Combining Estimators to Improve Performance

A survey of “model bundling” techniques -- from boosting and bagging, to Bayesian model averaging -- creating a breakthrough in the practice of Data Mining.

John F. Elder IV, Ph.D.
Elder Research, Charlottesville, Virginia
www.datamininglab.com

Greg Ridgeway, Ph.D.
University of Washington, Dept. of Statistics
www.stat.washington.edu/greg

Outline

- Why combine? A motivating example
- Hidden dangers of model selection
- Reducing modeling uncertainty through *Bayesian Model Averaging*
- Stabilizing predictors through *bagging*
- Improving performance through *boosting*
- Emerging theory illuminates empirical success
- Bundling, in general
- Latest algorithms
- Closing Examples & Summary

Reasons to combine estimators

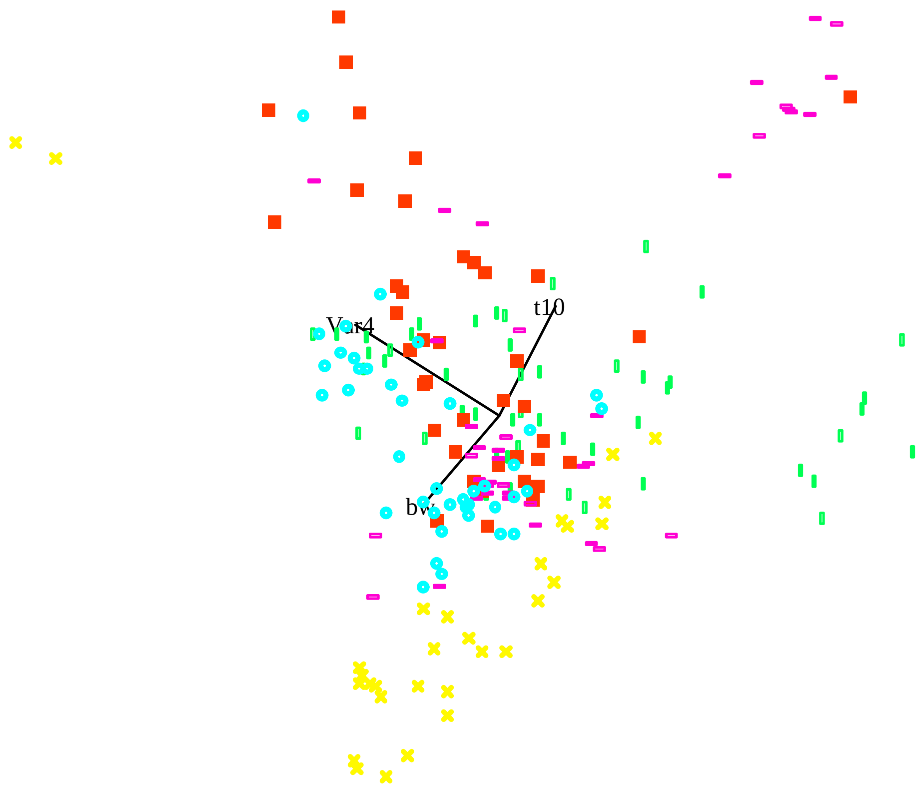
- Decreases variability in the predictions.
- Accounts for uncertainty in the model class.

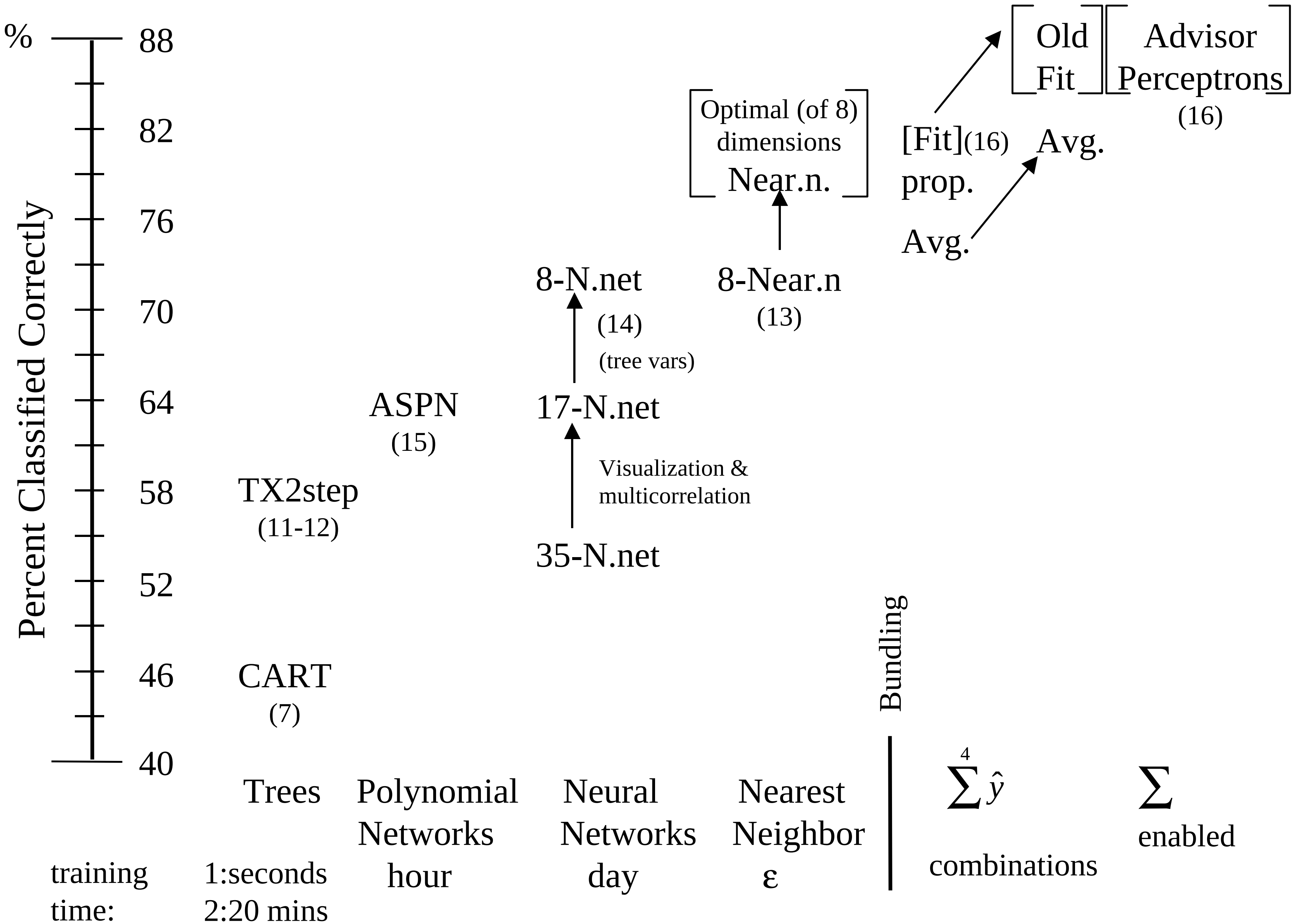
☆–> Improved accuracy on new data.

A Motivating Example: Classifying a bat's species from its chirp

- Goal: Use time-frequency features of echolocation signals to classify bats by species in the field (avoiding capture and physical inspection).
- U. Illinois biologists gathered data: 98 signals from 19 bats representing 6 species: Southeastern, Grey, Little Brown, Indiana, Pipistrelle, Big-Eared.
- ~35 data features (dimensions) calculated from signals, such as low frequency at the 3db level, time position of the signal peak, and amplitude ratio of 1st and 2nd harmonics.
- Turned out to have a nice level of difficulty for comparing methods: overlap in classes, but some separability.

Sample Projection




%
88
82
76
70
64
58
52
46
40
Percent Classified Correctly
CART
(7)
TX2step
(11-12)
ASPN
(15)
35-N.net
Visualization &
multicorrelation
17-N.net
(tree vars)
(14)
8-N.net
8-Near.n
(13)
Optimal (of 8)
dimensions
Near.n.
Bundling
Avg.
[Fit](16)
prop.
Avg.
Old
Fit
Advisor
Perceptrons
(16)
Trees
Polynomial
Networks
Neural
Networks
Nearest
Neighbor
$\sum^{4} \hat{y}$
combinations
$\sum$
enabled
training
time:
1:seconds
2:20 mins
hour
day
ε

What is model uncertainty?

- Suppose we wish to predict *y* from predictors *x*.
- Given a dataset of observations, *D*, for a new observation with predictors $\boldsymbol{x}^*$ we want to derive the predictive distribution of y^* given $\boldsymbol{x}^*$ and *D*.

$$\mathrm{P}(y^* \mid \boldsymbol{x}^*, D)$$

In practice…

- Although we want to use all the information in D to make the best estimate of y^* for an individual with covariates $\boldsymbol{x}^*$…

$$\mathrm{P}(y^* \mid \boldsymbol{x}^*, D)$$

- In practice, however, we always use

$$\mathrm{P}(y^* \mid \boldsymbol{x}^*, M)$$

where M is a model constructed from D.

Selecting *M*

- The process of selecting a model usually involves
  - Model class selection
    - Linear regression, tree regression, neural network
  - Variable selection
    - variable exclusion, transformation, smoothing
  - Parameter estimation
- We tend to choose the one model that fits the data or performs best as *the* model.

What’s wrong with that?

- Two models may equally fit a dataset (with repect to some loss) but have different predictions.
- Competing interpretable models with equivalent performance offer ambiguious conclusions.
- Model search dilutes the evidence. “Part of the evidence is spent specifying the model.”

Bayesian Model Averaging

Goal: Account for model uncertainty

Method: Use Bayes' Theorem and average the models by their posterior probabilities

Properties:

- Improves predictive performance
- Theoretically elegant
- Computationally costly

Averaging the models

Consider a set containing the K candidate models — $M_1,\ldots, M_K$.

With a few probability manipulations we can make predictions using all of them.

$$\mathrm{P}(y^* \mid \boldsymbol{x}^*, D) = \sum_k \mathrm{P}(y^* \mid \boldsymbol{x}^*, M_k)\,\mathrm{P}(M_k \mid D)$$

The probability mass for a particular prediction value of y is a weighted average of the probability mass that each model places on that value of y. The weight is based on the posterior probability of that model given the data.

Bayes' Theorem

$$P(M_k \mid D) = \frac{P(D \mid M_k)P(M_k)}{\sum_{l=1}^{K} P(D \mid M_l)P(M_l)}$$

- M_k - model
- D - data
- P($D|M_k$) - integrated likelihood of M_k
- P(M_k) - prior model probability

Challenges

- The size of the model set may cause exhaustive summation to be impossible.
- The integrated likelihood of each model is usually complex.
- Specifying a prior distribution (even a non-informative one) across the space of models is non-trivial.
- Proposed solutions to these challenges often involve MCMC, BIC approximation, MLE approximation, Occam's window, Occam's razor.

Performance

- Survival model: Primary biliary cirrhosis
  - BMA vs. Stepwise regression — 2% improvement
  - BMA vs. expert selected model — 10% improvement
- Linear regression: Body fat prediction
  - BMA provides best 90% predictive coverage.
- Graphical models
  - BMA yields an improvement

BMA References

- Chris Volinsky's BMA homepage *www.research.att.com/~volinsky/bma.html*
- J. Hoeting, D. Madigan, A. Raftery, C. Volinsky (1999). "Bayesian Model Averaging: A Practical Tutorial" (to appear in *Statistical Science*), *www.stat.colostate.edu/~jah/documents/bma2.ps*

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

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

Bias-variance decomposition (cont.)

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

- 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:

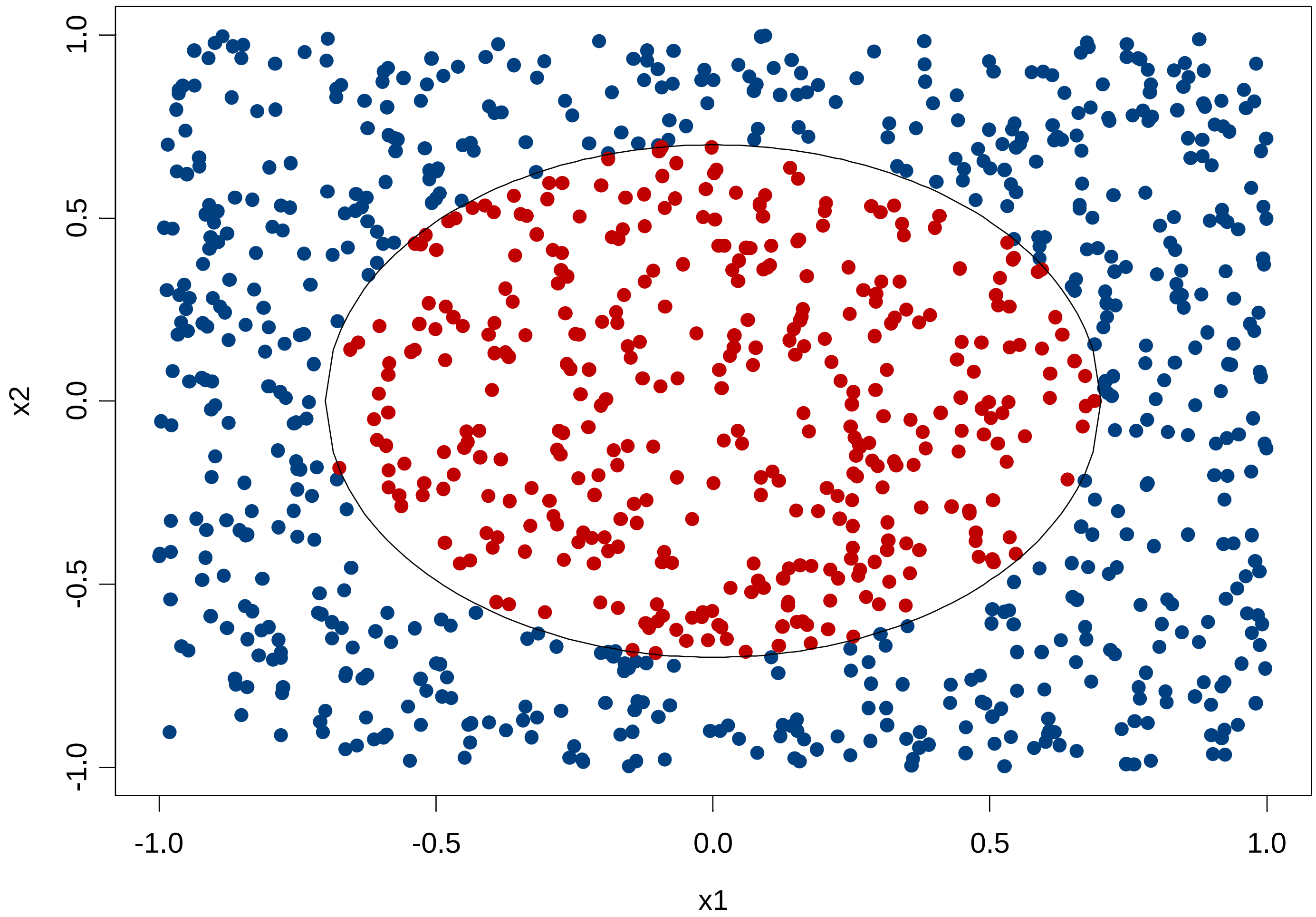
- Stabilizes “unstable” methods
- Easy to implement, parallelizable
- Theory is not fully explained

Bagging algorithm

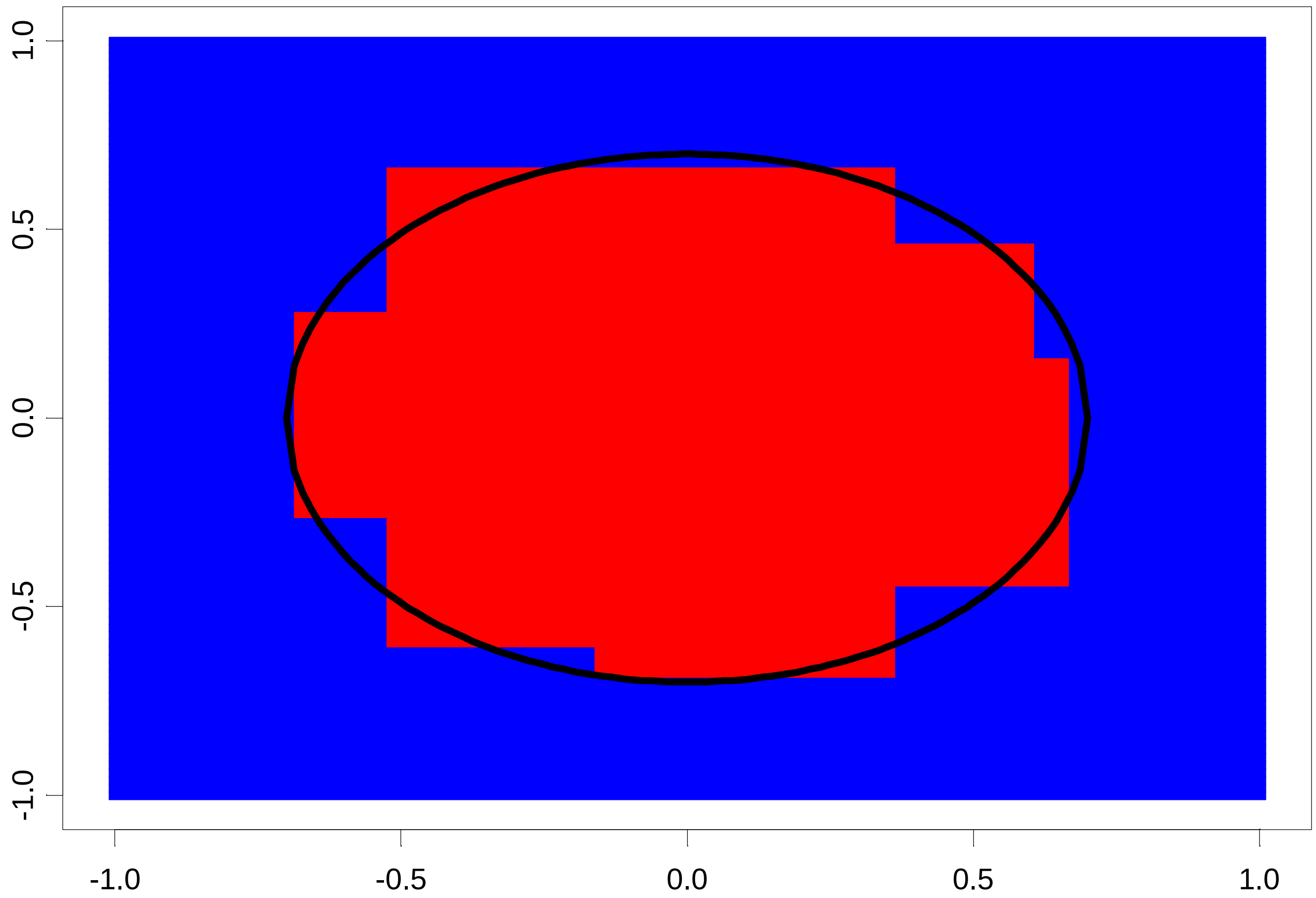
1. Create *K* bootstrap replicates of the dataset.
2. Fit a model to each of the replicates.
3. Average (or vote) the predictions of the *K* models.

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

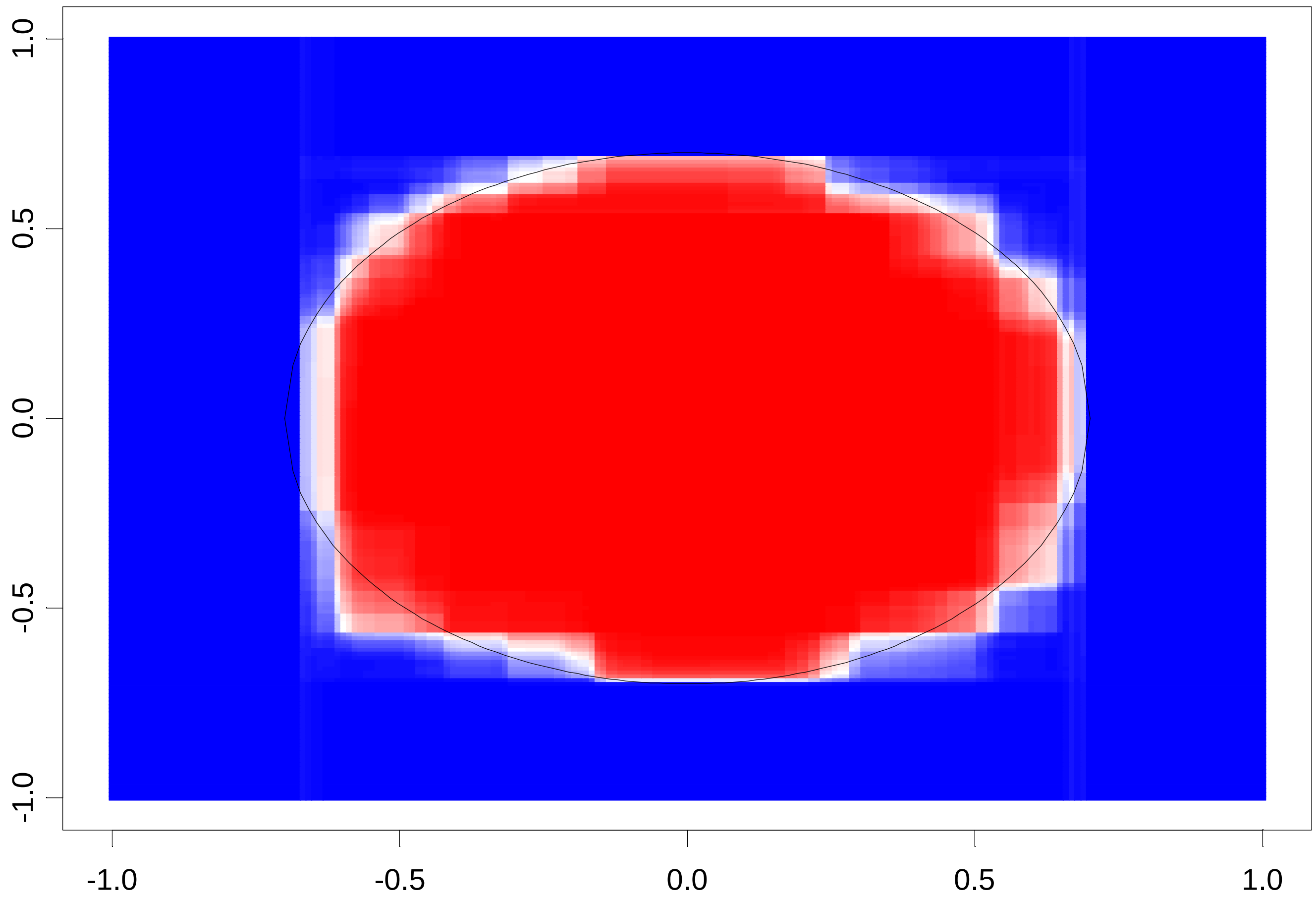
Bagging Example



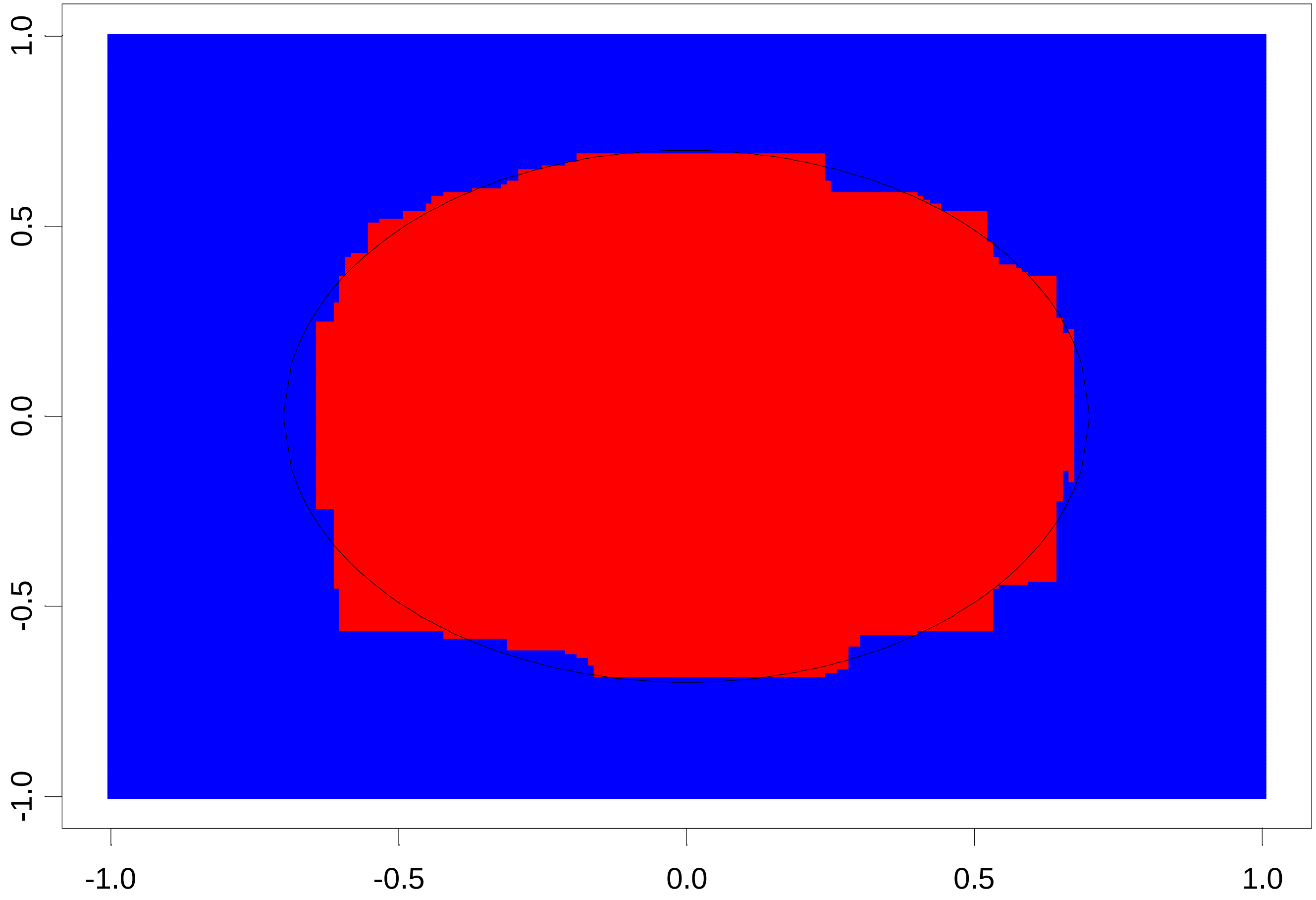
CART decision boundary



100 bagged trees

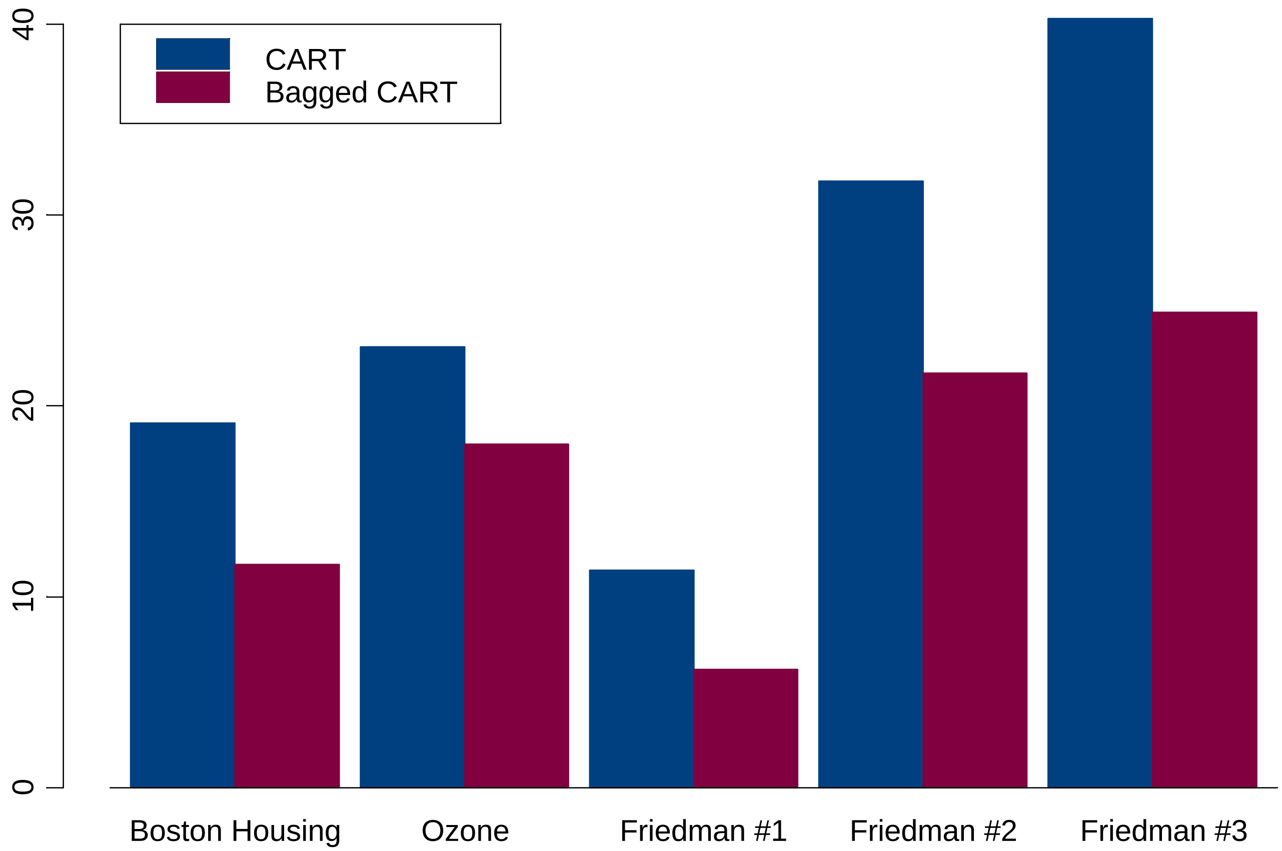


Bagged tree decision boundary



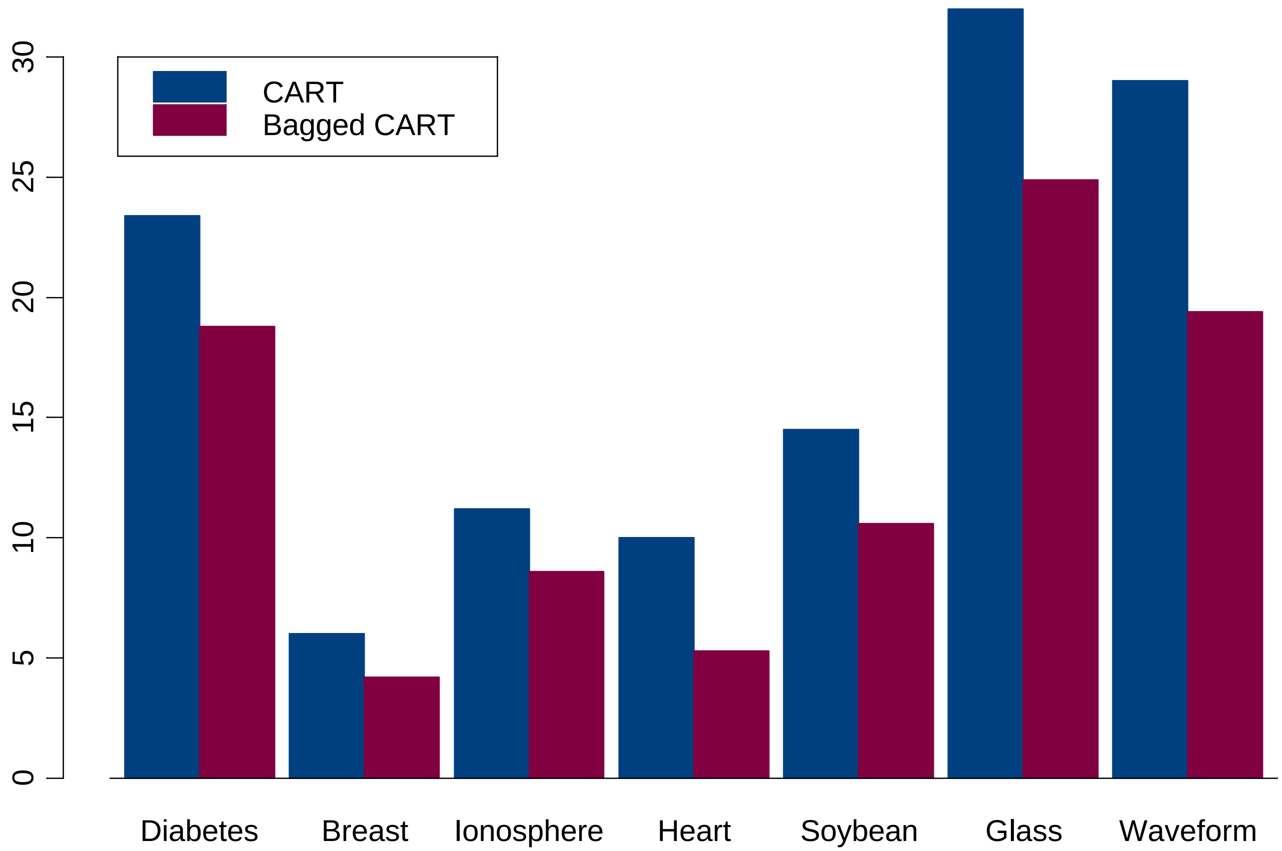
Regression results

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*

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

Origin of Boosting

Classification problems

$$\{y, \boldsymbol{x}\}_i \ , \ i = 1, \ldots, n$$

$$y \in \{0, 1\}$$

The task - construct a function,

$$F(\boldsymbol{x}) : \boldsymbol{x} \rightarrow \{0, 1\}$$

so that F minimizes misclassification error.

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

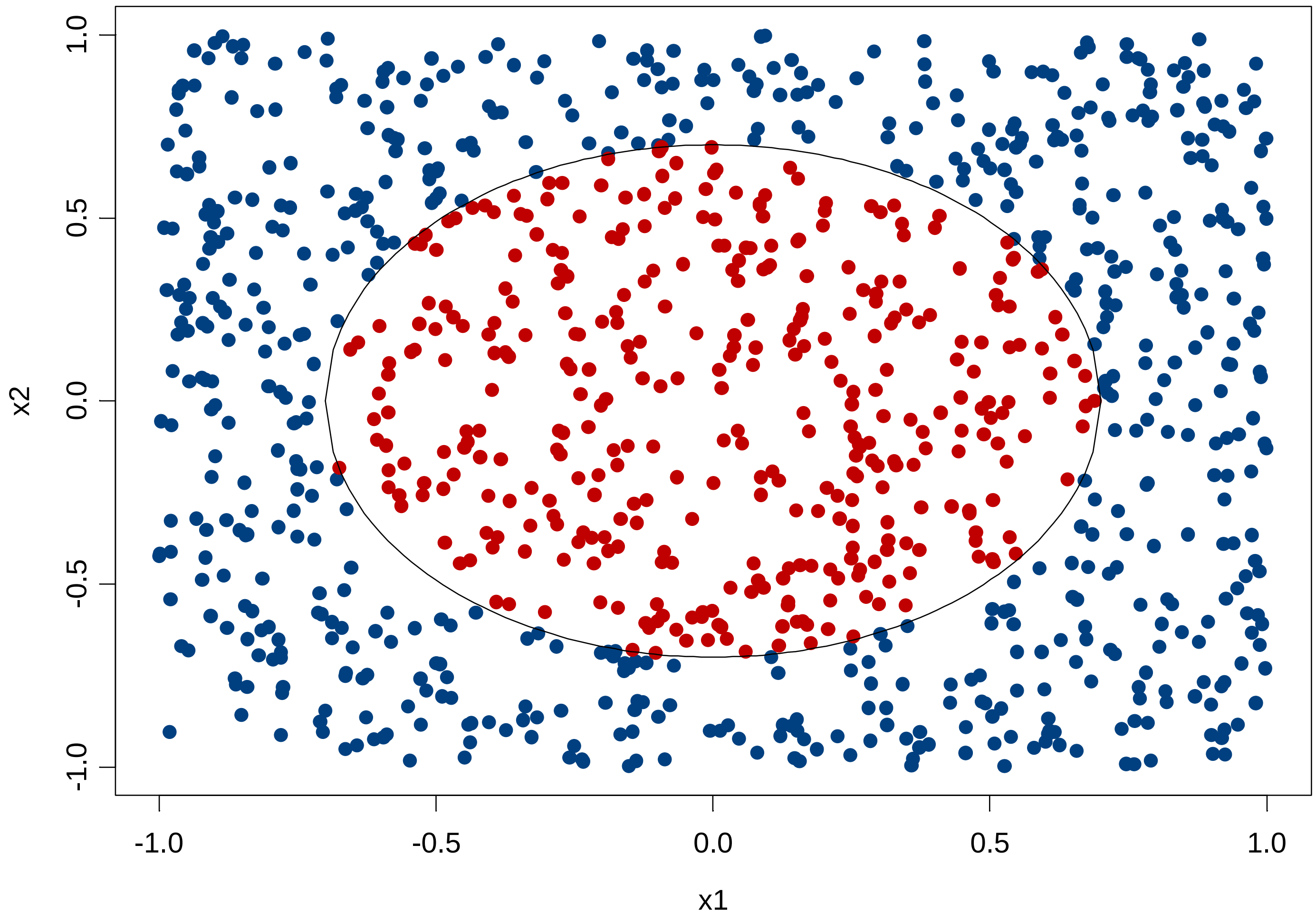
Output the classifier $\boldsymbol{F}(\boldsymbol{x}) = \text{sign}\left(\sum f_t(\boldsymbol{x})\right)$

AdaBoost’s Performance

Freund & Schapire [1996]

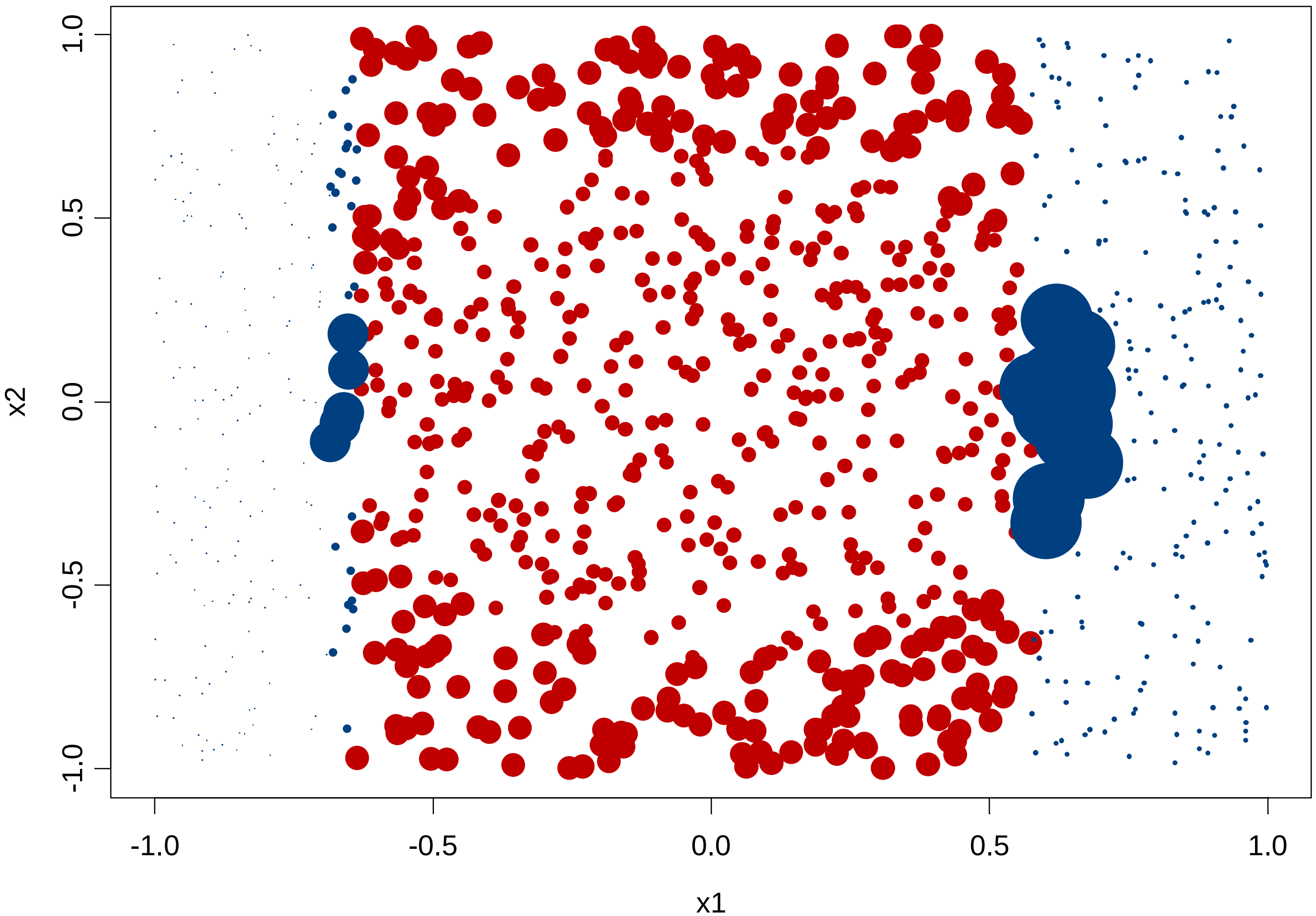
- Leo Breiman - AdaBoost with trees is the “best off-the-shelf classifier in the world.”
- Performs well with many base classifiers and in a variety of problem domains.
- AdaBoost is generally slow to overfit.
- Boosted naïve Bayes tied for first place in the 1997 KDD Cup. (Elkan [1997])
- Boosted naïve Bayes is a scalable, interpretable classifier (Ridgeway, *et al* [1998]).

Boosting Example

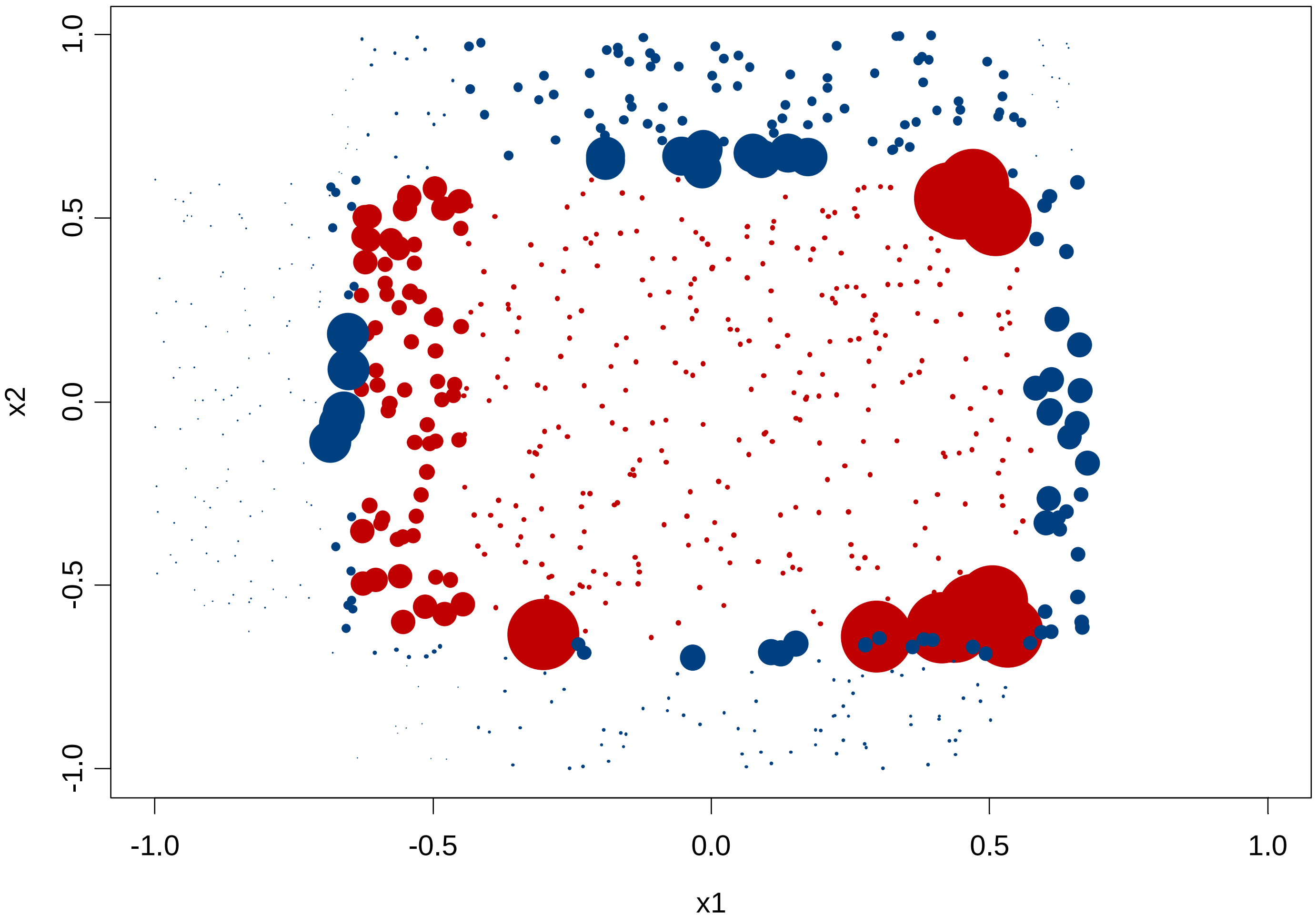


After one iteration

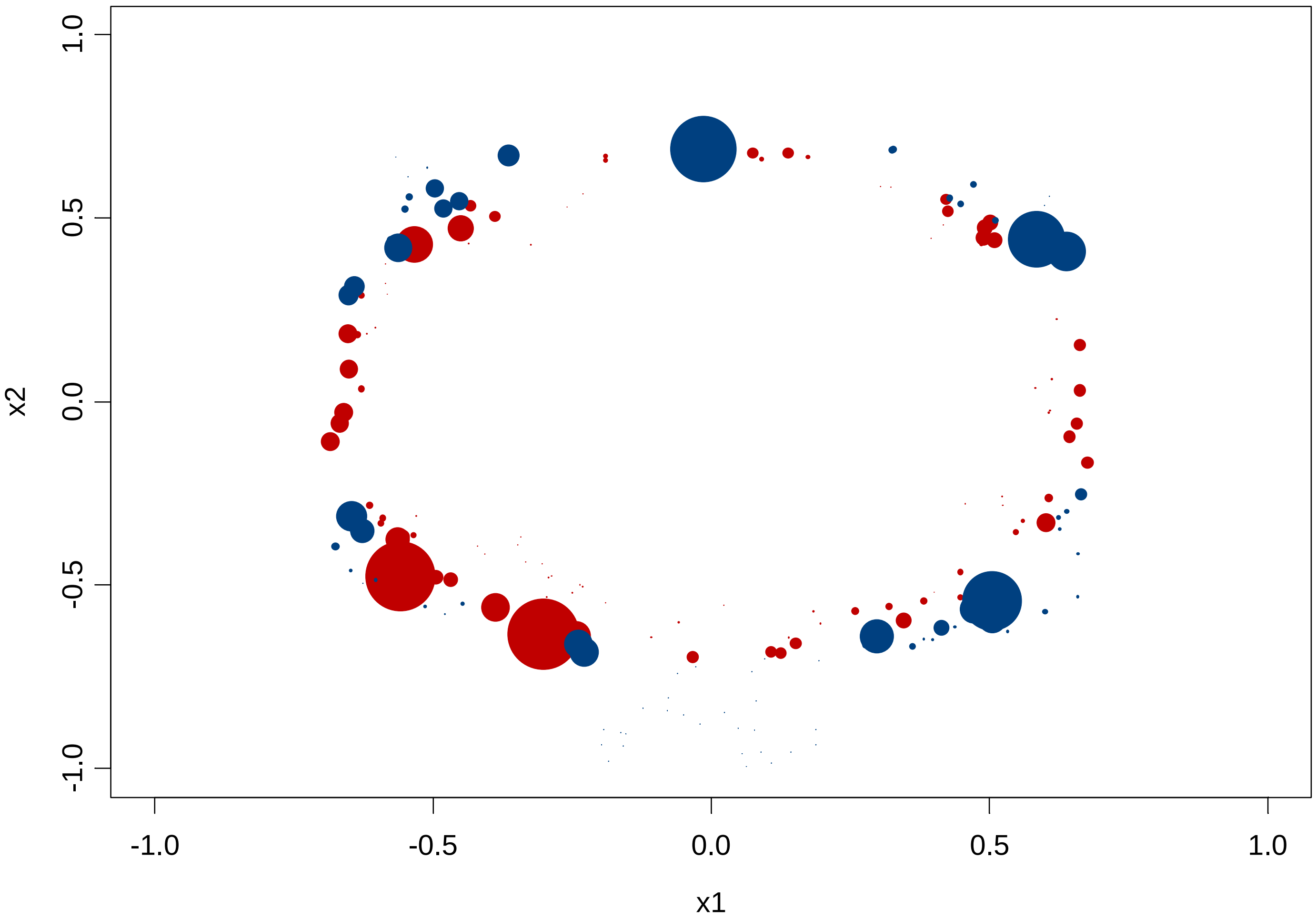
CART splits, larger points have great weight



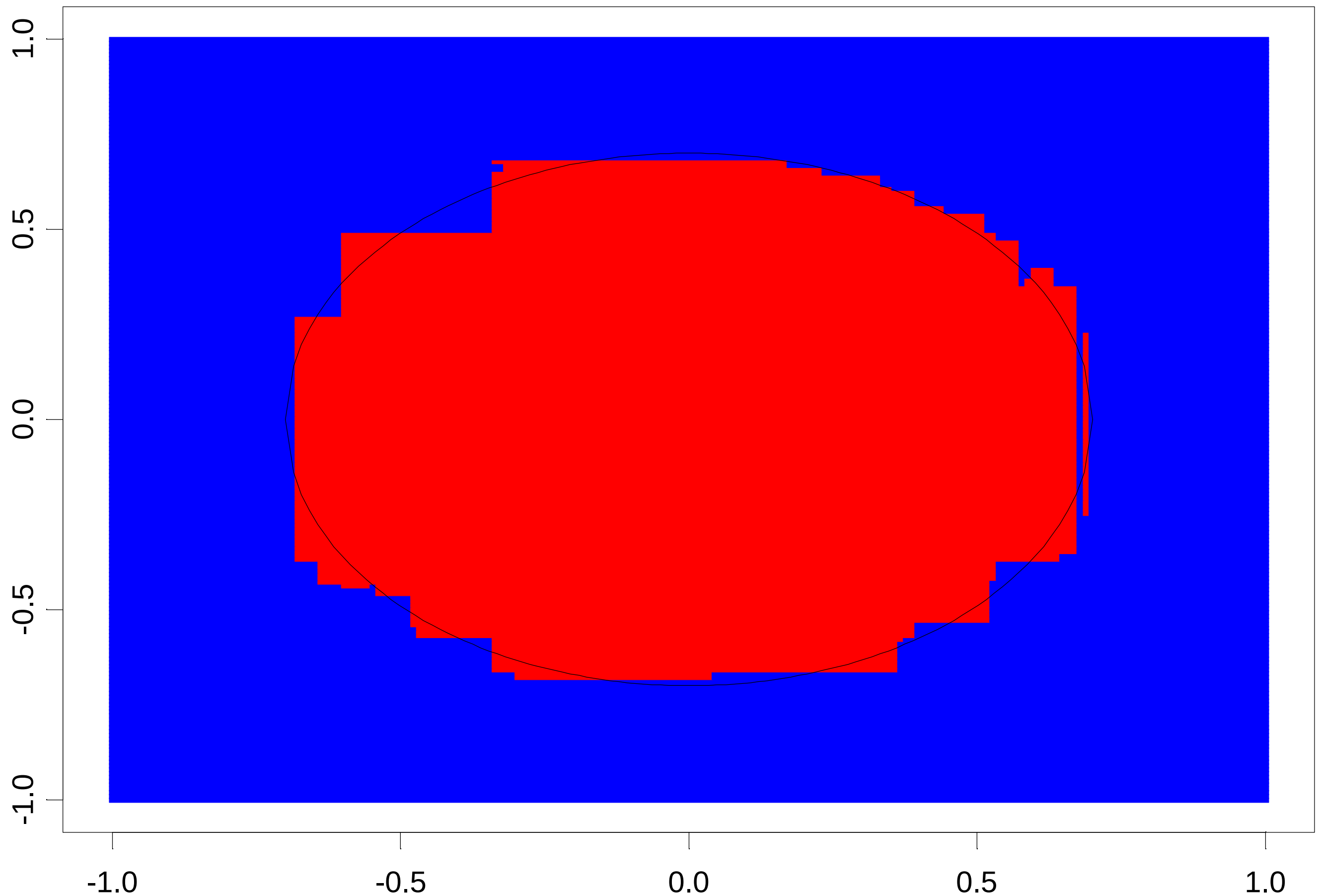
After 3 iterations



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.

$$E\big(\ell(p(x), y)\big) = -E\log(1 + e^{-2yF(x)})$$

Optimization step

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

- Select *f* to minimize *J*…

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

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

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

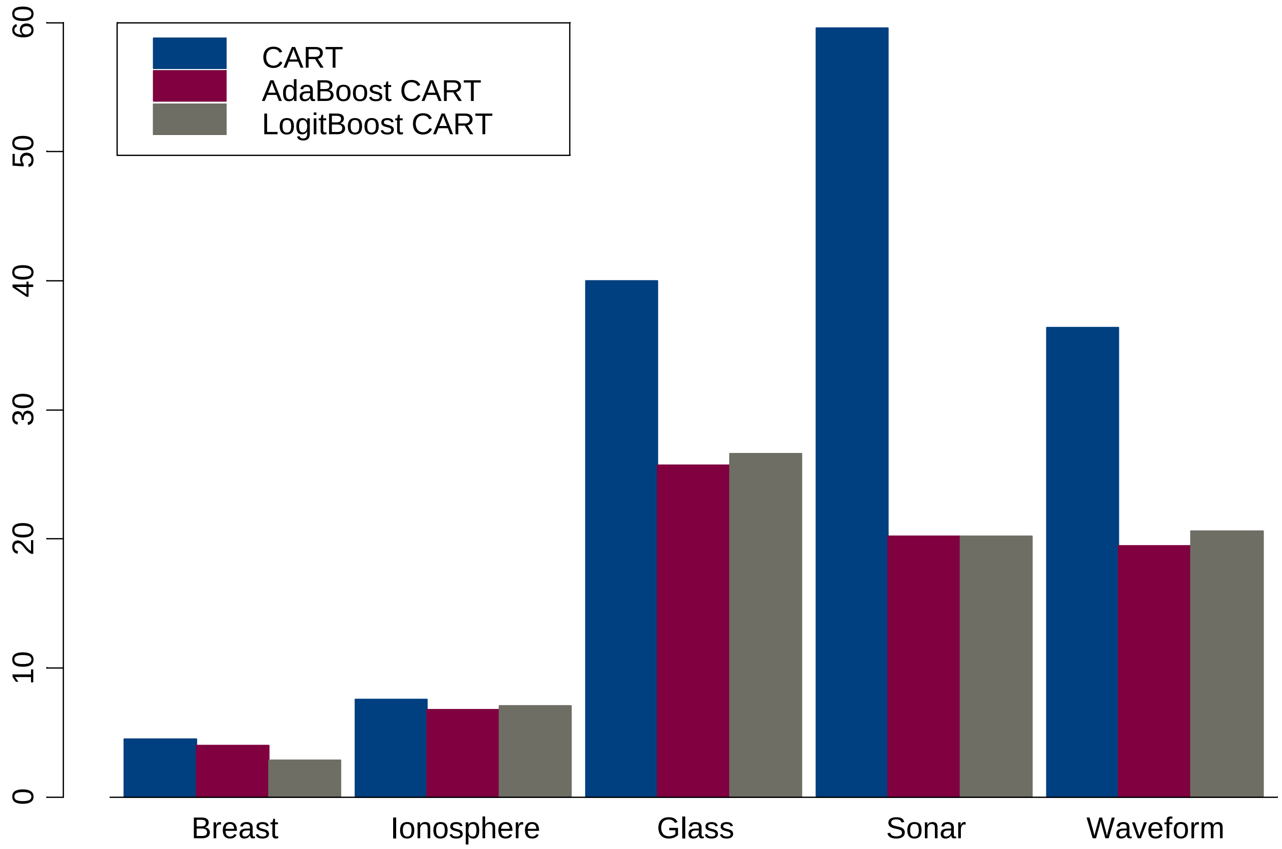
- Newton steps for Bernoulli likelihood

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

- 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]



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.

In general, combining ("bundling") estimators consists of two steps:

1) Constructing varied models, and
2) Combining their estimates

Generate component models by varying:

- Case Weights
- Data Values
- Guiding Parameters
- Variable Subsets

Combine estimates using:

- Estimator Weights
- Voting
- Advisor Perceptrons
- Partitions of Design Space, *X*

Other Bundling Techniques

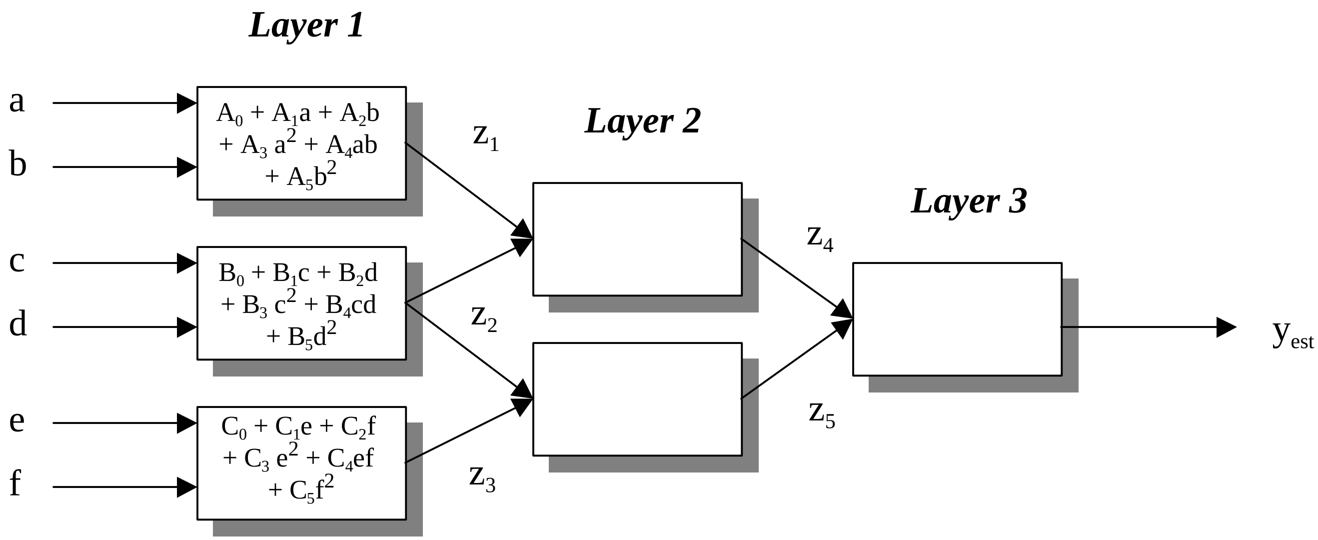
We've Examined:

- ***Bayesian Model Averaging***: sum estimates of possible models, weighted by posterior evidence
- ***Bagging*** (Breiman 96) (*b*ootstrap *agg*regating) -- bootstrap data (to build trees mostly); take majority vote or average
- ***Boosting*** (Freund & Shapire 96) -- weight error cases by $b_t = (1\text{-}e(t))/e(t)$, iteratively re-model; average, weighing model *t* by $\ln(b_t)$

Additional Example Techniques:

- ***GMDH*** (Ivakhenko 68) -- multiple layers of quadratic polynomials, using two inputs each, fit by Linear Regression
- ***Stacking*** (Wolpert 92) -- train a 2nd-level (LR) model using leave-1-out estimates of 1st-level (neural net) models
- ***ARCing*** (Breiman 96) (Adaptive Resampling and Combining) -- Bagging with reweighting of error cases; superset of boosting
- ***Bumping*** (Tibshirani 97) -- bootstrap, select single best
- ***Crumpling*** (Anderson & Elder 98) -- average cross-validations
- ***Born-Again*** (Breiman 98) -- invent new X data...

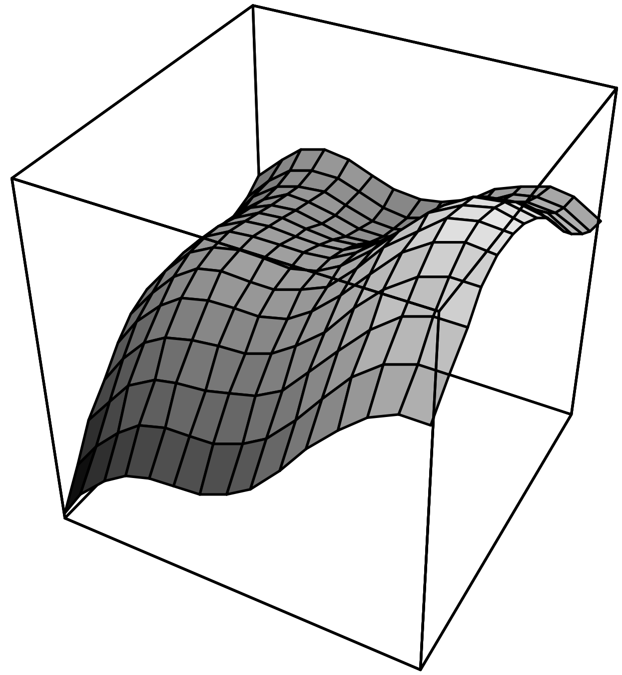
Group Method of Data Handling (GMDH)

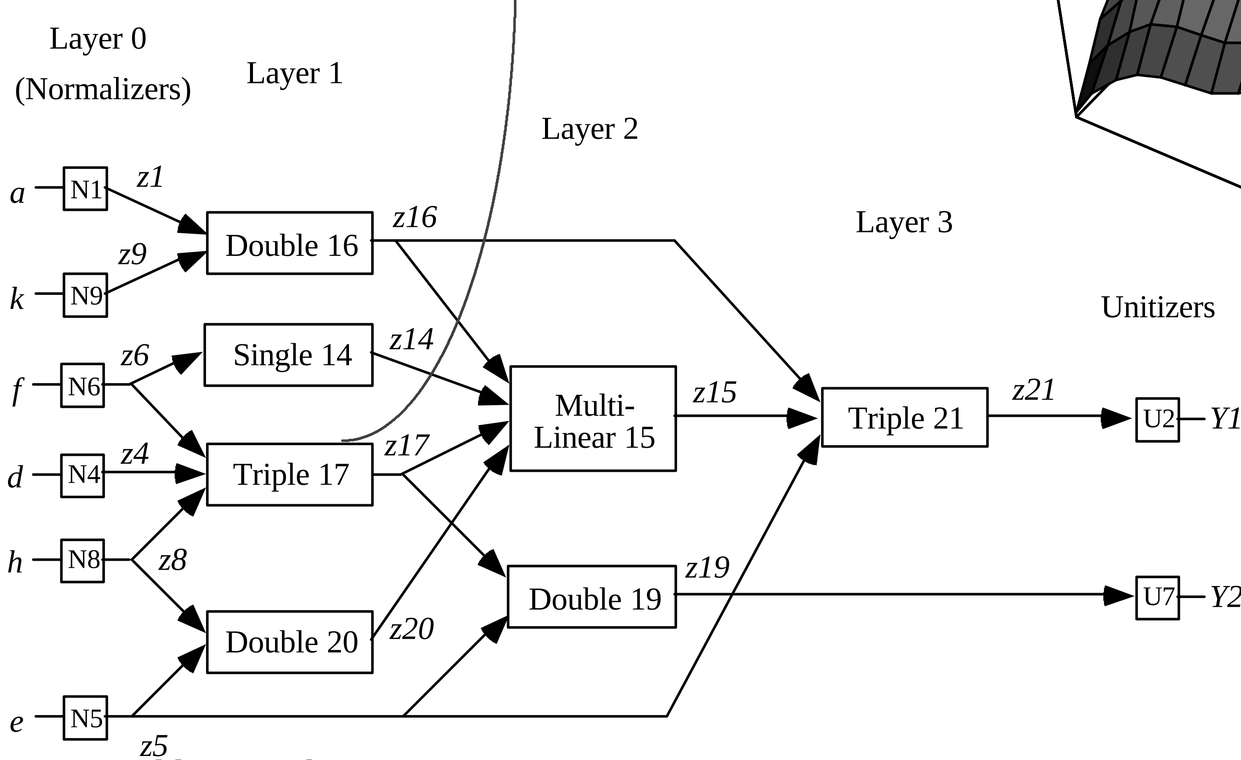


- Try all pairs of variables (*K* choose 2) in quadratic polynomial nodes.
- Fit coefficients using regression.
- Keep best *M* nodes.
- Train model on one training data set, score on test data set. (Need a third data set for independent confirmation of model.)

Polynomial Networks (ASPN)

$$Z_{17} = 3.1 + 0.4a - .15b^2 + 0.9bc - 0.62abc + 0.5c^3$$





When does Bundling work?

Hypotheses:

- Breiman (1996): when the prediction method is *unstable* (significantly different models are constructed)
- Ali & Pazzani (1996): when there is low noise, lots of irrelevant variables, and good individual predictors which make different errors
- when models are slightly overfit
- when models are from different families

Advanced techniques

- Stochastic gradient boosting
- Adaptive bagging
- Example regression and classification results

Stochastic Gradient Boosting

Goal: Non-parametric function estimation

Method: Cast the problem as optimization and use gradient ascent to obtain predictor

Properties:

- Bias and variance reduction
- Widely applicable
- Can make use of existing algorithms
- Many tuning parameters

Improving boosting

- Boosting usually has the form

$$F^{(t+1)}(x) \leftarrow F^{(t)}(x) + \lambda E_w\left(z(y, x) \middle| x\right)$$

Improve by...

- Sub-sampling a fraction of the data at each step when computing the expectation.
- "Robustifying" the expectation.
- Trimming observations with small weights.

Stochastic gradient boosting offers...

- Application to likelihood based models (GLM, Cox models)
- Bias reduction - non-linear fitting
- Massive datasets - bagging, trimming
- Variance reduction - bagging
- Interpretability - additive models
- High-dimensional regression - trees
- Robust regression

SGB References

- Friedman, J. (1999). "Greedy function approximation: a gradient boosting machine," Technical report, Dept. of Statistics, Stanford University.
- Friedman, J. (1999). "Stochastic gradient boosting," Technical report, Dept. of Statistics, Stanford University.

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

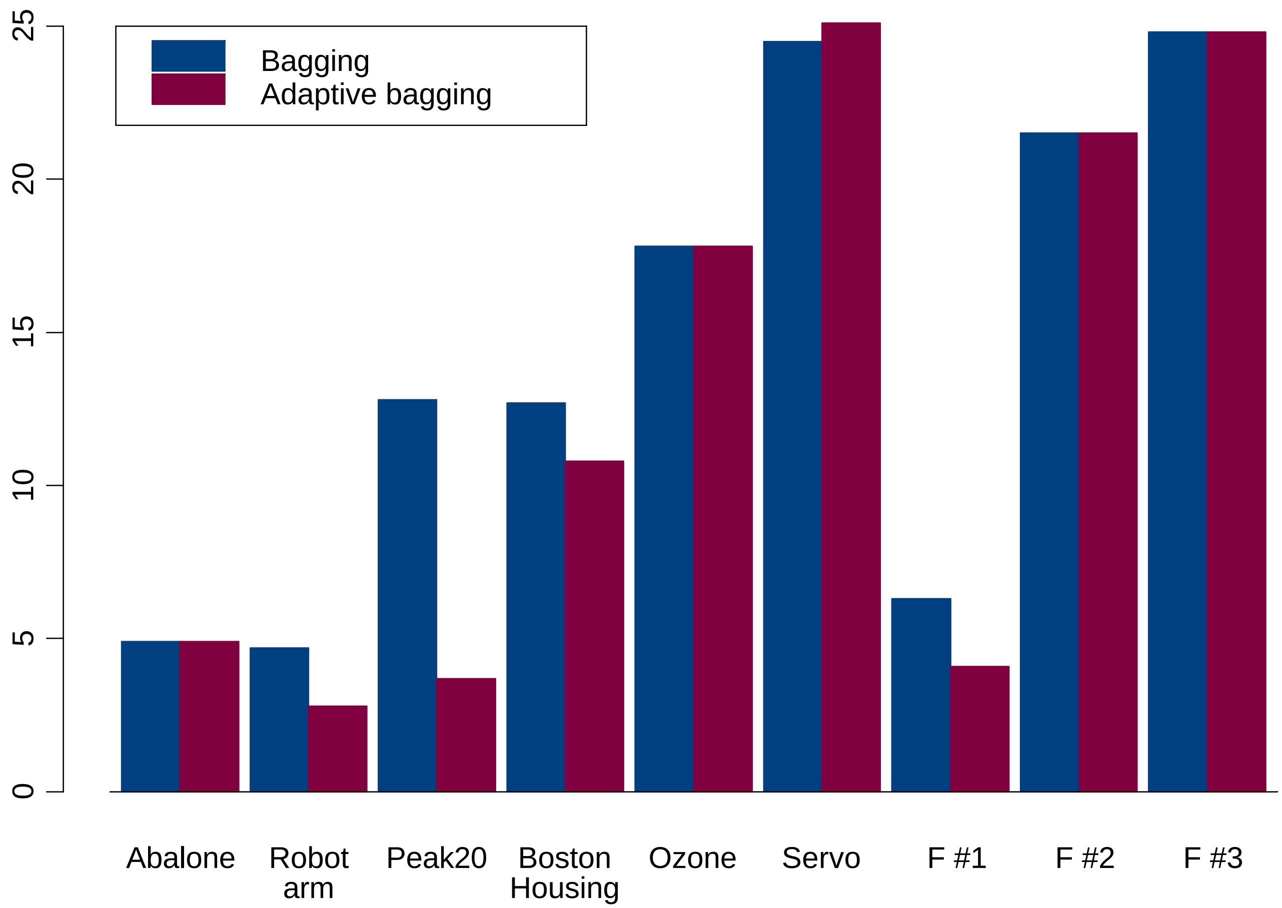
Adaptive bagging algorithm

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.

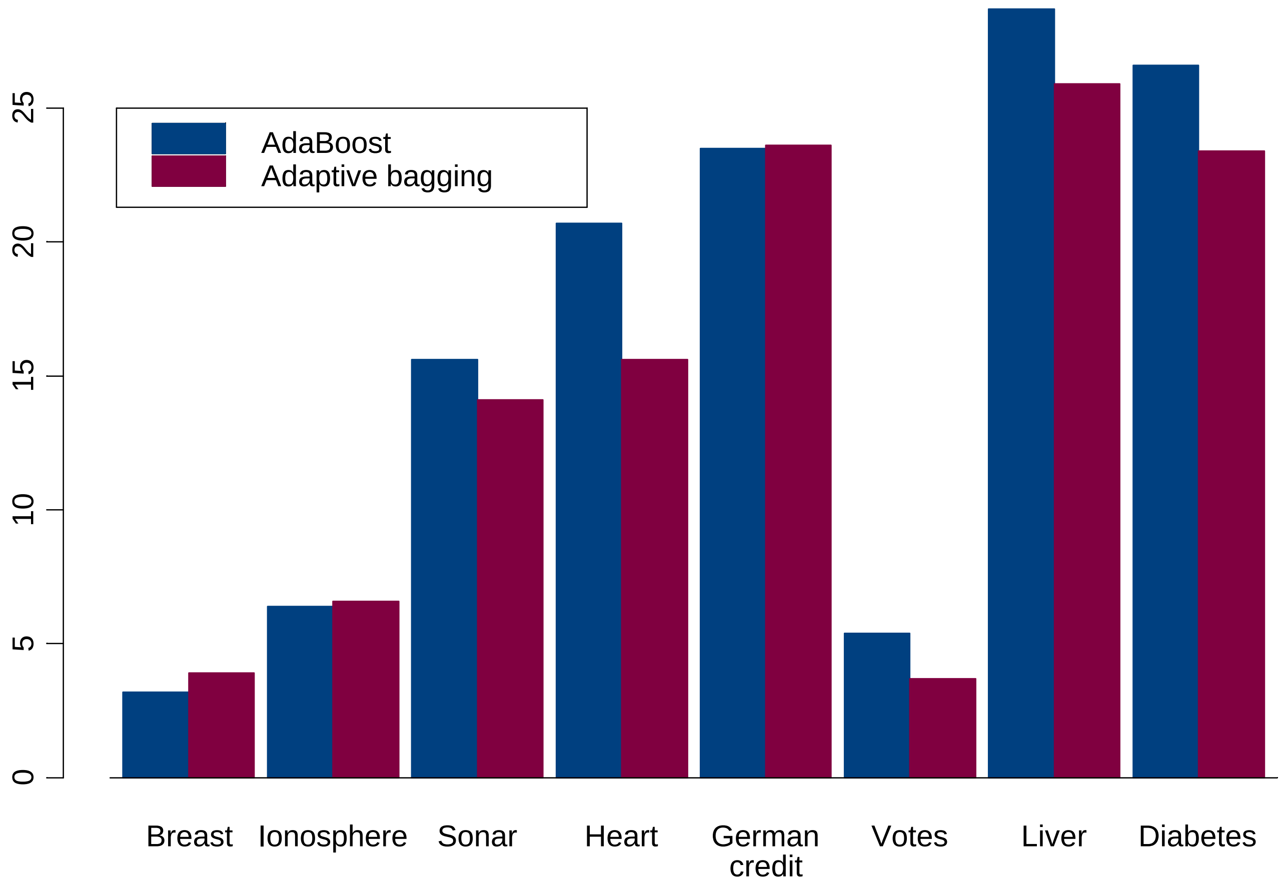
Regression results

Squared error loss



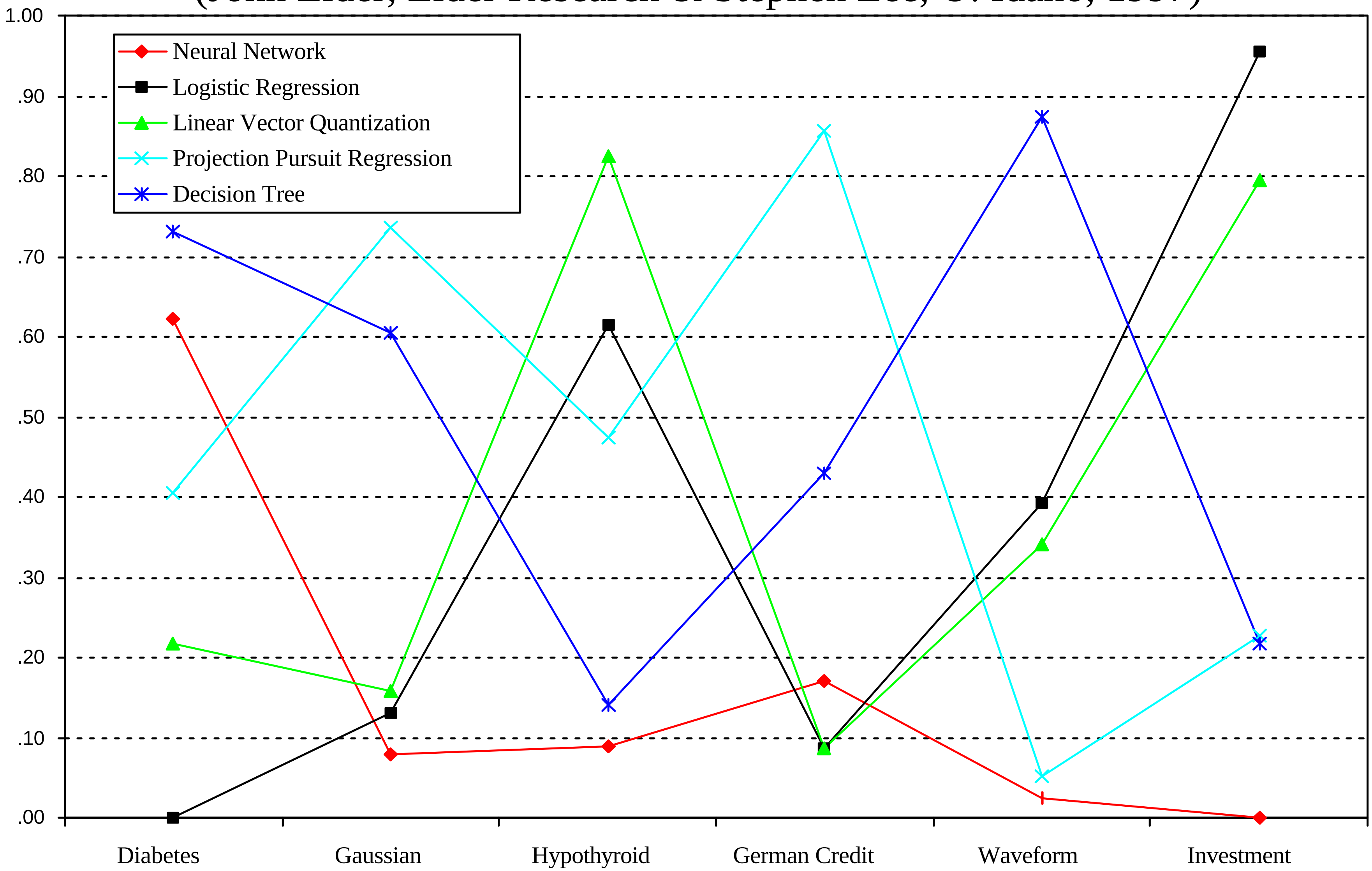
Classification results

Misclassification rates

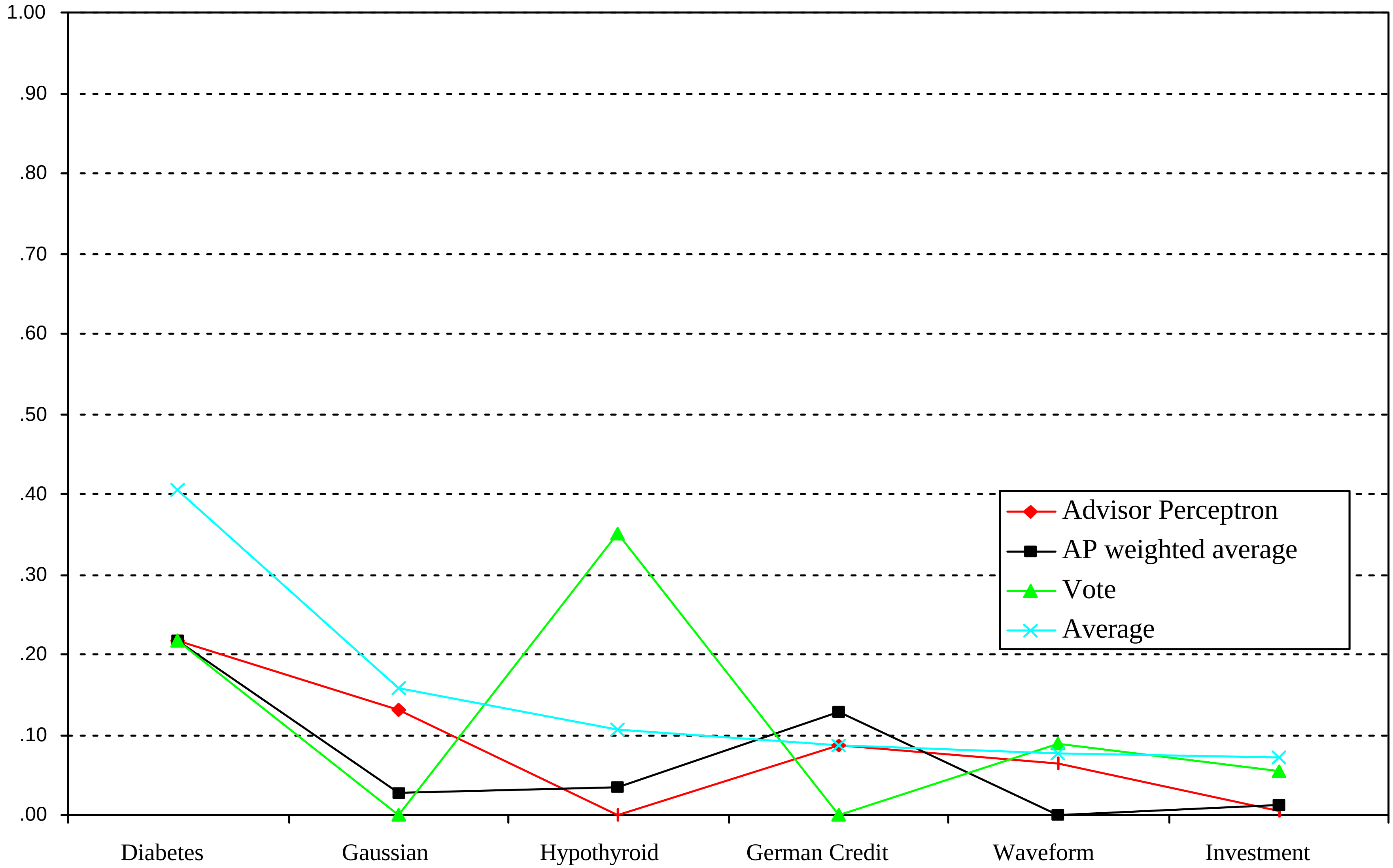


Relative Performance Examples: 5 Algorithms on 6 Datasets

(John Elder, Elder Research & Stephen Lee, U. Idaho, 1997)



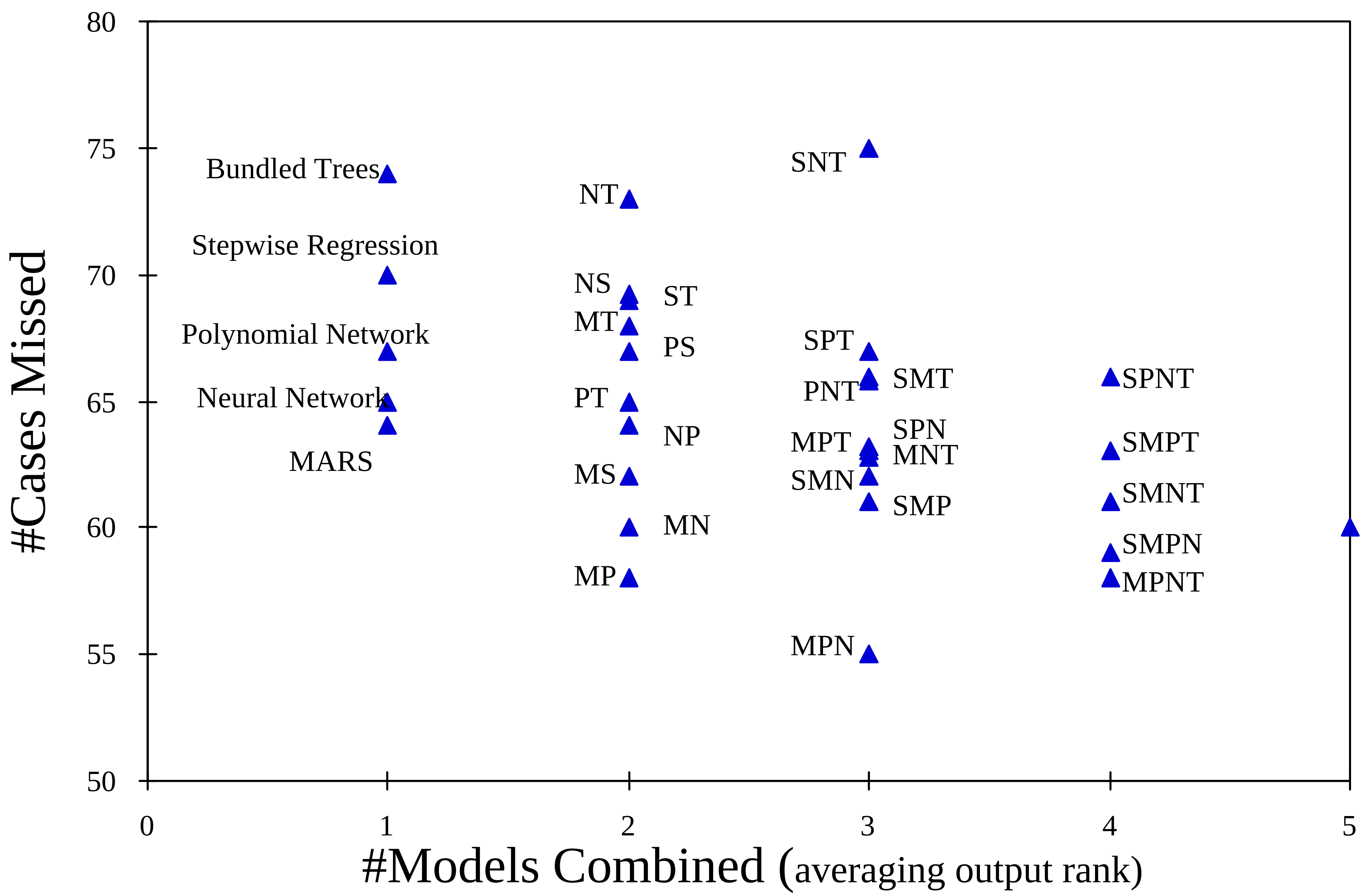
Essentially every Bundling method improves performance



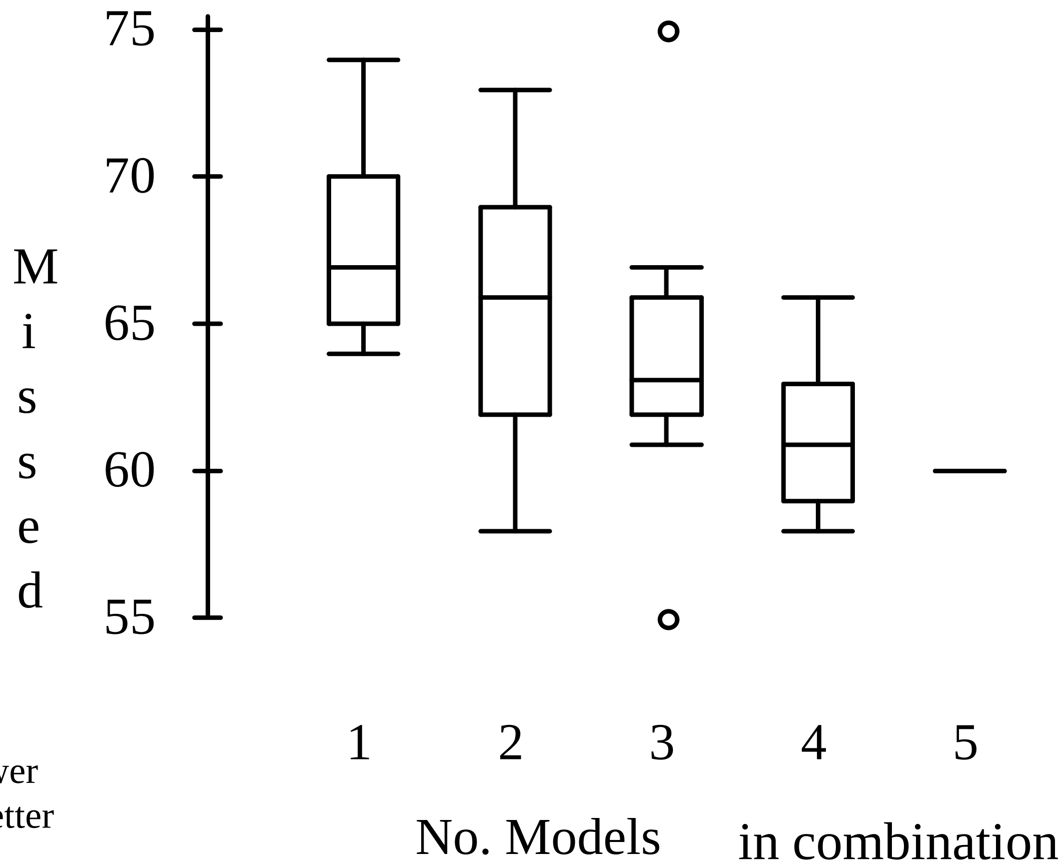
Application Ex.: Direct Marketing
(Elder Research 1996-1998)

- Model respondants to direct marketing as binary variable: 0 (no response), 1 (response).
- Create models using several (here, 5) different algorithms, all employing the same candidate model inputs.
- Rank-order model responses:
  - Give highest-probability response value a rank of 1, second highest value 2, etc.
  - For bundling, combine model ranks (not estimates) into a new consensus estimate (which is again ranked).
- Report number of response cases missed (in top portion).

Marketing Application Performance



Median (and Mean) Error Reduced with each Stage of Combination



NOTE: Fewer misses is better

...and in a multitude of counselors there is safety.
Proverbs 24:6b

Why Bundling works

- (semi-) Independent Estimators
- Bayes Rule - weighing evidence
- Shrinking (ex.: stepwise LR)
- Smoothing (ex.: decision trees)
- Additive modeling and maximum likelihood (Friedman, Hastie, & Tibshirani 8/20/98)

… Open research area.
Meanwhile, we recommend bundling competing candidate models both within, and between, model families.