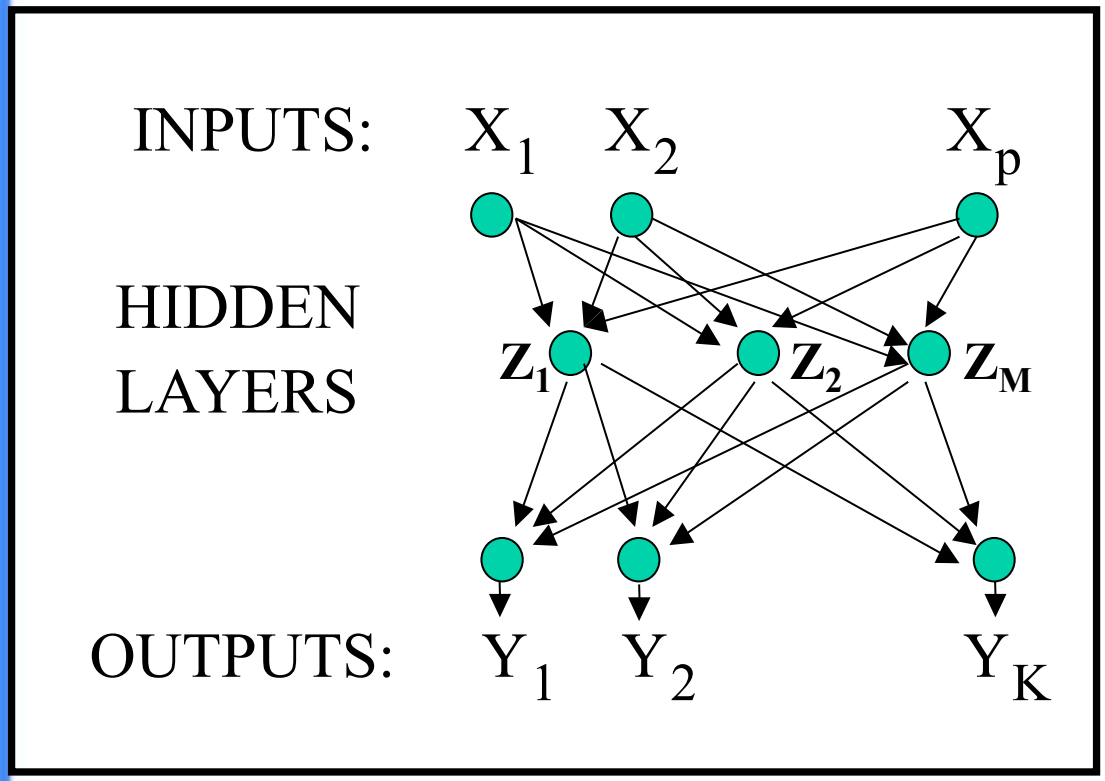
Artificial Neural Nets

Outline

- ANN
- Examples
- Support Vector Machines



$$Z_m = \sigma(a_{0m} + a_m{}^T X),$$

$$T_k = \beta_{0k} + \beta_k{}^T Z,$$

$$f_k(X) = g_k(T)$$

$$Z = (Z_1, Z_2, ..., Z_M)$$

$$T = (T_1, T_2, ..., T_K)$$

$$\sigma(v) = \frac{1}{1+e^{-v}} (SIGMOIDAL)$$

$$g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}} (SOFTMAX)$$

Estimation

$$Z_m = \sigma(a_{0m} + a_m{}^T X),$$

$$T_k = \beta_{0m} + \beta_m{}^T Z,$$

$$f_k(X) = g_k(T)$$

$$Z = (Z_1, Z_2, \ldots, Z_K)$$

$$T = (T_1, T_2, \ldots, T_K)$$

$$\sigma(v) = \frac{1}{1+e^{-v}} \quad (SIGMOIDAL)$$

$$g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}} \quad (SOFTMAX)$$

Minimize

$$R(\theta) = \sum_{k=1,i=1}^{K,N} (y_{ik} - f_k(x_i))^2$$

$$R(\theta) = -\sum_{k=1,i=1}^{K,N} y_{ik} \log f_k(x_i)$$

How to Use it

```
Example 1
library(nnet)
pex= read.table("project2/pex23.txt")
p = pex[sample(2993,200),]
predict(nnet(p[,1:10],p[,24],size=10,subset=rep(c(T,F),c(100,100))))-> y
table(round(y),p[,24])

Example 2
library(nnet)
ird <- data.frame(rbind(iris3[,,1], iris3[,,2], iris3[,,3]),
                  species = factor(c(rep("s",50), rep("c", 50), rep("v", 50))))
ir.nn2 <- nnet(species ~ ., data = ird, subset = samp, size = 6, rang = 0.1,
               decay = 1e-2, maxit = 2000)
labels.nnet <- predict(ir.nn2, ird[-samp,], type="class")
table(ird$species[-samp], labels.nnet)
#  labels.nnet
#   c  s  v
#c 22  0  3
#s  0 25  0
#v  3  0 22
# accuracy
mean(ird$species[-samp] == labels.nnet)
# 0.96
```

SVM

1. Suport vector machines can be generalized to
   Nonlinear separation.

2. It is an example of linear optimization.
   The algorithm is a simplex minimization

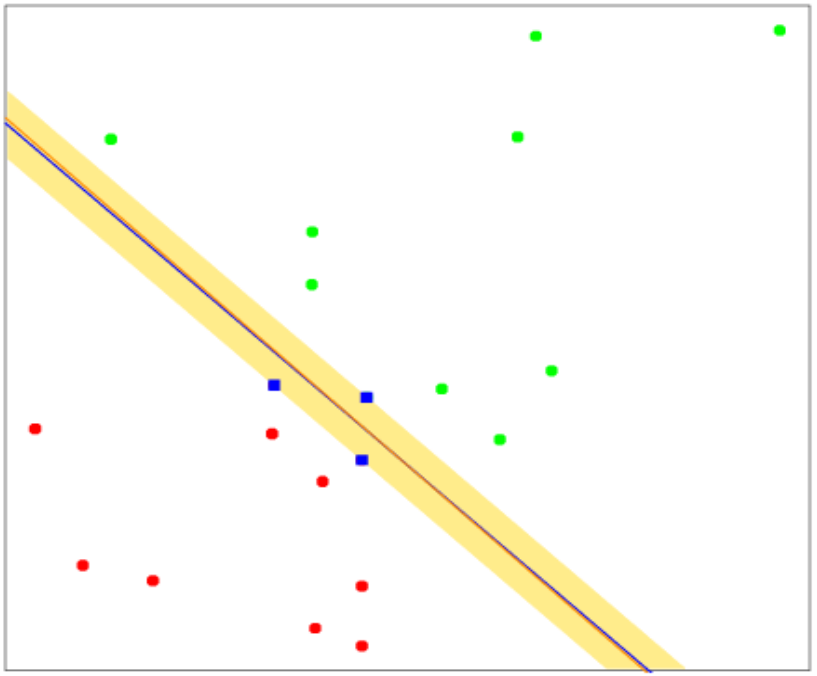
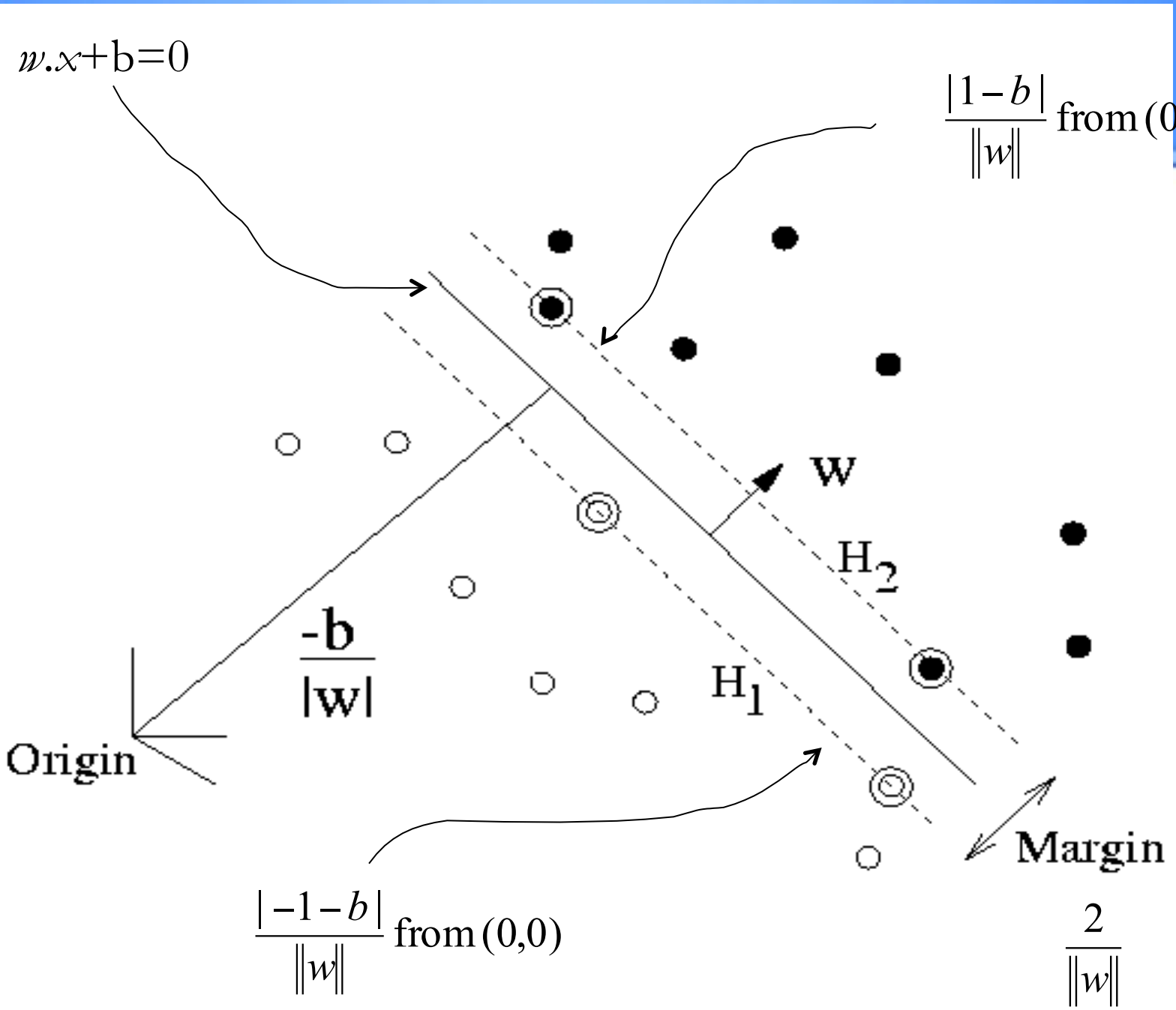


Figure 4.15: *The same data as in Figure 4.13. The shaded region delineates the maximum margin separating the two classes. There are three support points indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab. Included in the figure is the boundary found using logistic regression (red line), which is very close to the optimal separating hyperplane (see Section 12.3.3).*


w.x+b=0
$\frac{|1-b|}{\|w\|}$ from (0,0)
w
H_2
H_1
$\frac{-b}{|w|}$
Origin
Margin
$\frac{|-1-b|}{\|w\|}$ from (0,0)
$\frac{2}{\|w\|}$

How to Use it

```
********************** SVM **********************

library(e1071)
svm(p[,1:10],p[,24])
predict(svm(p[,1:10],p[,24]))
predict(svm(p[,1:10],factor(p[,24])))
predict(svm(p[1:100,1:10],factor(p[1:100,24])),p[101:200,])
predict(svm(p[1:100,1:10],factor(p[1:100,24])),p[101:200,1:10])
table(predict(svm(p[1:100,1:10],factor(p[1:100,24])),p[101:200,1:10]
    ),p[101:200,24])
```

MORE ON BAGGING BOOSTING

Unstable predictors

We can always assume

$$y = f(\boldsymbol{x}) + \varepsilon \text{ , where } \mathrm{E}(\varepsilon \mid \boldsymbol{x}) = 0$$

Assume that we have a way of constructing a predictor, $\hat{f}_D(\boldsymbol{x})$, from a dataset D.

We want to choose the estimator of f that minimizes J, squared loss for example.

$$J(\hat{f}, D) = \mathrm{E}_{y,\boldsymbol{x}}(y - \hat{f}_D(\boldsymbol{x}))^2$$

Bias-variance decomposition

If we could average over all possible datasets, let the average prediction be

$$\bar{f}(\boldsymbol{x}) = \mathrm{E}_D\, \hat{f}_D(\boldsymbol{x})$$

The average prediction error over all datasets that we might see is decomposable

$$\mathrm{E}_D\, J(\hat{f}, D) = \mathrm{E}\,\boldsymbol{\varepsilon}^2 + \mathrm{E}_x (f(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}))^2 + \mathrm{E}_{x,D} (\hat{f}_D(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}))^2$$
$$= \text{noise} + \text{bias} + \text{variance}$$

Bias-variance decomposition (cont.)

$$\mathrm{E}_D\, J(\hat{f}, D) = \mathrm{E}\,\boldsymbol{\varepsilon}^2 + \mathrm{E}_x (f(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}))^2 + \mathrm{E}_{x,D} (\hat{f}_D(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}))^2$$
$$= \text{noise} + \text{bias} + \text{variance}$$

- The noise cannot be reduced.
- The squared-bias term might be reducible
- The variance term is 0 if we use

$$\hat{f}_D(\boldsymbol{x}) = \bar{f}(\boldsymbol{x})$$

But this requires having an infinite number of datasets

Bagging (*B*ootstrap *Agg*regating)

Goal: Variance reduction

Method: Create bootstrap replicates of the dataset and fit a model to each. Average the predictions of each model.

Properties:

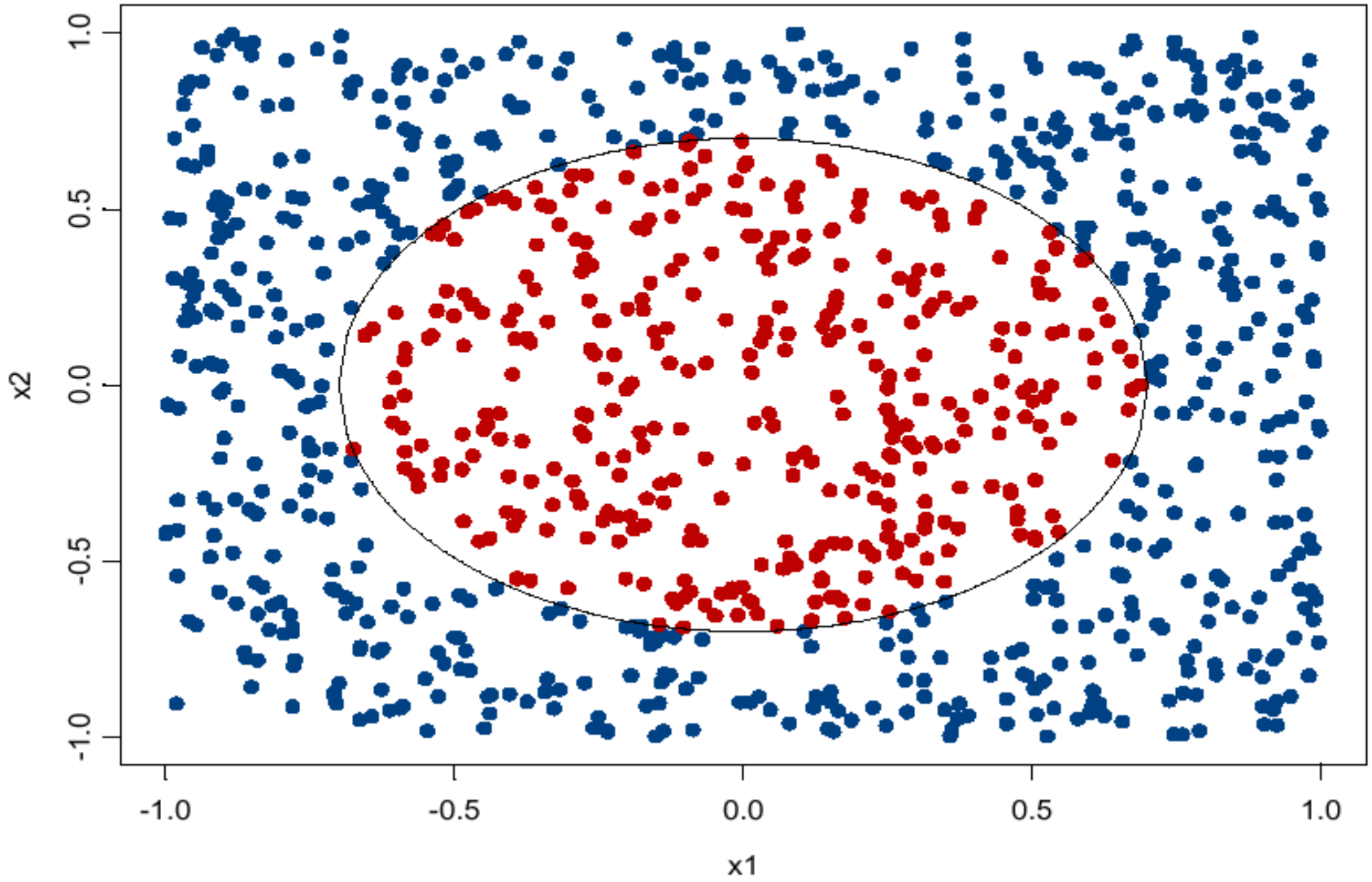
- Stabilizes "unstable" methods
- Easy to implement, parallelizable
- Theory is not fully explained

Bagging algorithm

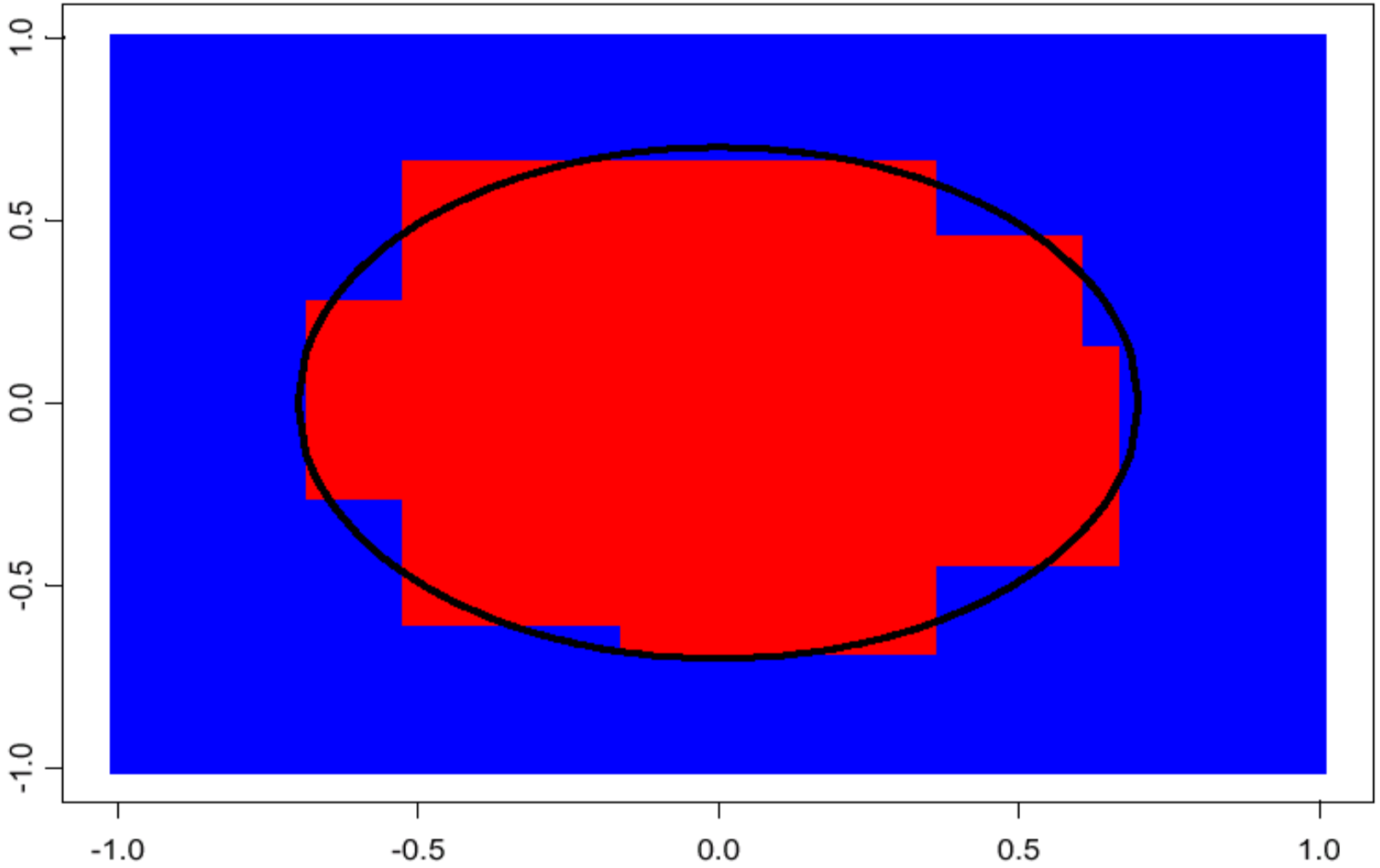
1. Create K bootstrap replicates of the dataset.
2. Fit a model to each of the replicates.
3. Average (or vote) the predictions of the K models.

Bootstrapping simulates the stream of infinite datasets in the bias-variance decomposition.

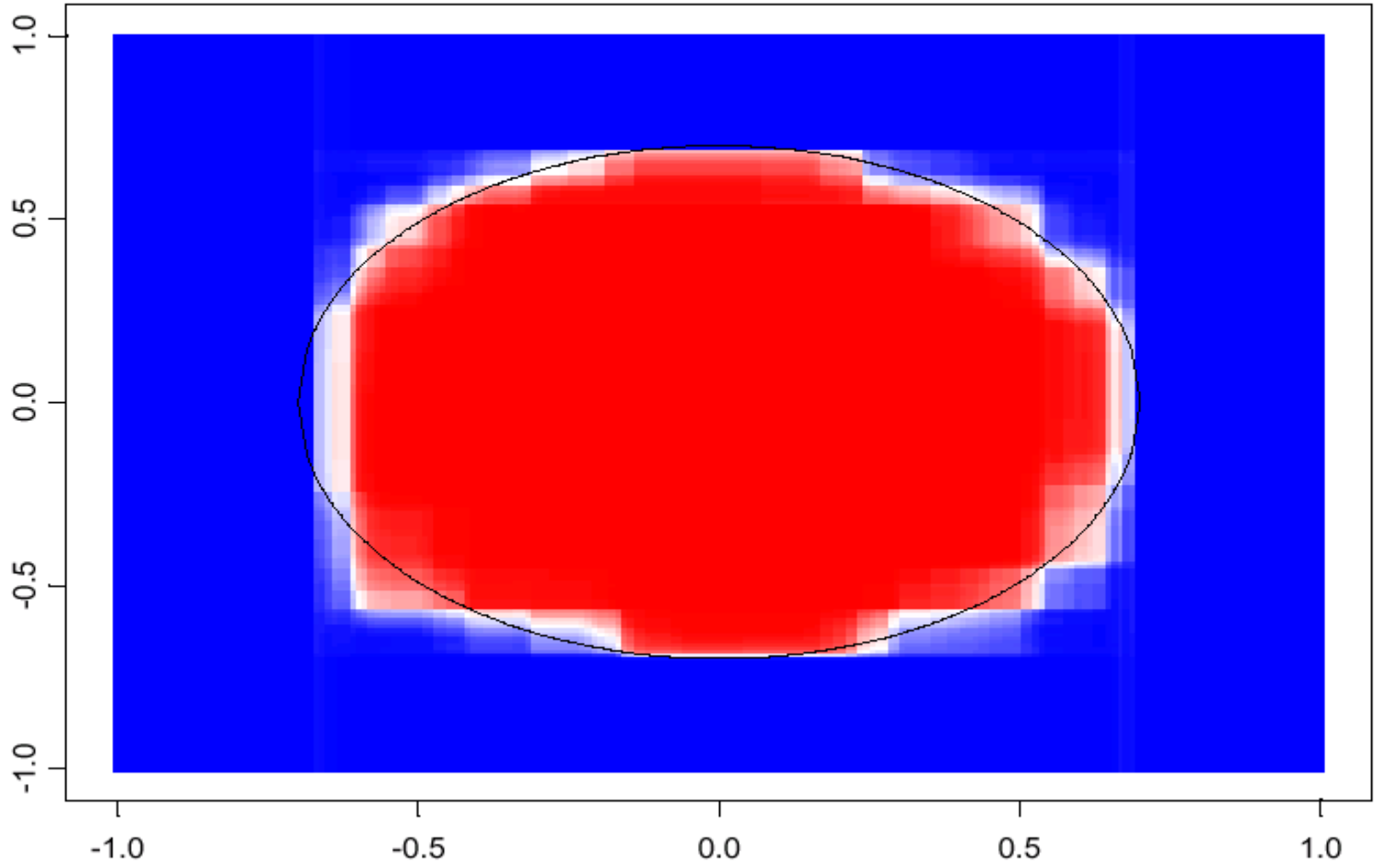
Bagging Example



CART decision boundary

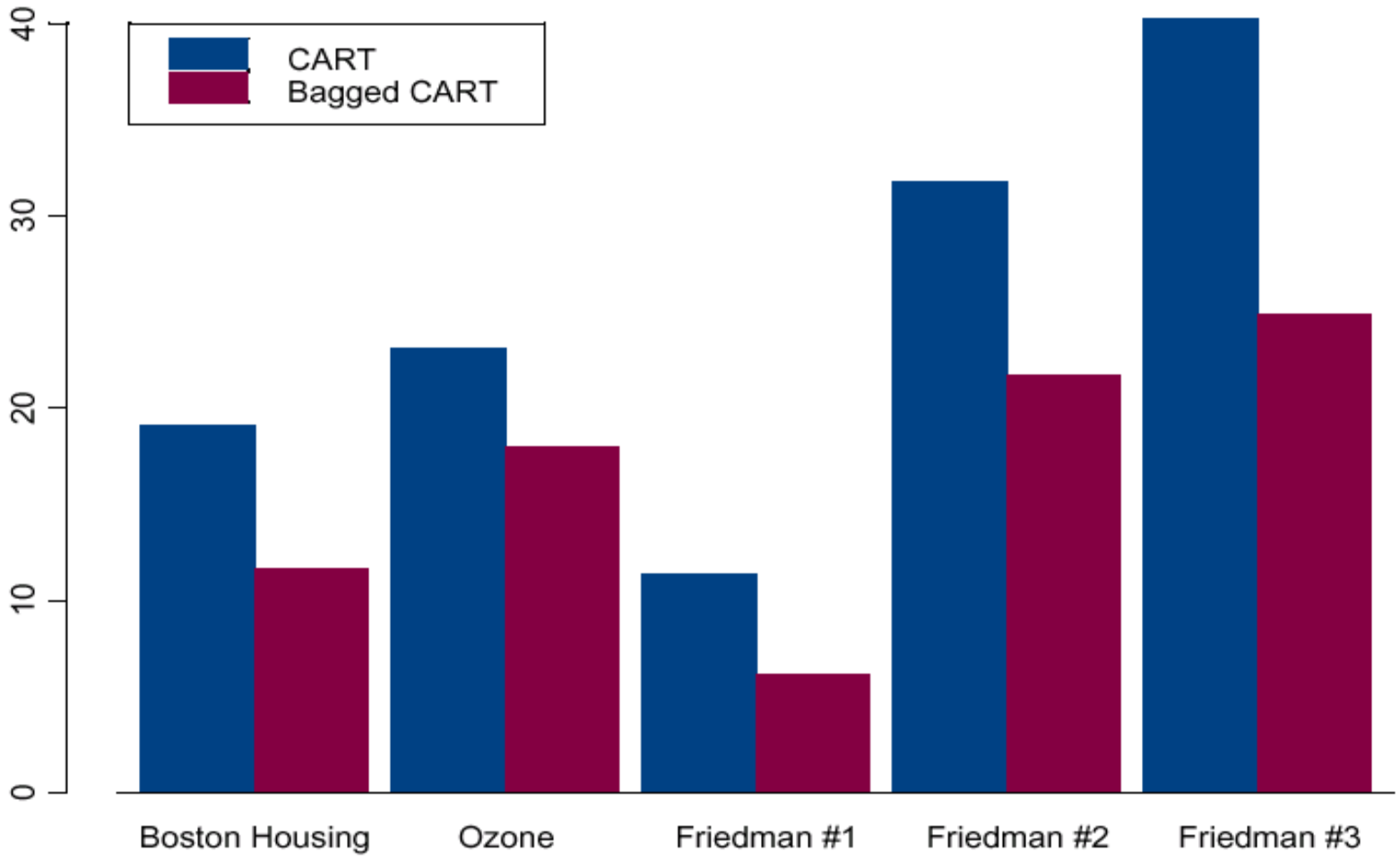


100 bagged trees



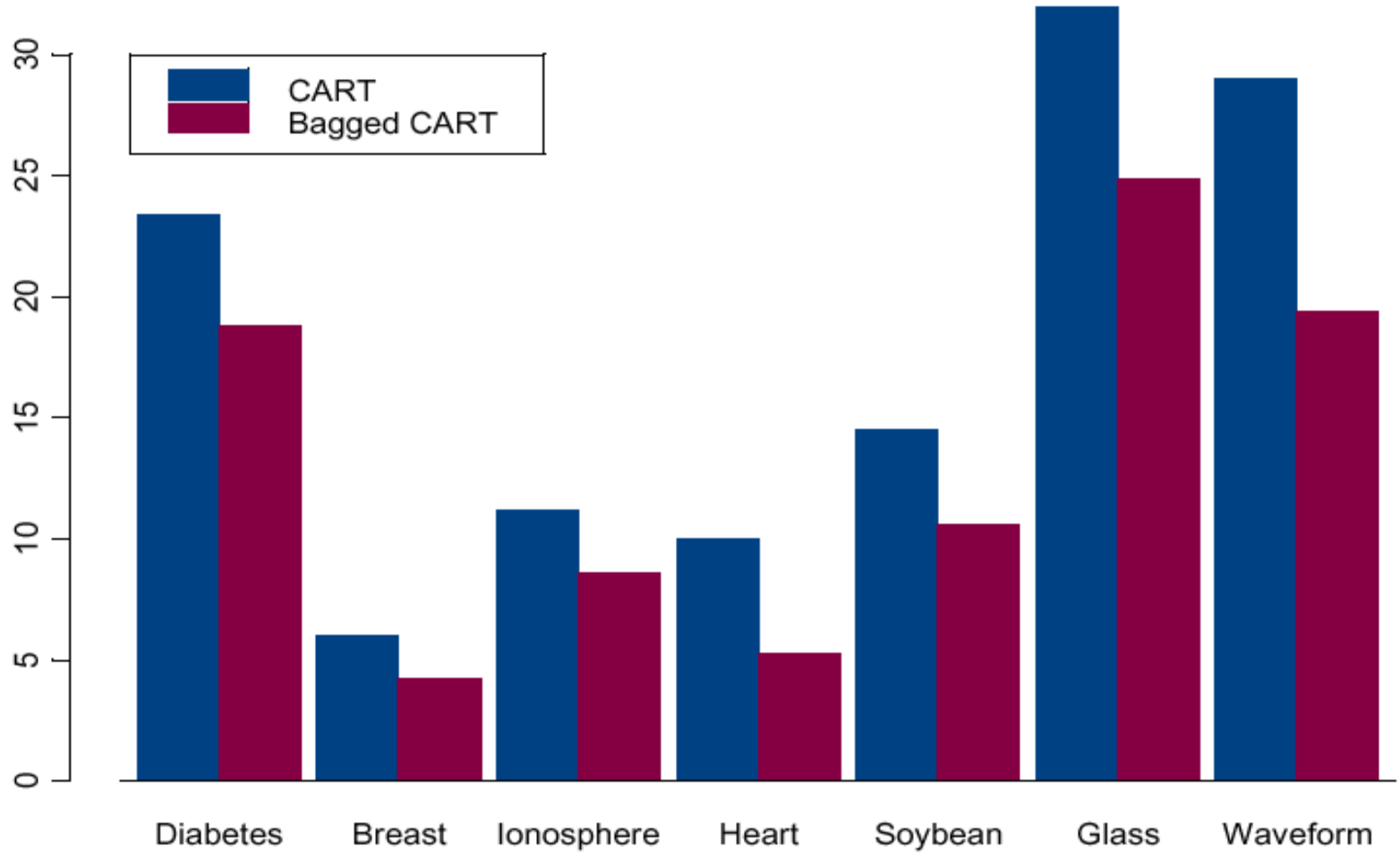
Regression results

Squared error loss



Classification results

Misclassification rates



Random Forests

“ The key to accuracy is low correlation and bias. To keep bias low, trees are grown to maximum depth.

To keep correlation low, the current version uses this randomization.

1) Each tree is grown on a bootstrap sample of the training set.

2) A number m is specified much smaller than the total number of variables M. At each node, m variables are selected at random out of the M, and the split is the best split on these m variables. ”

(see Random Forests , Machine Learning(2001) 45 5-320)

An important feature is that it carries along an internal test set estimate of the prediction error.

For every tree grown, about one-third of the cases are out-of-bag (out of the bootstrap sample). Abbreviated oob.

Put these oob cases down the corresponding tree and get response estimates for them.

For each case n, average or pluralize the response estimates over all time that n was oob to get a test set estimate $\hat{y}_n$ for y_n.

Averaging the loss over all n give the test set estimate of prediction error.

``

Table 3 Test Set Errors (%)

Data Set	Adaboost	Forest-RC		
		Selection	Two Features	One Tree
glass	22.0	24.4	23.5	42.4
breast cancer	3.2	3.1	2.9	5.8
diabetes	26.6	23.0	23.1	32.1
sonar	15.6	13.6	13.8	31.7
vowel	4.1	3.3	3.3	30.4
ionosphere	6.4	5.5	5.7	14.2
vehicle	23.2	23.1	22.8	39.1
German credit	23.5	22.8	23.8	32.6
image	1.6	1.6	1.8	6.0
ecoli	14.8	12.9	12.4	25.3
votes	4.8	4.1	4.0	8.6
liver	30.7	27.3	27.2	40.3
letters	3.4	3.4	4.1	23.8
sat-images	8.8	9.1	10.2	17.3
zip-code	6.2	6.2	7.2	22.7
waveform	17.8	16.0	16.1	33.2
twonorm	4.9	3.8	3.9	20.9
threenorm	18.8	16.8	16.9	34.8
ringnorm	6.9	4.8	4.6	24.6

Adaptive Bagging

Goal: Bias and variance reduction

Method: Sequentially fit *bagged* models, where each fits the current residuals

Properties:

- Bias and variance reduction
- No tuning parameters

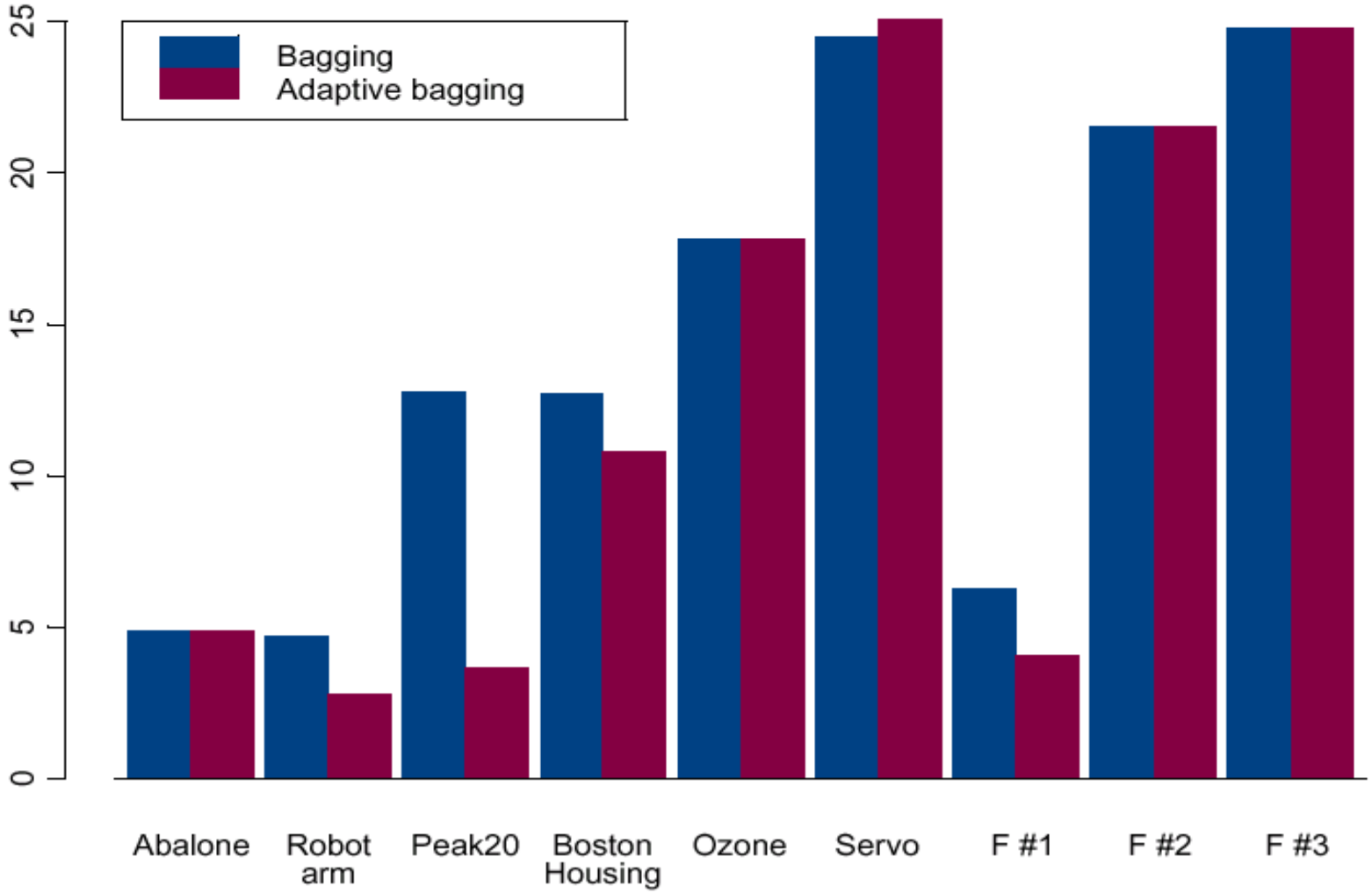
Adaptive bagging algorithm

1. Fit a bagged regressor to the dataset D.
2. Predict "out-of-bag" observations.
3. Fit a new bagged regressor to the bias (error) and repeat.

For a new observation, sum the predictions from each stage.

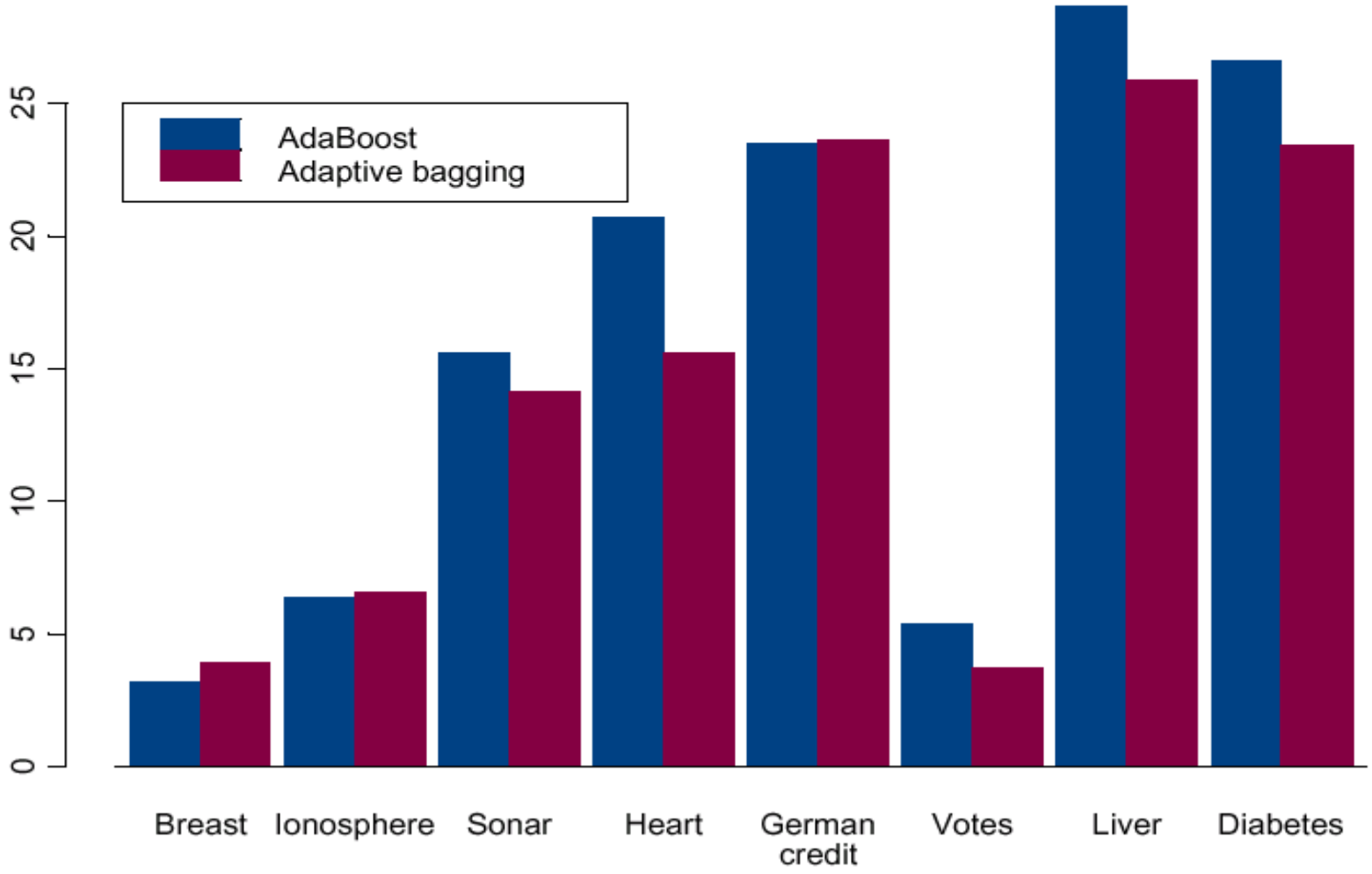
Regression results

Squared error loss



Classification results

Misclassification rates



Bagging References

- Leo Breiman's homepage
  www.stat.berkeley.edu/users/breiman/
- Breiman, L. (1996) "Bagging Predictors," *Machine Learning*, 26:2, 123-140.
- Friedman, J. and P. Hall (1999) "On Bagging and Nonlinear Estimation"
  www.stat.stanford.edu/~jhf

Peter Buhlmann and Bin Yu. Explaining bagging. Can be downloaded from http://stat.ethz.ch/~buhlmann/bibliog.html, September 2000.

J.H. Friedman and O. Hall. On bagging and nonlinear estimation. Can be downloaded from http://www-stat.stanford.edu/~jhf/#reports, May 2000.

Andreas Buja's home page:

``The Effect of Bagging on Variance, Bias and Mean Squared Error''
A. Buja, W. Stuetzle.
Bootstrap aggregation ("bagging") is a device for reducing the variance of learning algorithms. We give a complete second-order analysis of the effect of bagging on finite sums of U-statistics.

``Smoothing Effects of Bagging''
A. Buja, W. Stuetzle.
An short note on bagging. It relates the von Mises expansion of a bagged statistical functional to the Efron-Stein ANOVA expansion of the unbagged functional to show that the bagged functional is always smooth.

Boosting

Goal: Improve misclassification rates

Method: Sequentially fit models, each more heavily weighting those observations poorly predicted by the previous model

Properties:

- Bias and variance reduction
- Easy to implement
- Theory is not fully (but almost) explained

Generic boosting algorithm

Equally weight the observations $(y,\boldsymbol{x})_i$

For t in $1,\ldots,T$

- Using the weights, fit a classifier $f_t(\boldsymbol{x}) \rightarrow y$
- Upweight the poorly predicted observations
- Downweight the well-predicted observations

Merge $f_1,\ldots,f_T$ to form the boosted classifier

Real AdaBoost

Schapire & Singer 1998

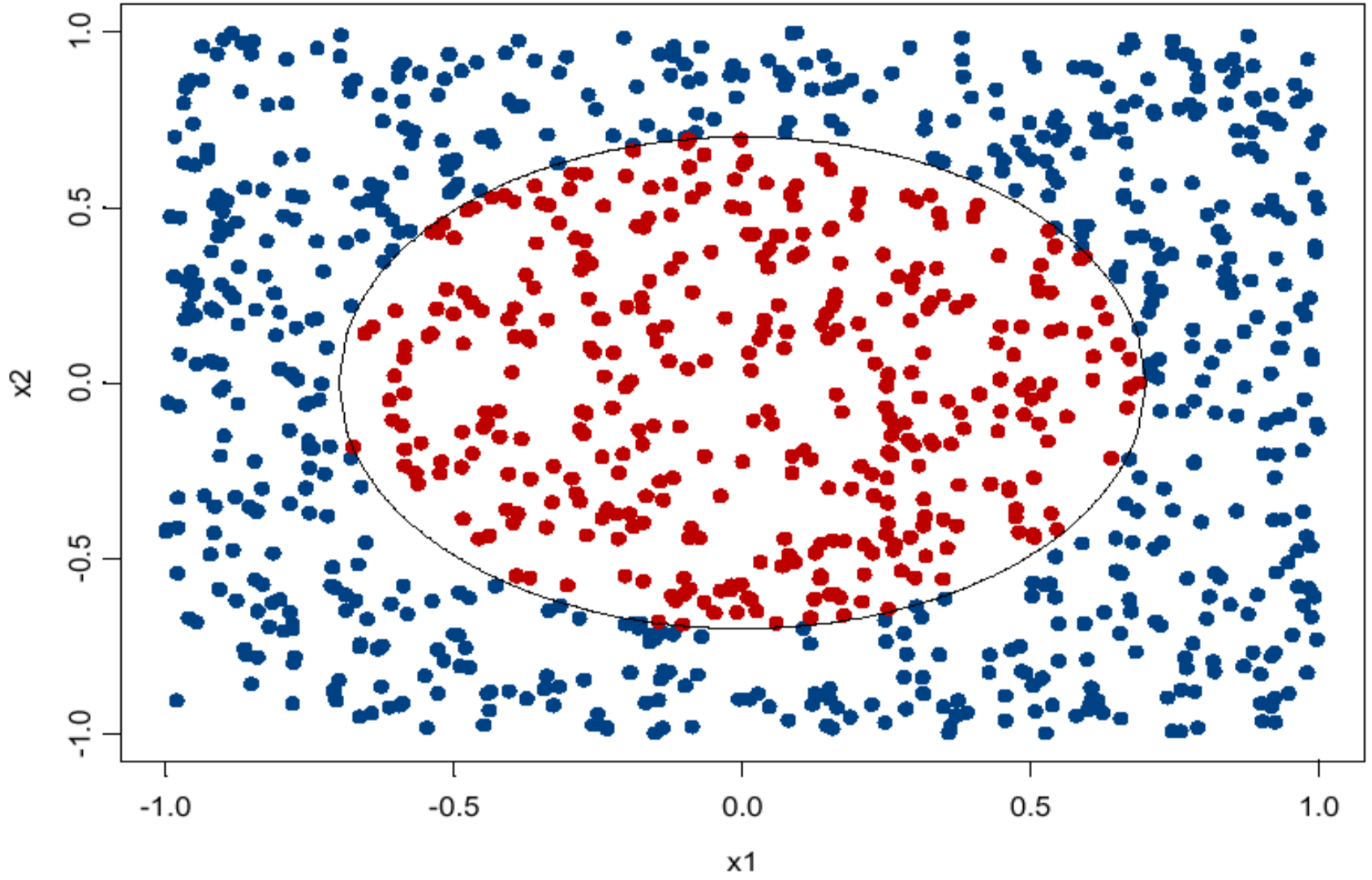
$y_i \in \{-1,1\}$, $w_i = 1/N$

For t in $1,\ldots,T$ do

1. Estimate $\mathrm{P}_w(y = 1|\boldsymbol{x})$.
2. Set $f_t(\boldsymbol{x}) = \frac{1}{2}\log\frac{\hat{\mathrm{P}}_w(y=1\,|\,\boldsymbol{x})}{\hat{\mathrm{P}}_w(y=-1\,|\,\boldsymbol{x})}$
3. $\boldsymbol{w}_i \leftarrow w_i \exp\left(-y_i f_t(\boldsymbol{x}_i)\right)$ and renormalize

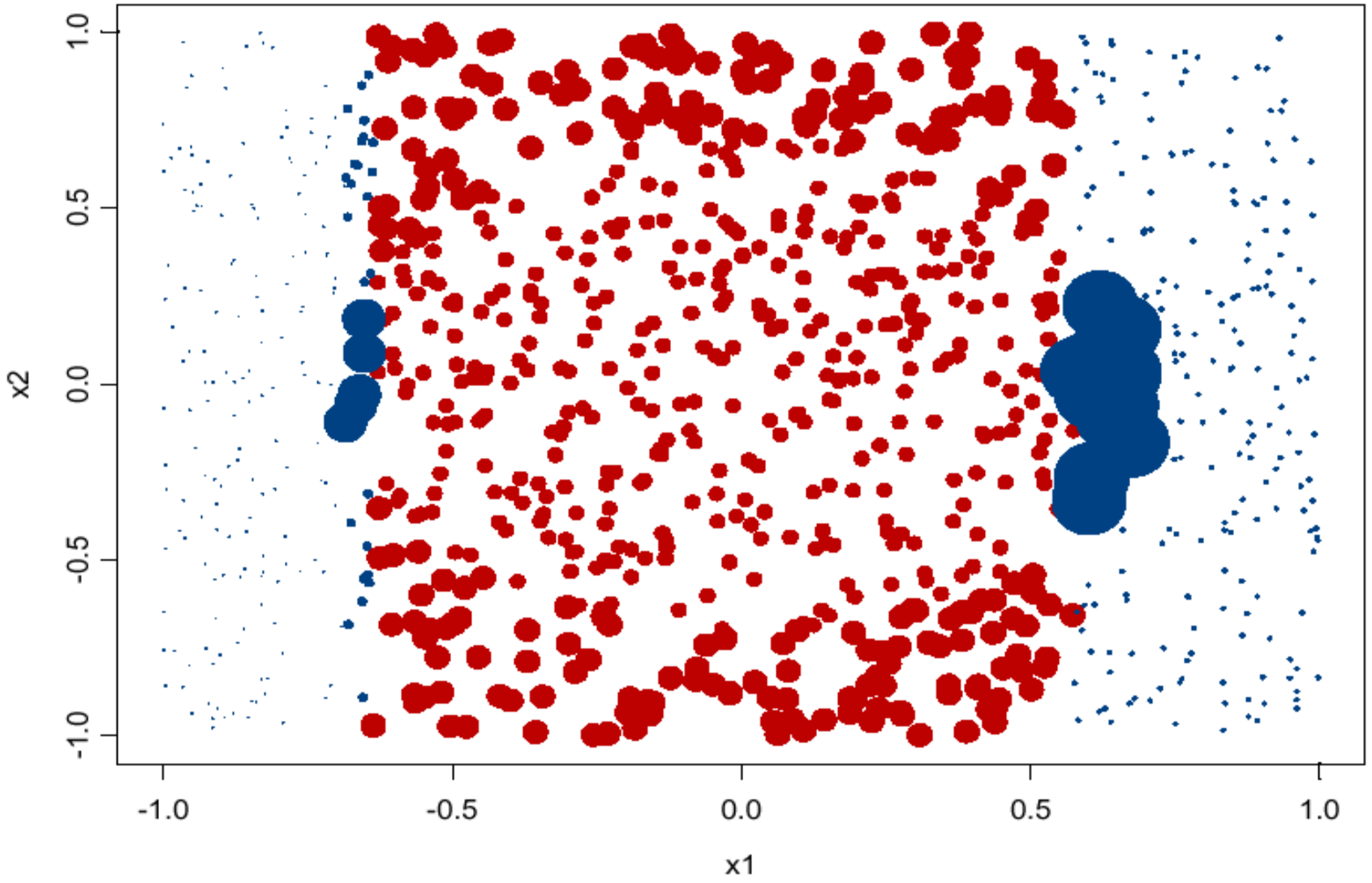
Output the classifier $\boldsymbol{F}(\boldsymbol{x}) = \mathrm{sign}\left(\sum f_t(\boldsymbol{x})\right)$

Boosting Example

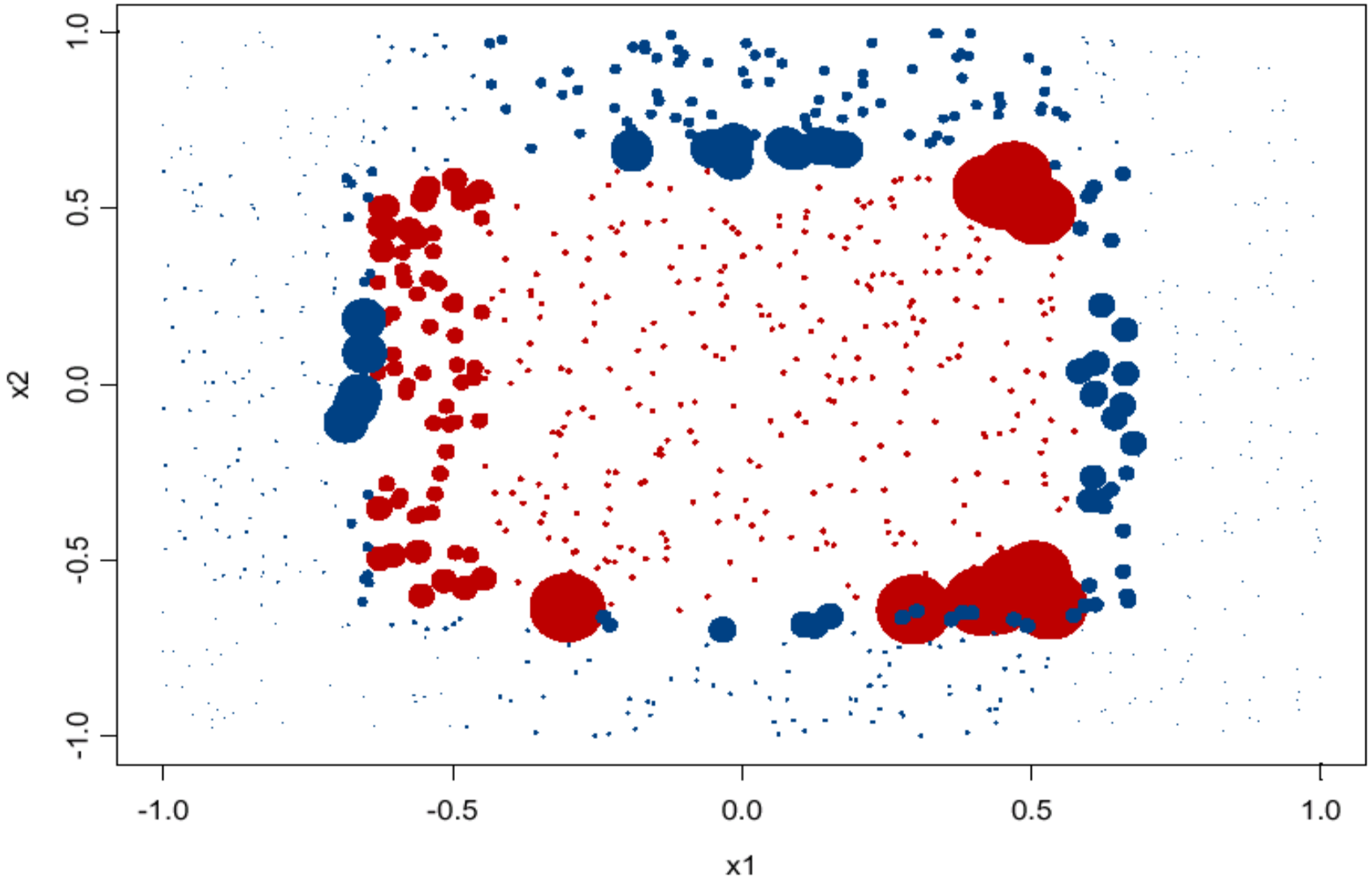


After one iteration

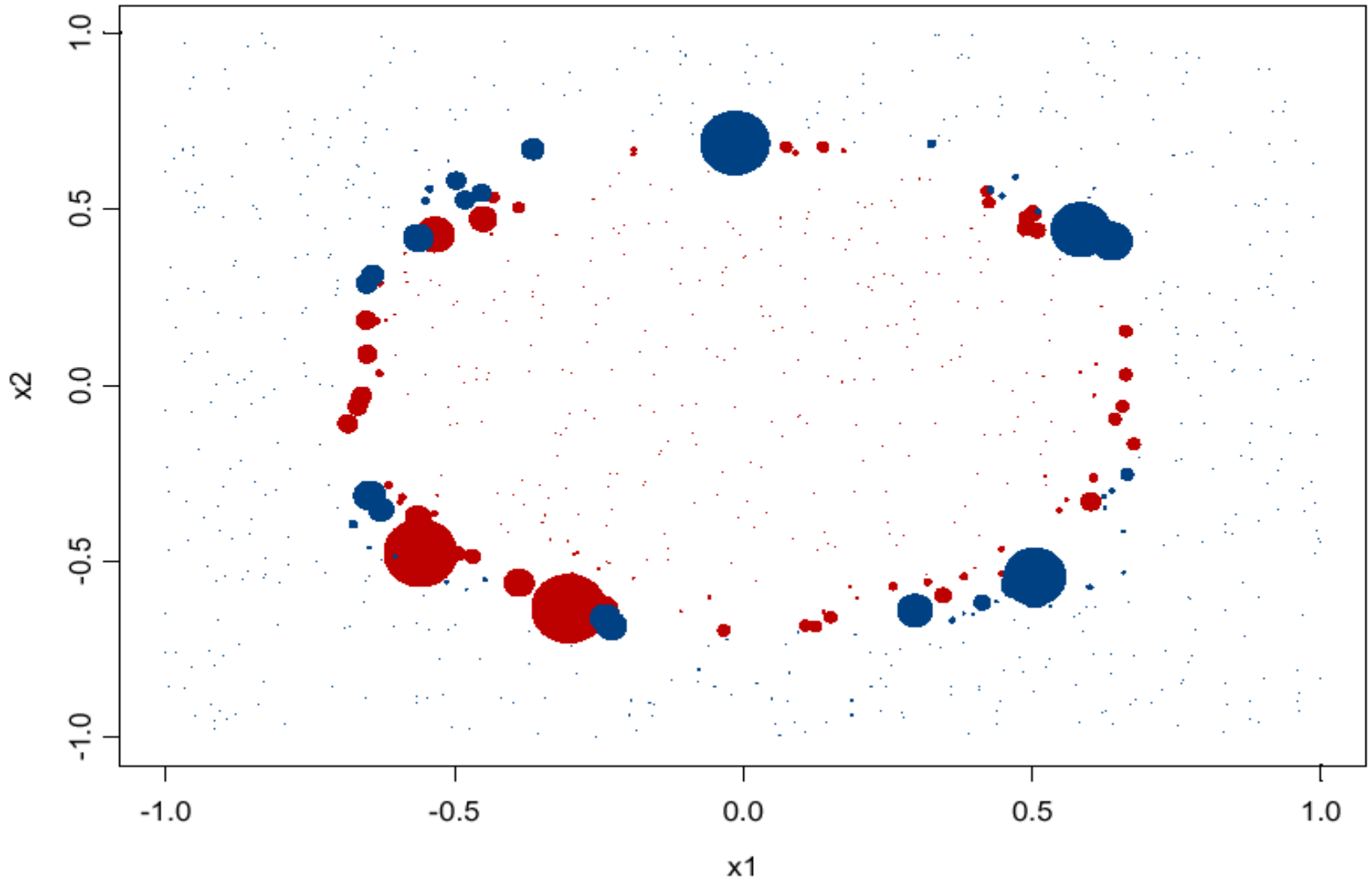
CART splits, larger points have great weight



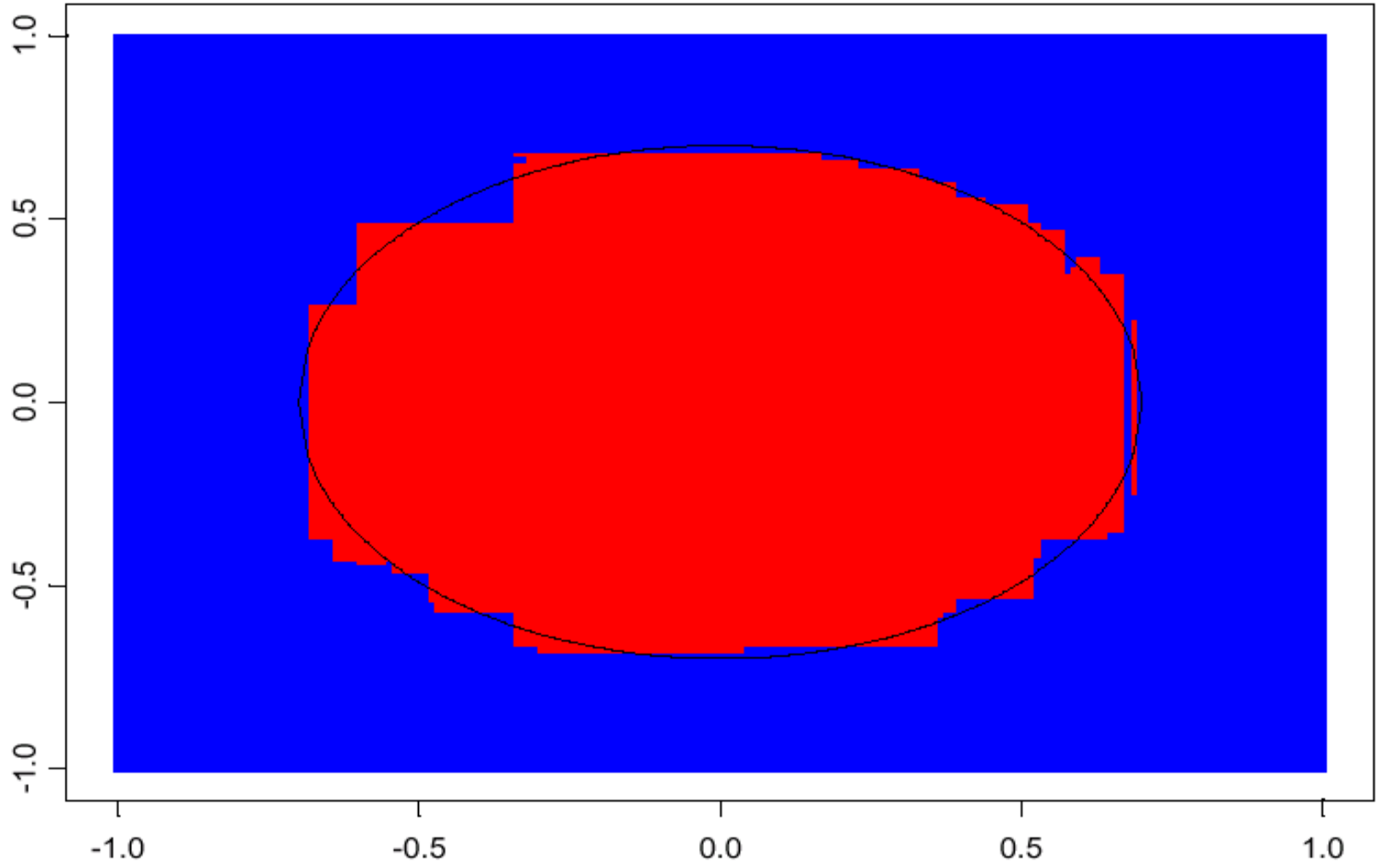
After 3 iterations



After 20 iterations



Decision boundary after 100 iterations



Boosting as optimization

- Friedman, Hastie, Tibshirani [1998] - AdaBoost is an optimization method for finding a classifier.
- Let $y \in \{-1,1\}$, $F(x) \in (-\infty,\infty)$

$$J(F) = E\left(e^{-yF(x)} \mid x\right)$$

Criterion

- $E(e^{-yF(x)})$ bounds the misclassification rate.

$$I(yF(x) < 0) < e^{-yF(x)}$$

- The minimizer of $E(e^{-yF(x)})$ coincides with the maximizer of the expected Bernoulli likelihood.

$$J(F) = \mathrm{E}\ell(F) = \mathrm{E}\left[y^* F(\mathbf{x}) - \log\left(1 + e^{F(\mathbf{x})}\right) | \mathbf{x}\right]$$

$$y^* = \tfrac{1}{2}(1+y) \in \{0, 1\}$$

Optimization step

$$J(F+f) = E\left(e^{-y(F(x)+f(x))} \mid x\right)$$

- Select f to minimize J…

$$F^{(t+1)} \leftarrow F^{(t)} + \tfrac{1}{2}\log\frac{E_w[I(y=1)\mid x]}{1-E_w[I(y=1)\mid x]}$$

$$w(x,y) = e^{-yF^{(t)}(x)}$$

Let $J(F) = E[e^{-yF(x)}]$. Suppose we have a current estimate $F(x)$ and seek an improved estimate $F(x) + cf(x)$. For fixed c (and x), we expand $J(F(x) + cf(x))$ to second order about $f(x) = 0$

$$\begin{aligned} J(F + cf) &= E[e^{-y(F(x)+cf(x))}] \\ &\approx E[e^{-yF(x)}(1 - ycf(x) + c^2y^2f(x)^2/2)] \\ &= E[e^{-yF(x)}(1 - ycf(x) + c^2/2)] \end{aligned}$$

since y^2=1 and $f(x)^2 = 1$. Minimizing pointwise with respect to $f(x) \in \{-1, 1\}$, we write

$$f(x) = \arg\min_f E_w(1 - ycf(x) + c^2/2|x) \tag{16}$$

Here the notation $E_w(\cdot|x)$ refers to a *weighted conditional expectation*, where $w = w(x, y) = e^{-yF(x)}$, and

$$E_w[g(x, y)|x] \stackrel{\text{def}}{=} \frac{E[w(x, y)g(x, y)|x]}{E[w(x, y)|x]}.$$

For $c > 0$, minimizing (16) is equivalent to maximizing

$$E_w[yf(x)] \tag{17}$$

The solution is

$$f(x) = \begin{cases} 1 & \text{if } E_w(y|x) = P_w(y = 1|x) - P_w(y = -1|x) > 0 \\ -1 & \text{otherwise} \end{cases} \tag{18}$$

LogitBoost

Friedman, Hastie, Tibshirani [1998]

- Logistic regression

$$y = \begin{cases} 1 & \text{with probability } p(x) \\ 0 & \text{with probability } 1 - p(x) \end{cases}$$

$$p(x) = \frac{1}{1 + e^{-F(x)}}$$

- Expected log-likelihood of a regressor, $F(x)$

$$\mathrm{E}\,\ell(F) = \mathrm{E}\left(yF(x) - \log(1 + e^{F(x)}) \mid x\right)$$

Newton steps

$$J(F+f) = E\left(y(F(x)+f(x)) - \log(1+e^{F(x)+f(x)}) \mid x\right)$$

- Iterate to optimize expected log-likelihood.

$$F^{(t+1)}(x) \leftarrow F^{(t)}(x) - \frac{\frac{\partial}{\partial f} J(F^{(t)}+f)\Big|_{f=0}}{\frac{\partial^2}{\partial f^2} J(F^{(t)}+f)\Big|_{f=0}}$$

LogitBoost, continued

- Newton steps for Bernoulli likelihood

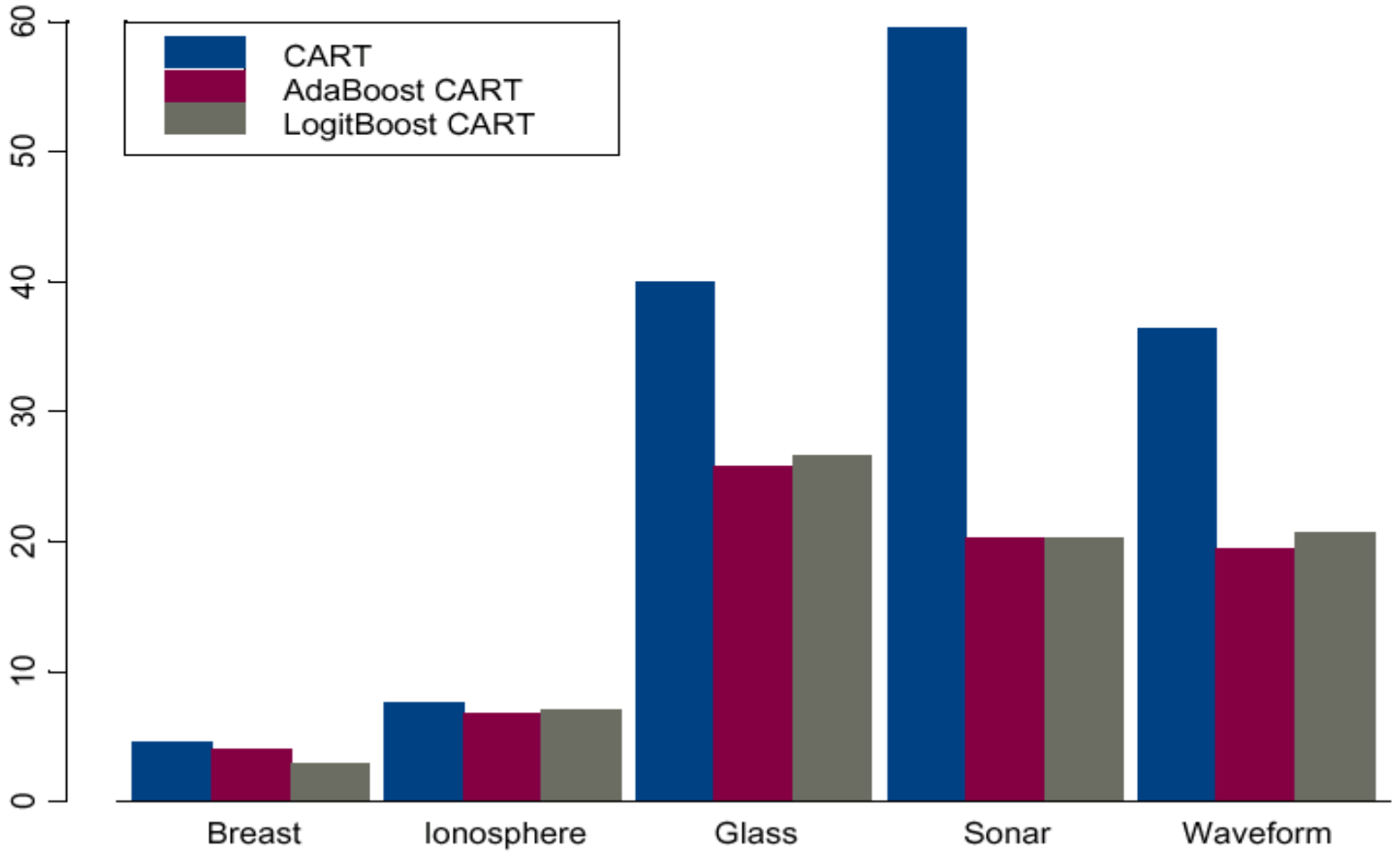
$$F(x) \leftarrow F(x) + E_w\left(\frac{y - p(x)}{p(x)(1-p(x))} \middle| x \right)$$

$$w(x) = p(x)(1-p(x))$$

- In practice the $E_w(\bullet|x)$ can be any regressor - trees, smoothers, etc.
- Trees are adaptive and work well for high dimensional data.

Misclassification rates

Friedman, Hastie, Tibshirani [1998]



Naïve Bayes Classification

Probabilistic Classification

$$P(Y = y \mid X_1 = x_1, \ldots, X_d = x_d) = \frac{P(\underline{X} \mid Y = y)P(Y = y)}{P(\underline{X})}$$

The naïve Bayes assumption

$$P(\underline{X} \mid Y = y) = P(X_1 = x_1 \mid Y = y) \cdots P(X_d = x_d \mid Y = y)$$

Estimation

- Probability estimates are trivial

$$\hat{P}(X_j = x_j \mid Y = y) = \frac{\text{count}(X_j = x_j \cap Y = y)}{\text{count}(X_j = x_j)}$$

- Estimation is linear in the number of predictors and the number of observations

Interpretability

Consider the log-odds in favor of Y=1

$$\log \frac{P(Y=1 \mid \underline{X})}{P(Y=0 \mid \underline{X})} = w_0 + \sum_{j=1}^{d} w_j(X_j)$$

- Positive w_j are evidence in favor of Y=1
- Negative w_j are evidence in favor of Y=0

Evidence balance sheets

Evidence in favor of knee surgery		**Evidence against knee surgery**	
Female	+8	Prior evidence	-10
Knee is unstable	+88	Age 50	-12
Knee locks	+172	No effusion	-62
Tender med JL	+49	Negative McMurray's	-38
Total positive evidence	+317	Total negative evidence	-122
Total evidence		+195	
Probability of knee surgery		88%	

Boosting algorithms

1. Learn a classifier from the data
2. Upweight observations poorly predicted, downweight observations well predicted
3. Refit the model using the new weighting
4. After T iterations, have each model vote on the final prediction.

AdaBoost algorithm

Freund & Shapire (1997)

- AdaBoost defines a particular reweighting scheme and a voting method for merging the classifiers
- AdaBoost decreases bias and variance in many settings - Bauer and Kohavi [1998]
- Boosted naïve Bayes tied for first place in the 1997 KDD Cup

AdaBoost

- Extremely dense voting scheme

$$P(Y=1\mid x)=\frac{1}{1+\prod_{t=1}^{T}\beta_t^{2r(x)-1}} \qquad r(x)=\frac{\sum_{t=1}^{T}(\log\frac{1}{\beta_t})P_t(Y=1\mid x)}{\sum_{t=1}^{T}(\log\frac{1}{\beta_t})}$$

- Destroys interpretability

Regaining Interpretability

Rewriting the voting scheme…

$$\log \frac{P(Y=1 \mid X)}{P(Y=0 \mid X)} = \sum_{t=1}^{T} (\log \beta_t) \left(1 - 2 \left(1 + e^{-\log \frac{P_t(Y=1|X)}{P_t(Y=0|X)}} \right)^{-1} \right)$$

Substitute Taylor expansion…

$$\frac{1}{1+e^{-x}} = \tfrac{1}{2} + \tfrac{1}{4} x - \tfrac{1}{48} x^3 + O\left(x^5\right)$$

Regained Interpretability

$$\sum_{t=1}^{T} \alpha_t \log \frac{P_t(Y=1)}{P_t(Y=0)} + \sum_{j=1}^{d} \sum_{t=1}^{T} \alpha_t \log \frac{P_t(X_j \mid Y=1)}{P_t(X_j \mid Y=0)}$$

$$= \text{boosted prior weight of evidence} +$$

$$\sum_{j=1}^{d} \text{boosted weight of evidence from } X_j$$

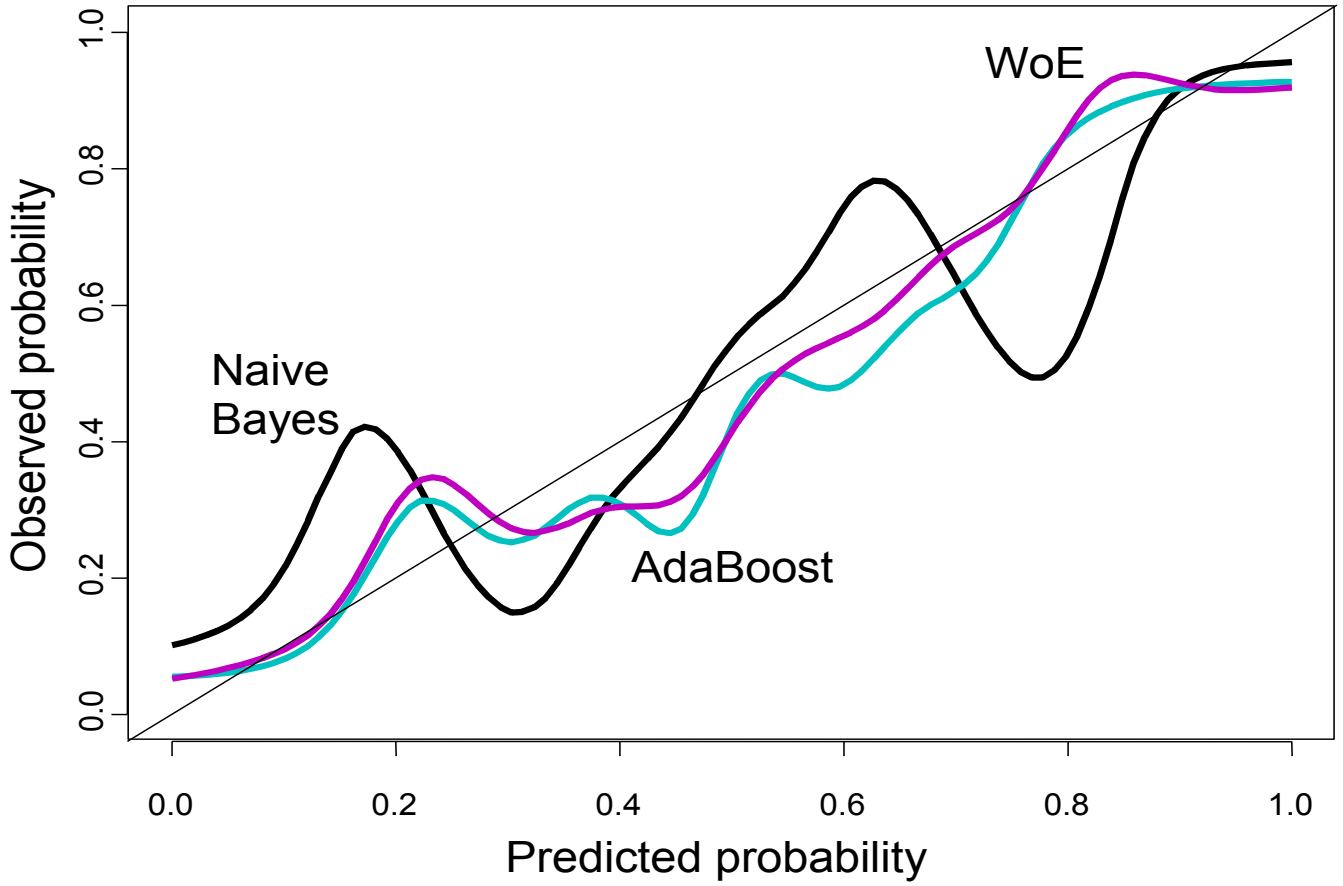
- **Boosting biases parameter estimates**
- **Adjusts naïve Bayes' for over-optimism**

Misclassification rates

	Naïve Bayes	AdaBoost	Weight of evidence
Knee diagnosis	14.0%	13.8%	13.4%
Diabetes	25.0%	24.4%	24.4%
Credit approval	16.8%	15.5%	15.5%
CAD	18.4%	18.3%	18.3%
Breast tumors	3.9%	3.8%	3.8%

- Boosting offers modest improvement
- Actual AdaBoost and approximation are close

Calibration



Boosting References

- Rob Schapire's homepage
  www.research.att.com/~schapire
- Freund, Y. and R. Schapire (1996). "Experiments with a new boosting algorithm," Machine Learning: Proceedings of the 13th International Conference, 148-156.
- Jerry Friedman's homepage
  www.stat.stanford.edu/~jhf
- Friedman, J., T. Hastie, R. Tibshirani (1998). "Additive Logistic Regression: a statistical view of boosting," Technical report, Statistics Department, Stanford University.