CS 229: Machine Learning

Christian Shelton

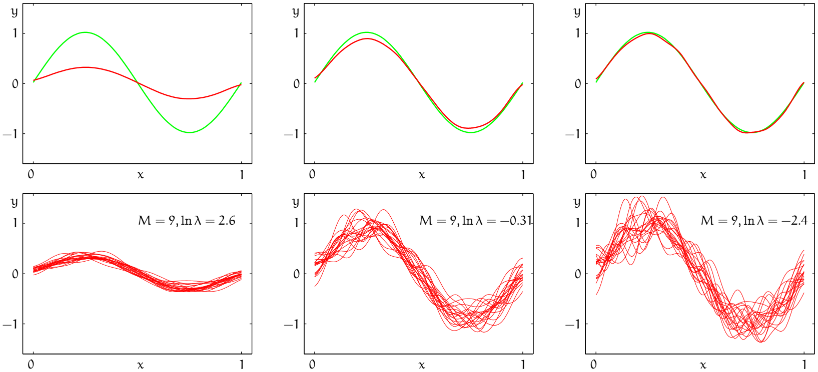
UC Riverside

Lecture 14

Slides from Lecture 14

- From UC Riverside
  - CS 229: Machine Learning
  - Professor Christian Shelton
- DO NOT REDISTRIBUTE
  - These slides contain copyrighted material (used with permission) from
    - Elements of Statistical Learning (Hastie, et al.)
    - Pattern Recognition and Machine Learning (Bishop)
  - For use only by enrolled students in the course

Bias-Variance Trade-off (again!)



Combining to Reduce Variance

If we had B different data sets, $\{D_b\}_b$:

$$D_1 \longrightarrow f_1(\cdot) \qquad D_2 \longrightarrow f_2(\cdot) \qquad \cdots \qquad D_B \longrightarrow f_m(\cdot)$$

And

$$f(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$$

Combining to Reduce Variance

If we had B different data sets, $\{D_b\}_b$:

$$D_1 \to f_1(\cdot) \qquad D_2 \to f_2(\cdot) \qquad \cdots \qquad D_B \to f_m(\cdot)$$

And

$$f(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$$

If data sets are independent, then fs are independent.
Variance of f is $O(\frac{1}{B})$.

Bootstrap

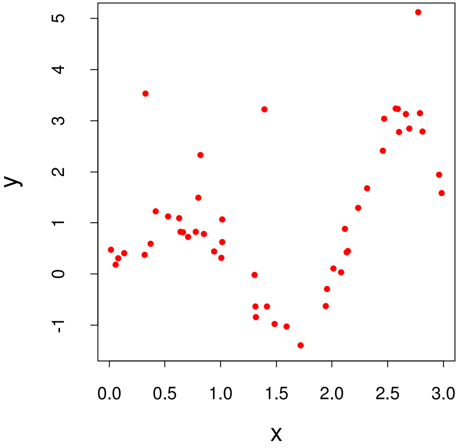
Bootstrap: Draw multiple data sets with replacement from original data set.
(all the same size as the original data set)

Bootstrap

Bootstrap: Draw multiple data sets with replacement from original data set.
(all the same size as the original data set)
Common use: Estimate variance of statistic of original data set.

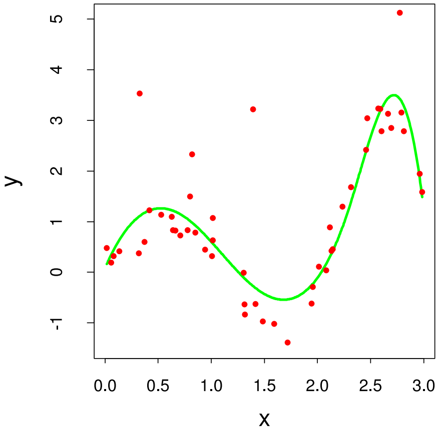
Bootstrap

Bootstrap: Draw multiple data sets with replacement from original data set.
(all the same size as the original data set)
Common use: Estimate variance of statistic of original data set.



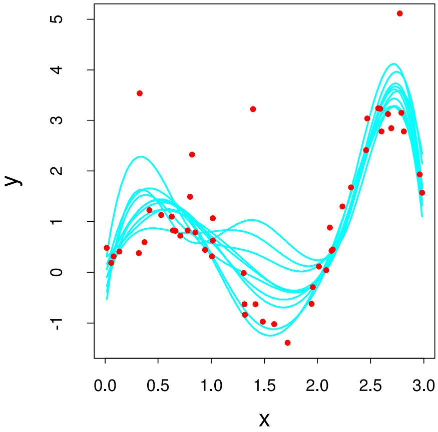
Bootstrap

Bootstrap: Draw multiple data sets with replacement from original data set.
(all the same size as the original data set)
Common use: Estimate variance of statistic of original data set.



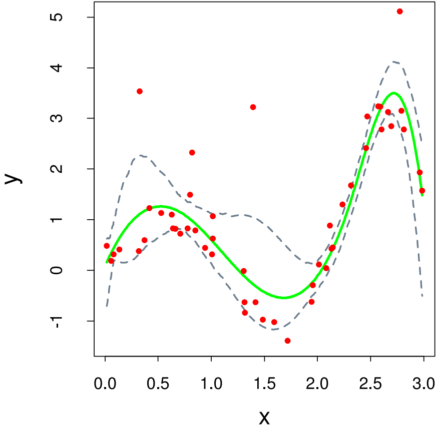
Bootstrap

Bootstrap: Draw multiple data sets with replacement from original data set.
(all the same size as the original data set)
Common use: Estimate variance of statistic of original data set.



Bootstrap

Bootstrap: Draw multiple data sets with replacement from original data set.
(all the same size as the original data set)
Common use: Estimate variance of statistic of original data set.



Bootstrap Aggregation / Bagging

Bagging:

1. for B times
   1. Draw D_b with replacement from D
   2. $f_b \leftarrow \text{learn}(D_b)$
2. Let $f(x) = \frac{1}{B} \sum_b f_b(x)$

Bootstrap Aggregation / Bagging

Bagging:

1. for B times
   1. Draw D_b with replacement from D
   2. $f_b \leftarrow \text{learn}(D_b)$
2. Let $f(x) = \frac{1}{B}\sum_b f_b(x)$

If f_b is linear, this will not change the hypothesis space.
If f_b is non-linear, this will change the hypothesis space.

Bootstrap Aggregation / Bagging

Bagging:

1. for B times
   1. Draw D_b with replacement from D
   2. $f_b \leftarrow \text{learn}(D_b)$
2. Let $f(x) = \frac{1}{B}\sum_b f_b(x)$

If f_b is linear, this will not change the hypothesis space.
If f_b is non-linear, this will change the hypothesis space.

This does not affect the bias.
This does reduce the variance (in general).

Bagged Classifier

For classification, train as before, but let

$$f(x) = \arg\max_k \sum_b \mathbb{1}(f_b(x) = k)$$

or, if $f_{b,k}(x)$ is $p(y = k \mid x)$ from f_b

$$f(x) = \arg\max_k \sum_b f_{b,k}(x)$$

Bagged Classifier

For classification, train as before, but let

$$f(x) = \arg\max_k \sum_b \mathbb{1}(f_b(x) = k)$$

or, if $f_{b,k}(x)$ is $p(y = k \mid x)$ from f_b

$$f(x) = \arg\max_k \sum_b f_{b,k}(x)$$

Note, f_b and $f_{b,k}$ are (almost) always non-linear.

Bagging Trees

Example:

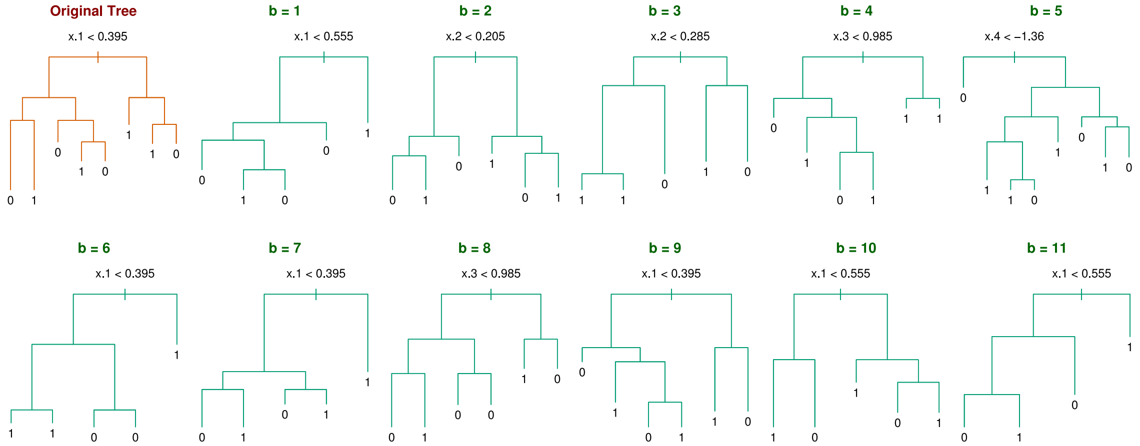
$$d = 5$$

$$n = 30$$

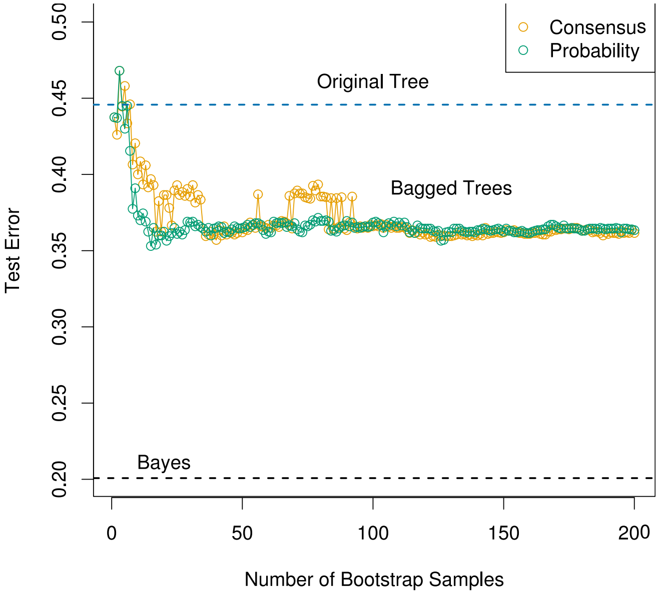
$$p(y = 1|x) = \begin{cases} 0.2 & \text{if } x_1 \leqslant 0.5 \\ 0.8 & \text{otherwise} \end{cases}$$

Classification trees with no pruning as base classifier

Bagging Trees



Bagging Trees



Boosting

Bagging builds a function

$$f(x) = \sum_{j=1}^{m} \frac{1}{m} f_j(x)$$

where f_j is draw from a base hypothesis space (a base classifier).

Boosting

Bagging builds a function

$$f(x) = \sum_{j=1}^{m} \frac{1}{m} f_j(x)$$

where f_j is draw from a base hypothesis space (a base classifier).

Let's be more general:

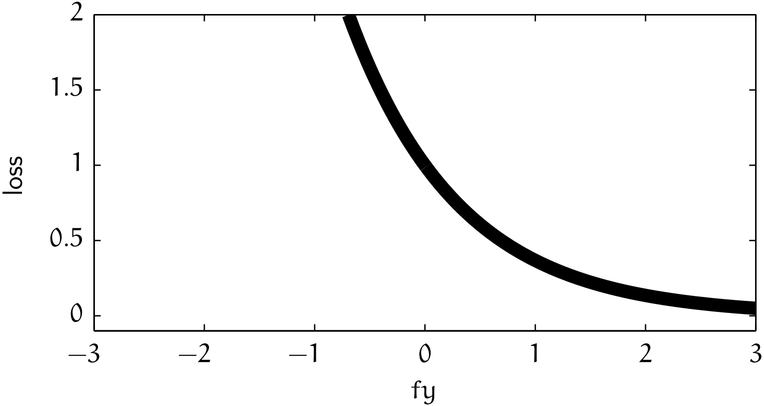
$$f(x) = \sum_{j=1}^{m} w_j f_j(x)$$

where f_j is draw from a base hypothesis space (a base classifier).

Exponential Loss

We will train with exponential loss:

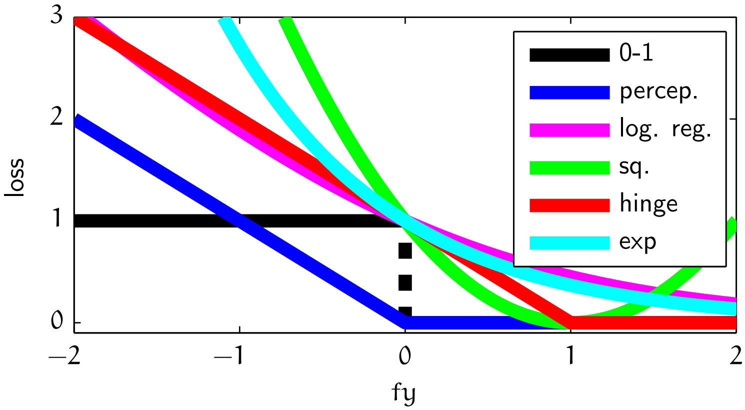
$$l(y, \hat{y}) = e^{-y\hat{y}}$$



Exponential Loss

We will train with exponential loss:

$$l(y, \hat{y}) = e^{-y\hat{y}}$$



Boosting

For binary classification:

Goal to minimize

$$
\begin{aligned}
L &= \sum_i l_{\text{exp}}(y_i, f(x_i)) \\
&= \sum_i e^{-y_i \sum_{j=1}^{m} w_j f_j(x_i)}
\end{aligned}
$$

Boosting

For binary classification:
Goal to minimize

$$\begin{aligned} L &= \sum_i l_{\text{exp}}(y_i, f(x_i)) \\ &= \sum_i e^{-y_i \sum_{j=1}^{m} w_j f_j(x_i)} \end{aligned}$$

Do greedy optimization: Add each $\{w_j, f_j\}$ one at a time.

Boosting

On round m, let $\tilde{f}(x) = \sum_{j=1}^{m-1} w_j f_j(x)$
Then we need to optimize over w_m and f_m:

$$\begin{aligned} L &= \sum_i e^{-y_i(\tilde{f}(x_i)+w_m f_m(x_i))} \\ &= \sum_i \alpha_i e^{-y_i w_m f_m(x_i)} && (\alpha_i = e^{-y_i \tilde{f}(x_i)}) \end{aligned}$$

Boosting

On round m, let $\tilde{f}(x) = \sum_{j=1}^{m-1} w_j f_j(x)$
Then we need to optimize over w_m and f_m:

$$L = \sum_i e^{-y_i(\tilde{f}(x_i)+w_m f_m(x_i))}$$

$$= \sum_i \alpha_i e^{-y_i w_m f_m(x_i)} \qquad (\alpha_i = e^{-y_i \tilde{f}(x_i)})$$

Let C_m be the points correctly classified by f_m

Let M_m be the points misclassified by f_m

Boosting

On round m, let $\tilde{f}(x) = \sum_{j=1}^{m-1} w_j f_j(x)$
Then we need to optimize over w_m and f_m:

$$\begin{aligned} L &= \sum_i e^{-y_i(\tilde{f}(x_i)+w_m f_m(x_i))} \\ &= \sum_i \alpha_i e^{-y_i w_m f_m(x_i)} && (\alpha_i = e^{-y_i \tilde{f}(x_i)}) \end{aligned}$$

Let C_m be the points correctly classified by f_m

Let M_m be the points misclassified by f_m

$$= \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m}$$

Boosting

On round m, let $\tilde{f}(x) = \sum_{j=1}^{m-1} w_j f_j(x)$

Then we need to optimize over w_m and f_m:

$$\begin{aligned} L &= \sum_i e^{-y_i(\tilde{f}(x_i) + w_m f_m(x_i))} \\ &= \sum_i \alpha_i e^{-y_i w_m f_m(x_i)} \qquad\qquad (\alpha_i = e^{-y_i \tilde{f}(x_i)}) \end{aligned}$$

Let C_m be the points correctly classified by f_m

Let M_m be the points misclassified by f_m

$$\begin{aligned} &= \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m} \\ &= e^{-w_m} \sum_i \alpha_i + \left(e^{w_m} - e^{-w_m}\right) \sum_{i \in M_m} \alpha_i \end{aligned}$$

Boosting

On round m, let $\tilde{f}(x) = \sum_{j=1}^{m-1} w_j f_j(x)$
Then we need to optimize over w_m and f_m:

$$\begin{aligned} L &= \sum_i e^{-y_i(\tilde{f}(x_i)+w_m f_m(x_i))} \\ &= \sum_i \alpha_i e^{-y_i w_m f_m(x_i)} && (\alpha_i = e^{-y_i \tilde{f}(x_i)}) \end{aligned}$$

Let C_m be the points correctly classified by f_m

Let M_m be the points misclassified by f_m

$$\begin{aligned} &= \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m} \\ &= e^{-w_m} \sum_i \alpha_i + \left(e^{w_m} - e^{-w_m}\right) \sum_{i \in M_m} \alpha_i \end{aligned}$$

If $w_m > 0$, $e^{w_m} - e^{-w_m} > 0$ and we need to pick f_m to minimize

Boosting

On round m, let $\tilde{f}(x) = \sum_{j=1}^{m-1} w_j f_j(x)$
Then we need to optimize over w_m and f_m:

$$\begin{aligned} L &= \sum_i e^{-y_i(\tilde{f}(x_i) + w_m f_m(x_i))} \\ &= \sum_i \alpha_i e^{-y_i w_m f_m(x_i)} && (\alpha_i = e^{-y_i \tilde{f}(x_i)}) \end{aligned}$$

Let C_m be the points correctly classified by f_m

Let M_m be the points misclassified by f_m

$$\begin{aligned} &= \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m} \\ &= e^{-w_m} \sum_i \alpha_i + \left(e^{w_m} - e^{-w_m}\right) \sum_{i \in M_m} \alpha_i \end{aligned}$$

If $w_m > 0$, $e^{w_m} - e^{-w_m} > 0$ and we need to pick f_m to minimize

$$\sum_{i \in M_m} \alpha_i \qquad \text{weighted 0-1 loss}$$

Boosting

Given f_m (and therefore C_m and M_m), we select w_m as minimium of

$$L = \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m}$$

Boosting

Given f_m (and therefore C_m and M_m), we select w_m as minimium of

$$\begin{aligned} L &= \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m} \\ &= e^{-w_m}\alpha_C + e^{w_m}\alpha_M \qquad \left(\alpha_C = \sum_{i \in C_m} \alpha_i \text{ and } \alpha_M = \sum_{i \in M_m} \alpha_i\right) \end{aligned}$$

Boosting

Given f_m (and therefore C_m and M_m), we select w_m as minimium of

$$
\begin{aligned}
L &= \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m} \\
&= e^{-w_m} \alpha_C + e^{w_m} \alpha_M \qquad \left(\alpha_C = \sum_{i \in C_m} \alpha_i \text{ and } \alpha_M = \sum_{i \in M_m} \alpha_i\right) \\
0 &= \frac{dL}{dw_m} \\
&= -e^{-w_m} \alpha_C + e^{w_m} \alpha_M
\end{aligned}
$$

Boosting

Given f_m (and therefore C_m and M_m), we select w_m as minimium of

$$\begin{aligned}
L &= \sum_{i \in C_m} \alpha_i e^{-w_m} + \sum_{i \in M_m} \alpha_i e^{w_m} \\
&= e^{-w_m} \alpha_C + e^{w_m} \alpha_M \qquad\qquad (\alpha_C = \sum_{i \in C_m} \alpha_i \text{ and } \alpha_M = \sum_{i \in M_m} \alpha_i) \\
0 &= \frac{dL}{dw_m} \\
&= -e^{-w_m} \alpha_C + e^{w_m} \alpha_M \\
\frac{\alpha_C}{\alpha_M} &= e^{2w_m} \\
w_m &= \frac{1}{2} \ln \frac{\alpha_C}{\alpha_M} = \frac{1}{2} \ln \frac{n(1 - \text{wt'd err. rate of } f_m)}{n(\text{wt'd err. rate of } f_m)}
\end{aligned}$$

Boosting Algorithm

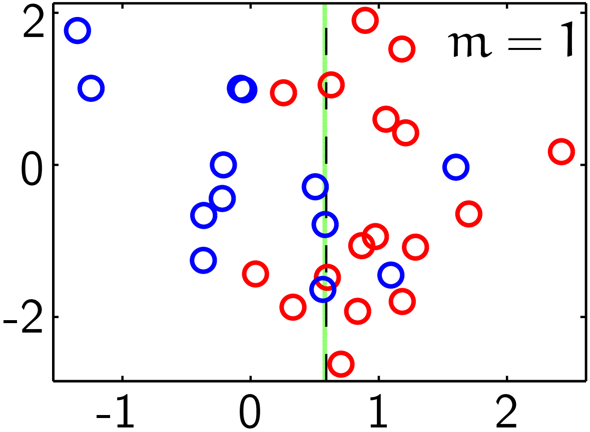
1. Let $\alpha_i = 1$ for all $i = 1, 2, \ldots, n$
2. For m in $1, 2, \ldots, M$
   1. Fit f_m to D with weights $\alpha_1, \alpha_2, \ldots, \alpha_n$
   2. Find $\text{err}_m = \frac{\sum_i \alpha_i \mathbf{1}(f_m(x_i) \neq y_i)}{\sum_i \alpha_i}$
   3. Set $w_m = \ln \frac{1 - \text{err}_m}{\text{err}_m}$
   4. Let $\alpha_i \leftarrow \alpha_i \cdot e^{w_m \mathbf{1}(y_i \neq f_m(x_i))}$
3. Return $f(x) = \sum_{m=1}^{M} w_m f_m(x)$

Boosting Algorithm

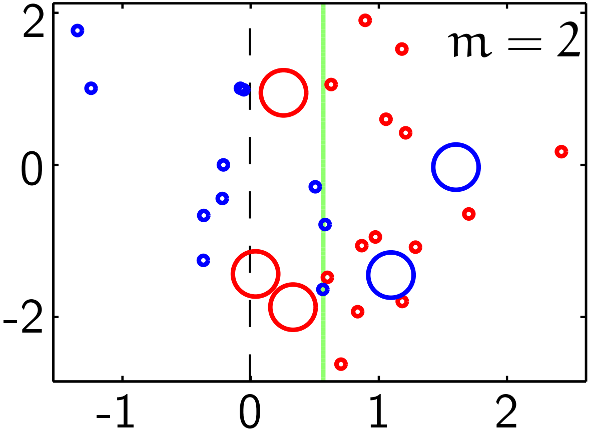
1. Let $\alpha_i = 1$ for all $i = 1, 2, \ldots, n$
2. For m in $1, 2, \ldots, M$
    1. Fit f_m to D with weights $\alpha_1, \alpha_2, \ldots, \alpha_n$
    2. Find $\text{err}_m = \frac{\sum_i \alpha_i \mathbf{1}(f_m(x_i) \neq y_i)}{\sum_i \alpha_i}$
    3. Set $w_m = \ln \frac{1 - \text{err}_m}{\text{err}_m}$
    4. Let $\alpha_i \leftarrow \alpha_i \cdot e^{w_m \mathbf{1}(y_i \neq f_m(x_i))}$
3. Return $f(x) = \sum_{m=1}^{M} w_m f_m(x)$

If base learner returns f with weighted error < 0.5 every time, will converge to zero *training* error.

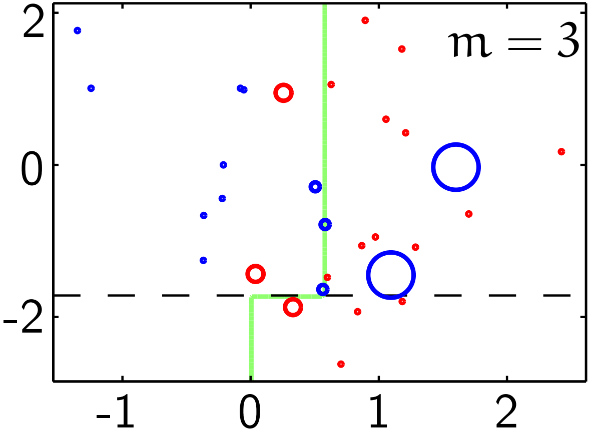
Boosting Example



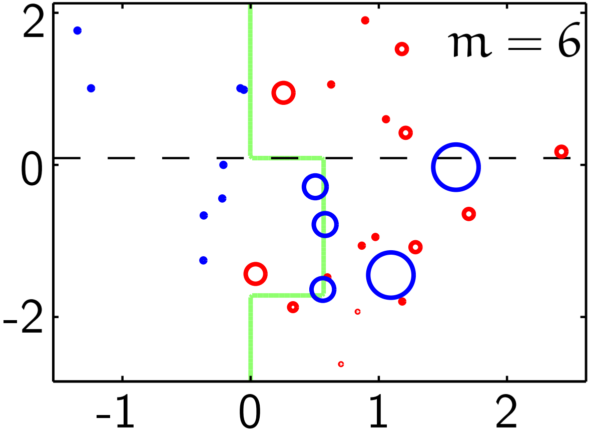
Boosting Example



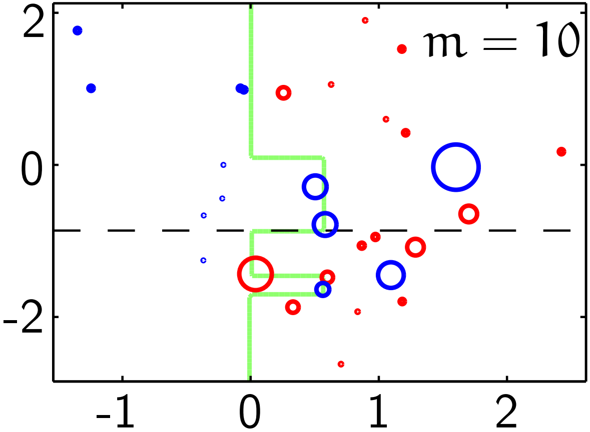
Boosting Example



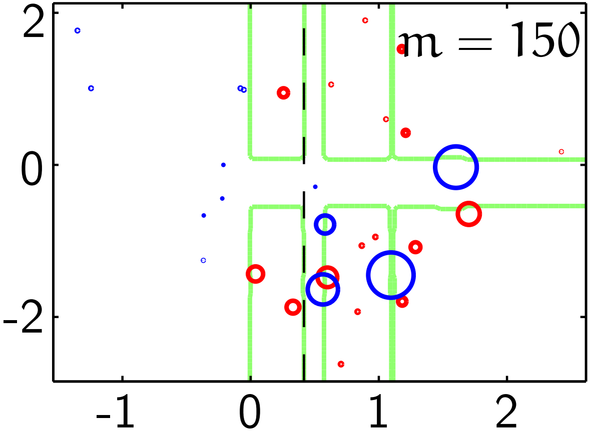
Boosting Example



Boosting Example

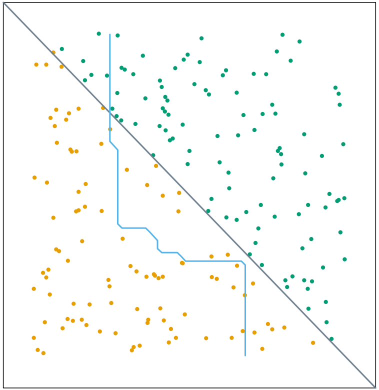


Boosting Example



Bagging and Boosting

Bagging



Boosting