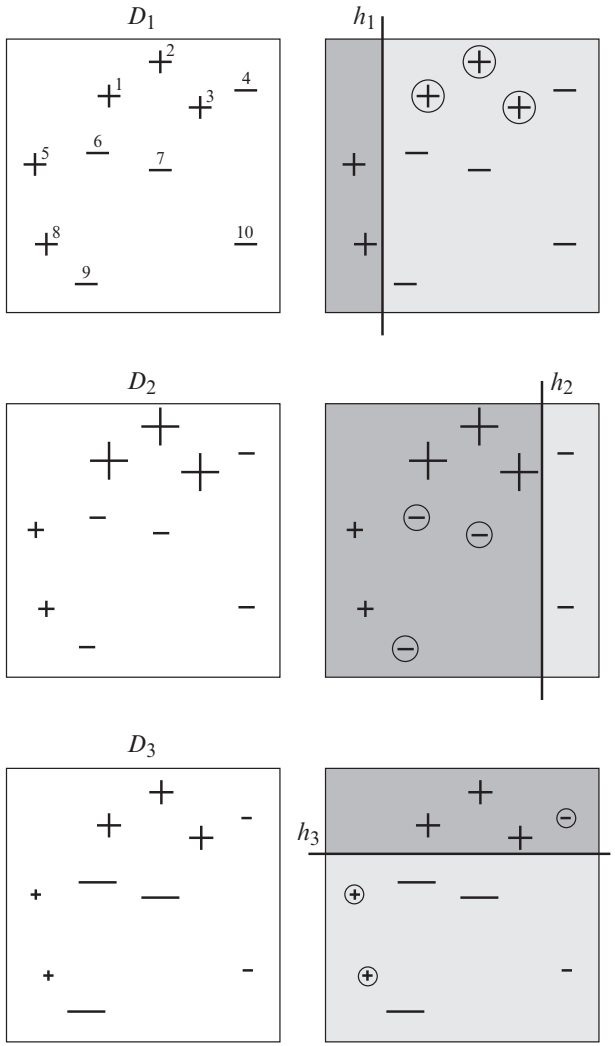


Figure 1.1
An illustration of how AdaBoost behaves on a tiny toy problem with $m = 10$ examples. Each row depicts one round, for $t = 1, 2, 3$. The left box in each row represents the distribution D_t, with the size of each example scaled in proportion to its weight under that distribution. Each box on the right shows the weak hypothesis h_t, where darker shading indicates the region of the domain predicted to be positive. Examples that are misclassified by h_t have been circled.

Boosting en acción: clasificadores débiles

Table 1.1
The numerical calculations corresponding to the toy example in figure 1.1

	1	2	3	4	5	6	7	8	9	10	
$D_1(i)$	<u>0.10</u>	<u>0.10</u>	<u>0.10</u>	0.10	0.10	0.10	0.10	0.10	0.10	0.10	$\epsilon_1 = 0.30, \alpha_1 \approx 0.42$
$e^{-\alpha_1 y_i h_1(x_i)}$	1.53	1.53	1.53	0.65	0.65	0.65	0.65	0.65	0.65	0.65	
$D_1(i)\ e^{-\alpha_1 y_i h_1(x_i)}$	0.15	0.15	0.15	0.07	0.07	0.07	0.07	0.07	0.07	0.07	$Z_1 \approx 0.92$
$D_2(i)$	0.17	0.17	0.17	0.07	0.07	<u>0.07</u>	<u>0.07</u>	0.07	<u>0.07</u>	0.07	$\epsilon_2 \approx 0.21, \alpha_2 \approx 0.65$
$e^{-\alpha_2 y_i h_2(x_i)}$	0.52	0.52	0.52	0.52	0.52	1.91	1.91	0.52	1.91	0.52	
$D_2(i)\ e^{-\alpha_2 y_i h_2(x_i)}$	0.09	0.09	0.09	0.04	0.04	0.14	0.14	0.04	0.14	0.04	$Z_2 \approx 0.82$
$D_3(i)$	0.11	0.11	0.11	<u>0.05</u>	<u>0.05</u>	0.17	0.17	<u>0.05</u>	0.17	0.05	$\epsilon_3 \approx 0.14, \alpha_3 \approx 0.92$
$e^{-\alpha_3 y_i h_3(x_i)}$	0.40	0.40	0.40	2.52	2.52	0.40	0.40	2.52	0.40	0.40	
$D_3(i)\ e^{-\alpha_3 y_i h_3(x_i)}$	0.04	0.04	0.04	0.11	0.11	0.07	0.07	0.11	0.07	0.02	$Z_3 \approx 0.69$

Calculations are shown for the ten examples as numbered in the figure. Examples on which hypothesis h_t makes a mistake are indicated by underlined figures in the rows marked D_t.

Boosting en acción: clasificadores débiles

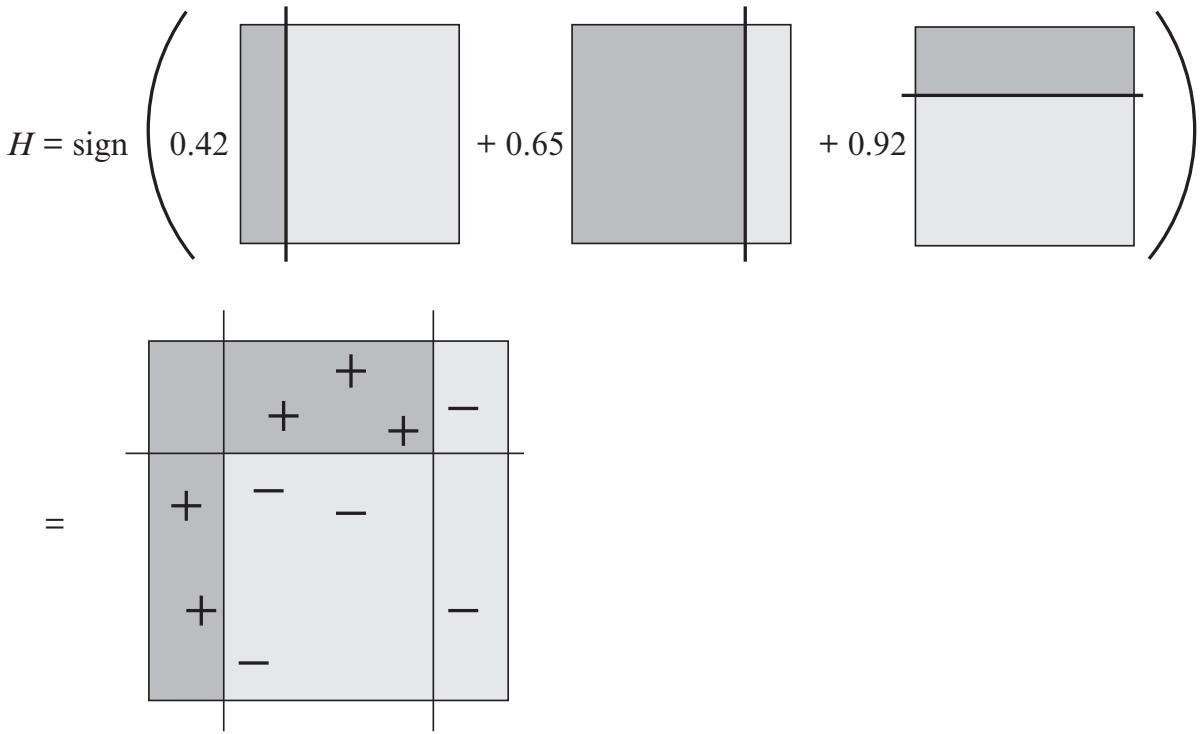


Figure 1.2
The combined classifier for the toy example of figure 1.1 is computed as the sign of the weighted sum of the three weak hypotheses, $\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3$, as shown at the top. This is equivalent to the classifier shown at the bottom. (As in figure 1.1, the regions that a classifier predicts positive are indicated using darker shading.)

Ejemplo de clasificación de enfermedad cardiaca

Table 1.2
The first six base classifiers found when using AdaBoost on the heart-disease dataset

Round	If	Then Predict	Else Predict
1	thalamus normal	healthy	sick
2	number of major vessels colored by fluoroscopy > 0	sick	healthy
3	chest pain type is asymptomatic	sick	healthy
4	ST depression induced by exercise relative to rest ≥ 0.75	sick	healthy
5	cholesterol ≥ 228.5	sick	healthy
6	resting electrocardiographic results are normal	healthy	sick

Ejemplo de clasificación de enfermedad cardiaca

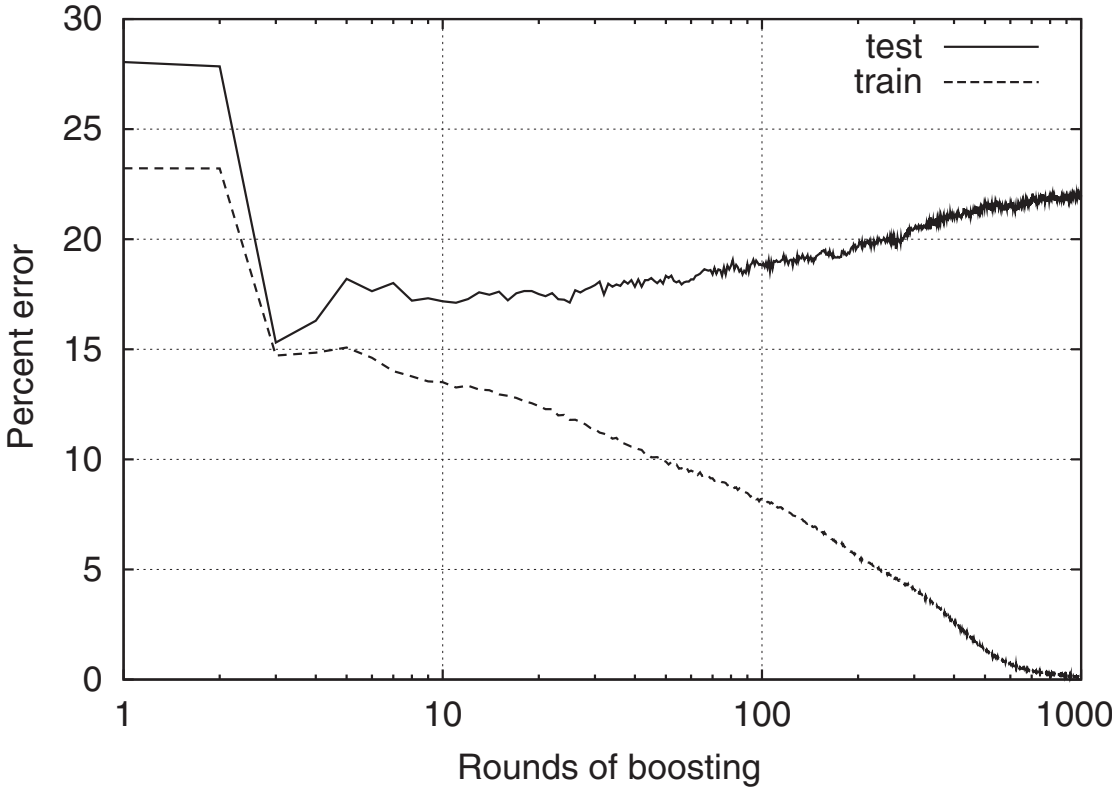


Figure 1.5
The training and test percent error rates obtained using boosting on the heart-disease dataset. Results are averaged over multiple train-test splits of the data.

Non Overfitting

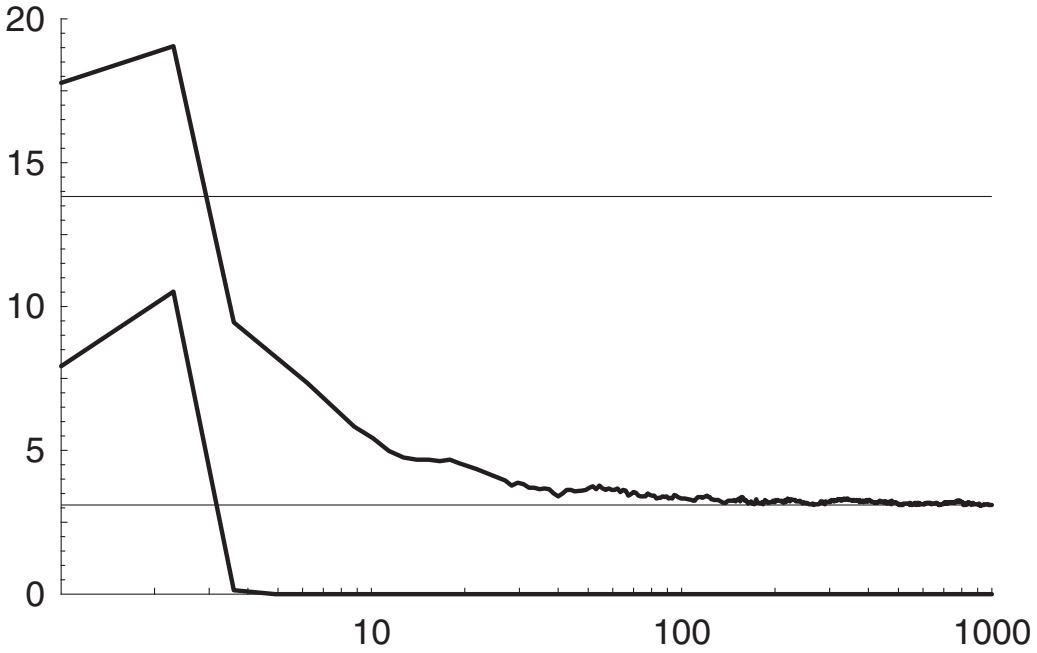


Figure 1.7
The training and test percent error rates obtained using boosting on an OCR dataset with C4.5 as the base learner. The top and bottom curves are test and training error, respectively. The top horizontal line shows the test error rate using just C4.5. The bottom line shows the final test error rate of AdaBoost after 1000 rounds. (Reprinted with permission of the Institute of Mathematical Statistics.)

Non Overfitting

- En el anterior ejemplo el error de prueba no empeora a pesar de que la complejidad aumenta.
- Esto es una ilustración de la teoría marginal del boosting. No solo importa el error sino la confianza en la clasficación.
- La confianza se mide como $y_t f(x_t) \in (-1, 1)$ donde $f = \sum_{t=1}^{T} \alpha_t h_t(x)$ y α se ha normalizado para que sumen 1.
- En este ejemplo la confianza aumenta con la complejidad.