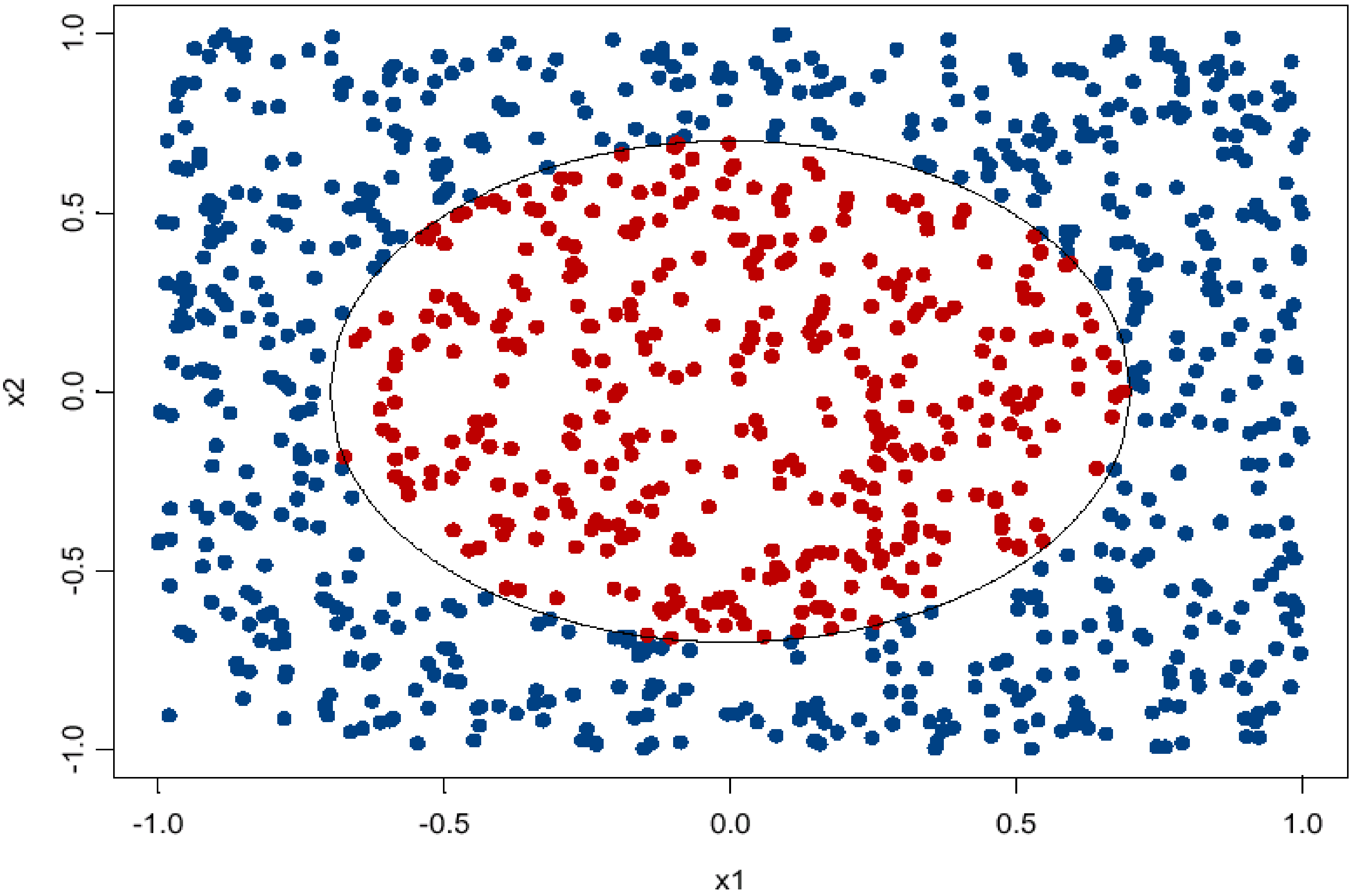
CSE 446
Ensembles

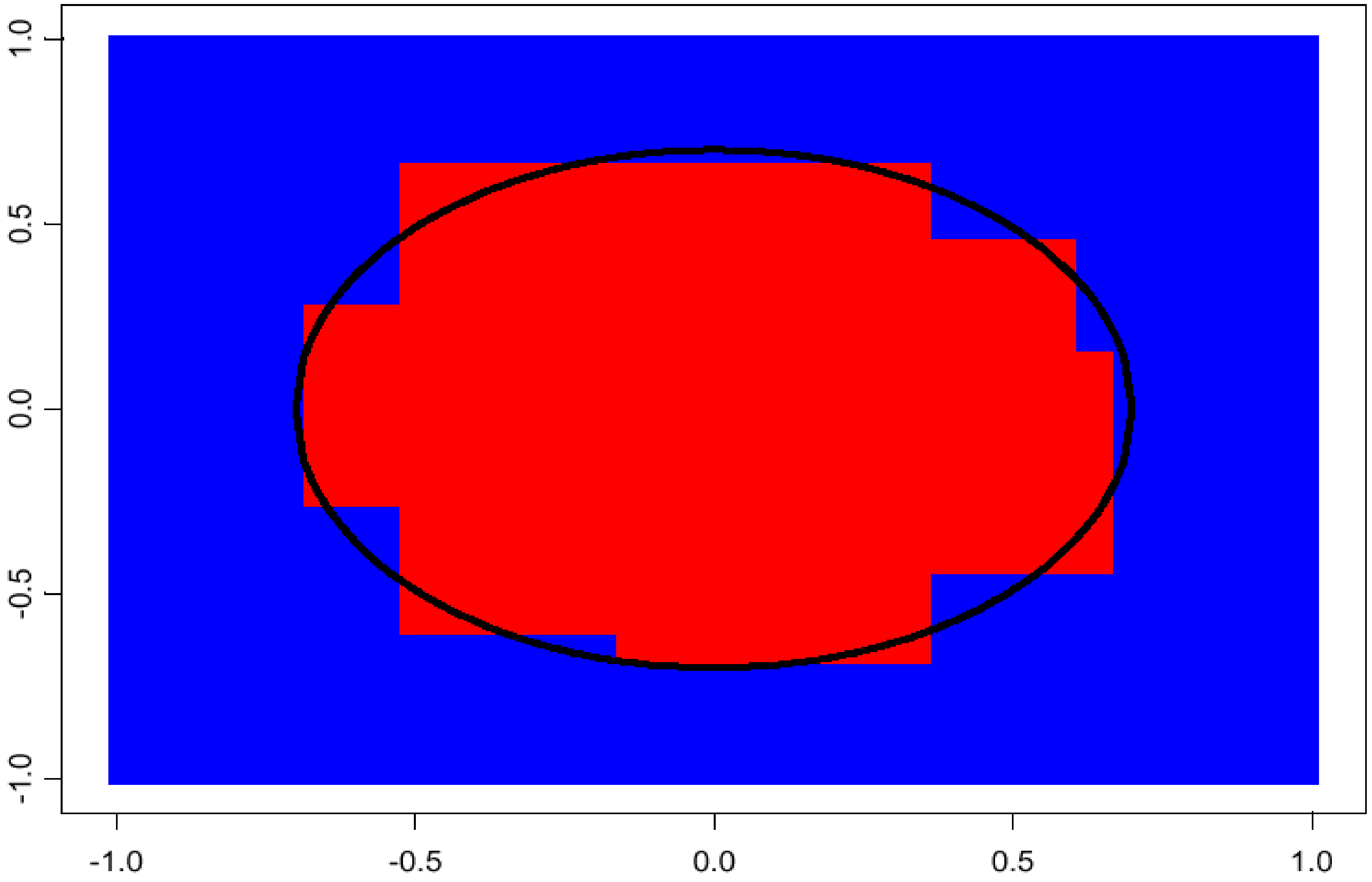
Administrative

- Grading
  - Homework 1 grades are out
  - Midterm grading in progress
- Homework 2 due today
- Homework 3 already out, start early!
- Today: model ensambles

Dataset



Decision Tree Fit



Voting (Ensemble Methods)

- Instead of learning a single classifier, learn **many weak classifiers** that are **good at different parts of the data**
- **Output class:** (Weighted) vote of each classifier
  - Classifiers that are most "sure" will vote with more conviction
  - Classifiers will be most "sure" about a particular part of the space

Simple version: BAGGing = Bootstrap AGGregation

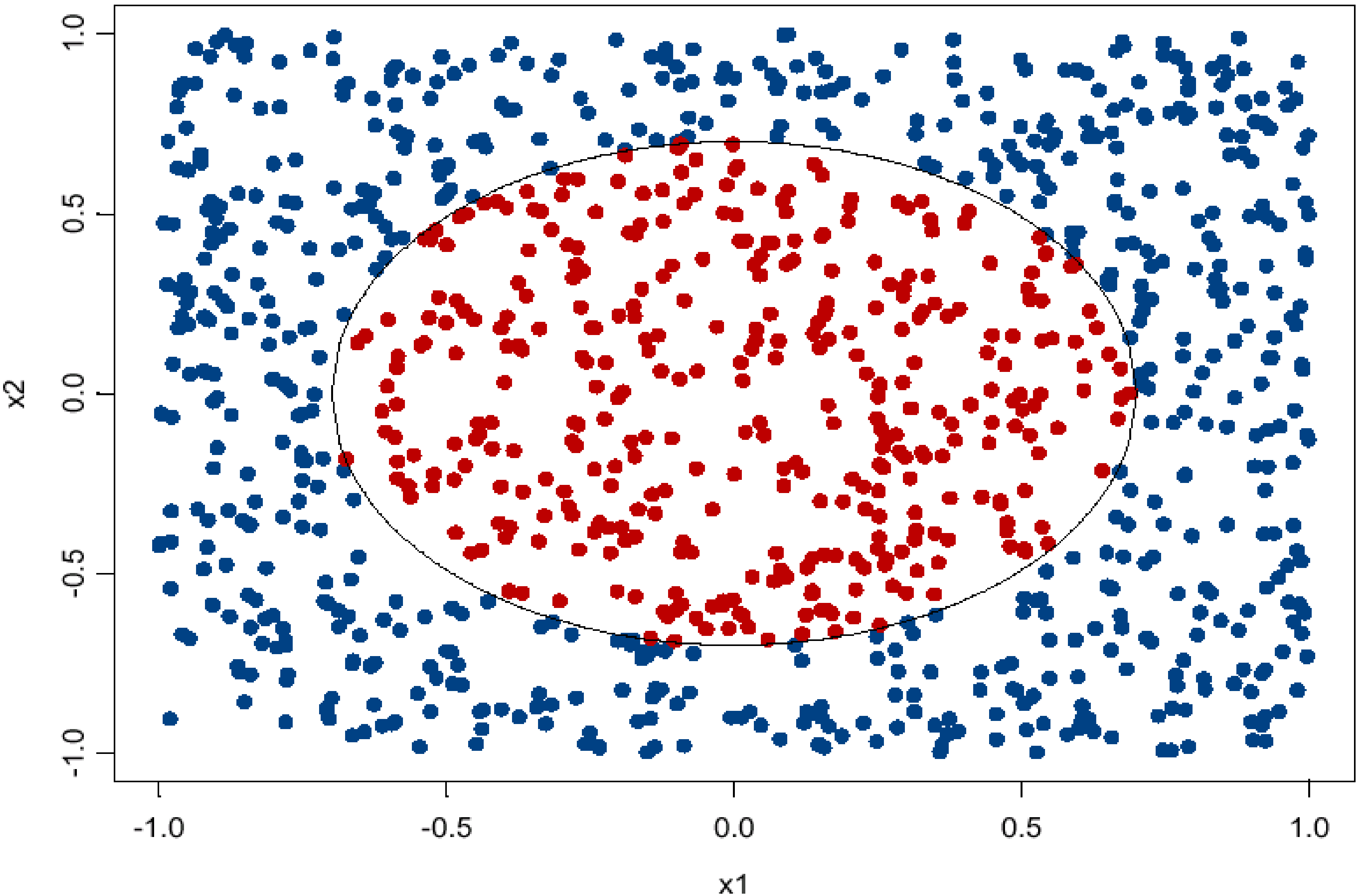
(Breiman, 1996)

- for t = 1, 2, …, T:
  - $D_t \leftarrow$ randomly select M training instances with replacement
  - $h_t \leftarrow$ learn(D_t) *[Decision Tree, Naive Bayes, …]*
- Now combine the h_t together with uniform voting (w_t=1/T for all t)

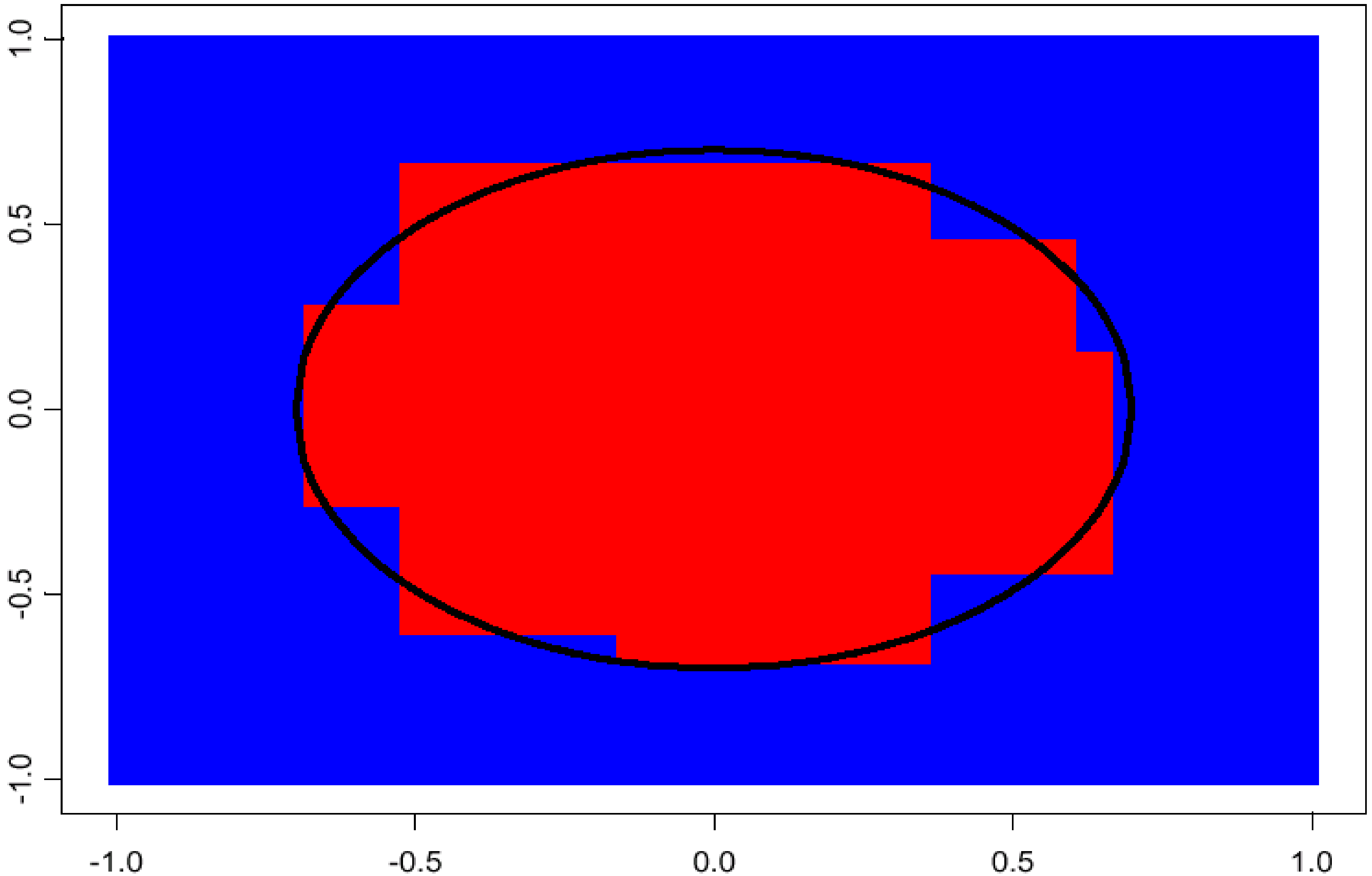
D = { [0, 0, 1 | 0], [1, 1, 0 | 0], [0, 1, 0 | 1], [1, 1, 1 | 0] }

D_t = { [1, 1, 0 | 0], [1, 1, 0 | 0], [1, 1, 1 | 0], [0, 0, 1 | 0] }

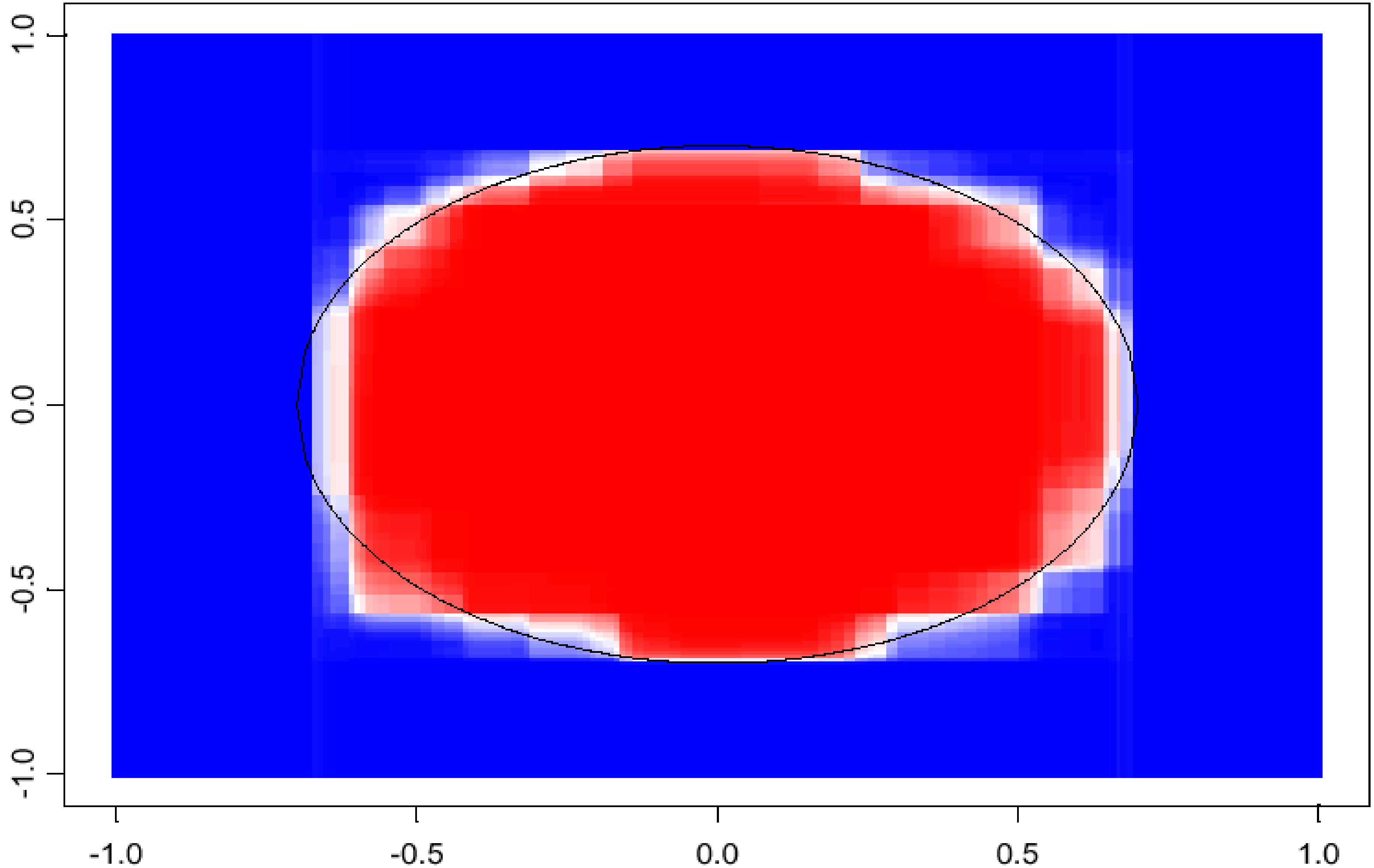
Dataset



Decision Tree Fit

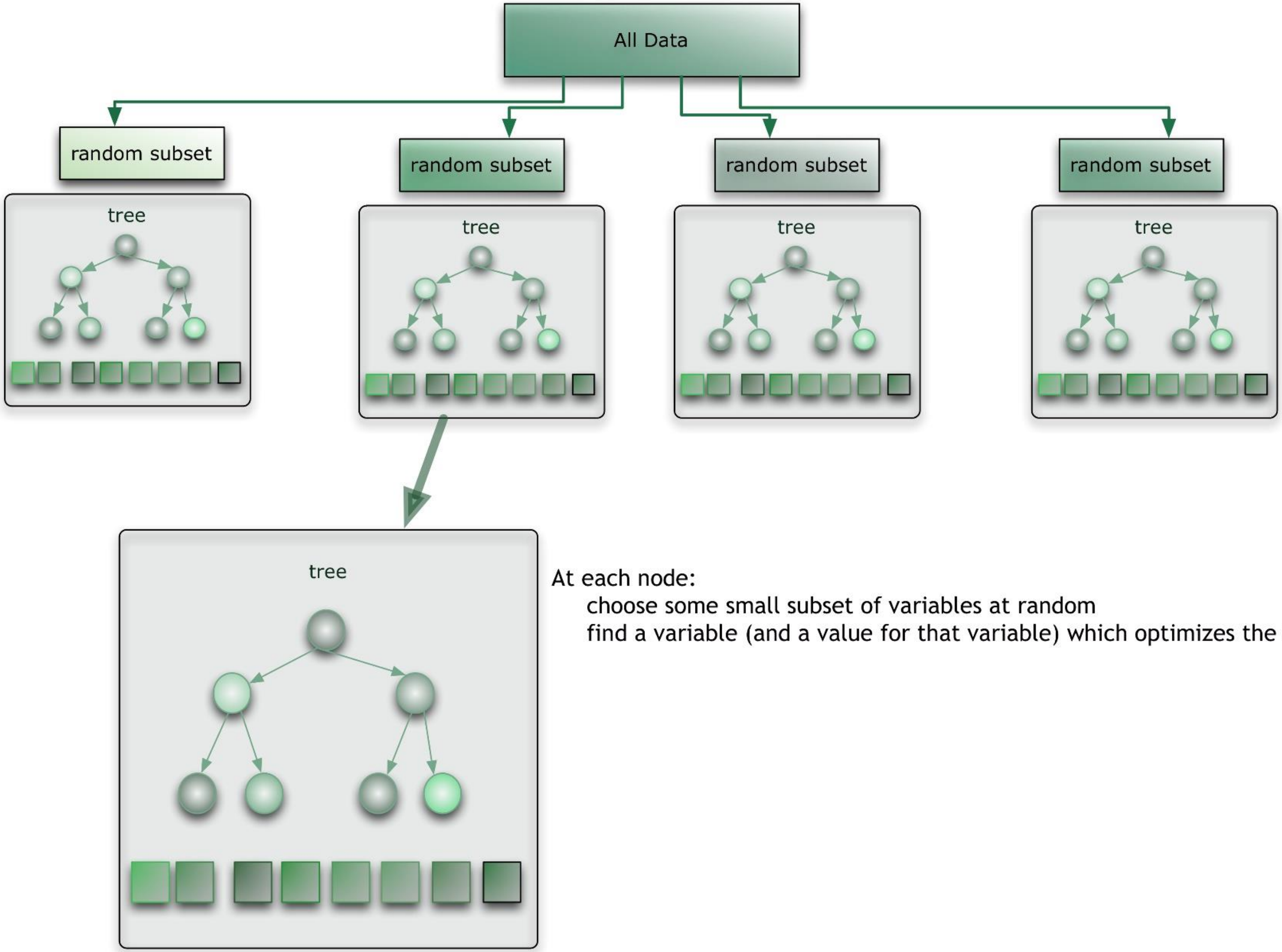


100 bagged trees



shades of blue/red indicate strength of vote for particular classification

Random Forests



At each node:
- choose some small subset of variables at random
- find a variable (and a value for that variable) which optimizes the split

Fighting the bias-variance tradeoff

- **Simple ("weak") learners**
  - e.g., naïve Bayes, logistic regression, decision stumps (or shallow decision trees)
  - Low variance, don't usually overfit
- **Why not use weak learners all the time?**
  - High bias, can't solve hard learning problems
- **Ensembles use independent weak learners (which don't overfit as much), and put many of them together to reduce bias**

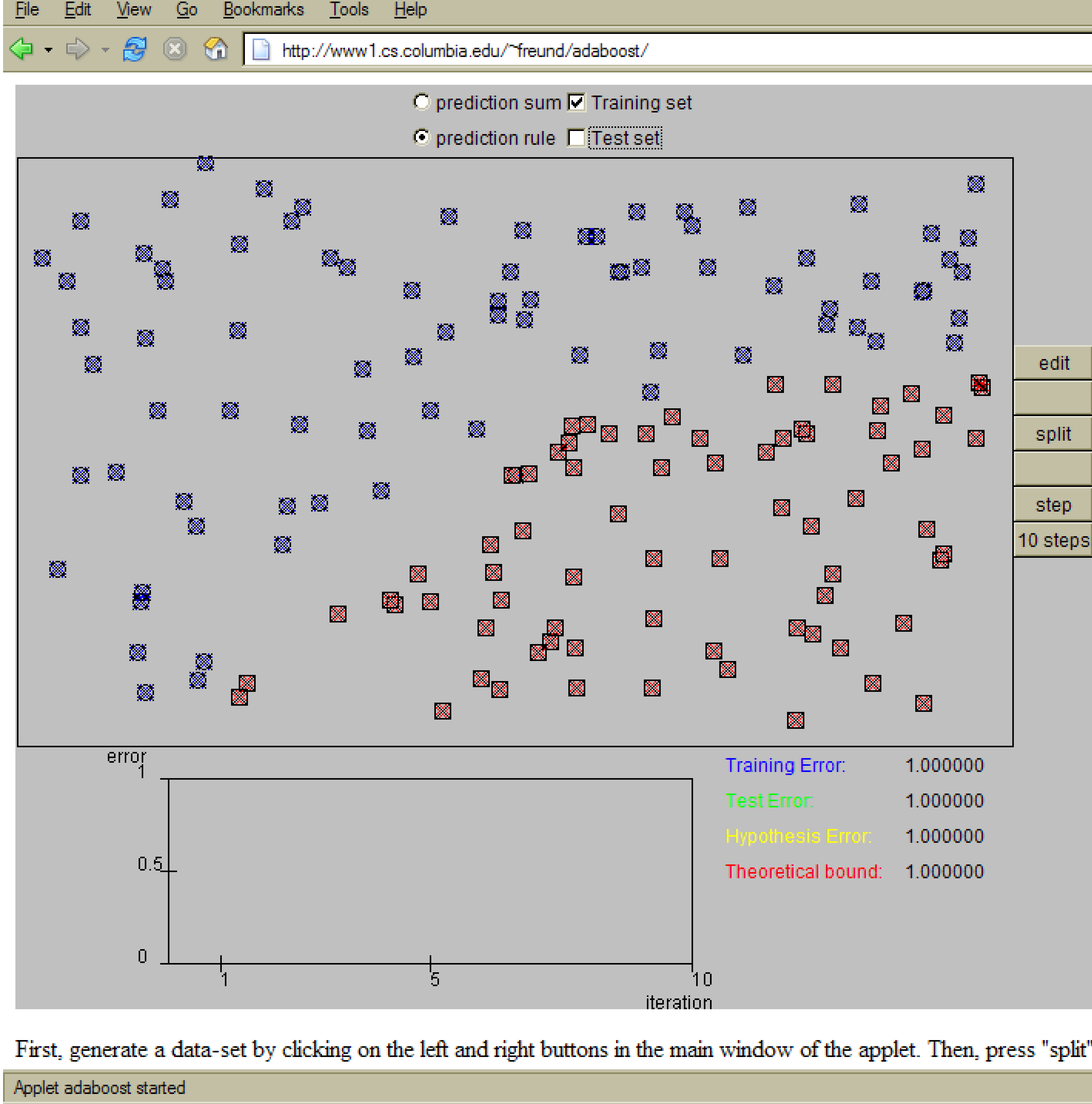
Boosting

[Schapire, 1989]

- Idea: if we give each weak learner a difference piece of the dataset, we get a really good complex classifier from letting them vote
- Learners must be different (how was this achieved in Bagging?)
- Learners must be better than random (not too weak)
- Approach: given a weak learner, run it multiple times on (reweighted) training data, then let learned classifiers vote
- On each iteration *t*:
  - weight each training example by how incorrectly it was classified
  - Learn a hypothesis – h_t
  - A strength for this hypothesis – α_t
- Final classifier:

$$\boxed{h(x) = \text{sign}\left(\sum_i \alpha_i h_i(x)\right)}$$

- **Practically useful**
- **Theoretically interesting**

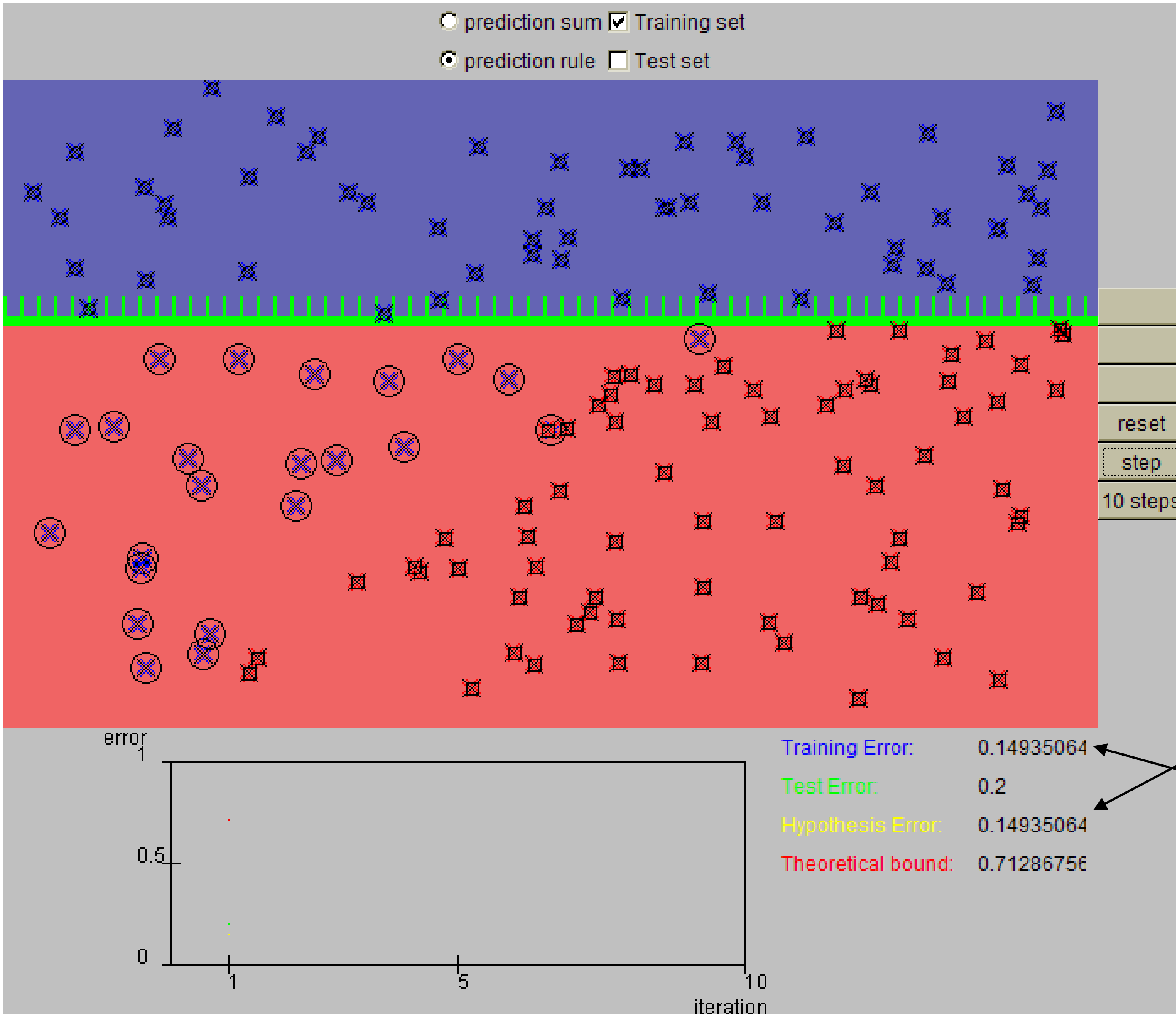


time = 0

blue/red = class

size of dot = weight

weak learner = Decision stub: horizontal or vertical http://www1.cs.columbia.edu/~freund/adaboost/



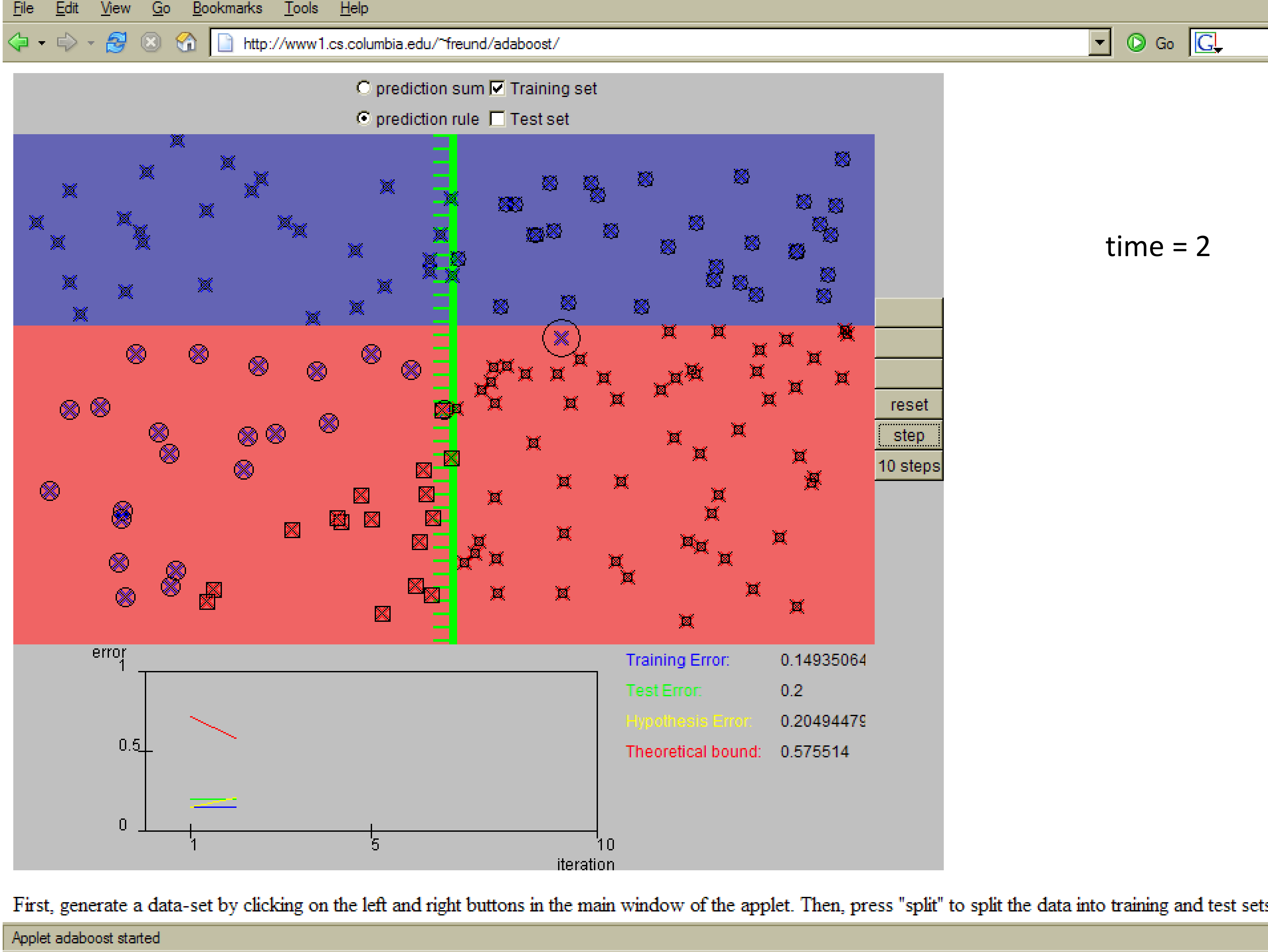
time = 1

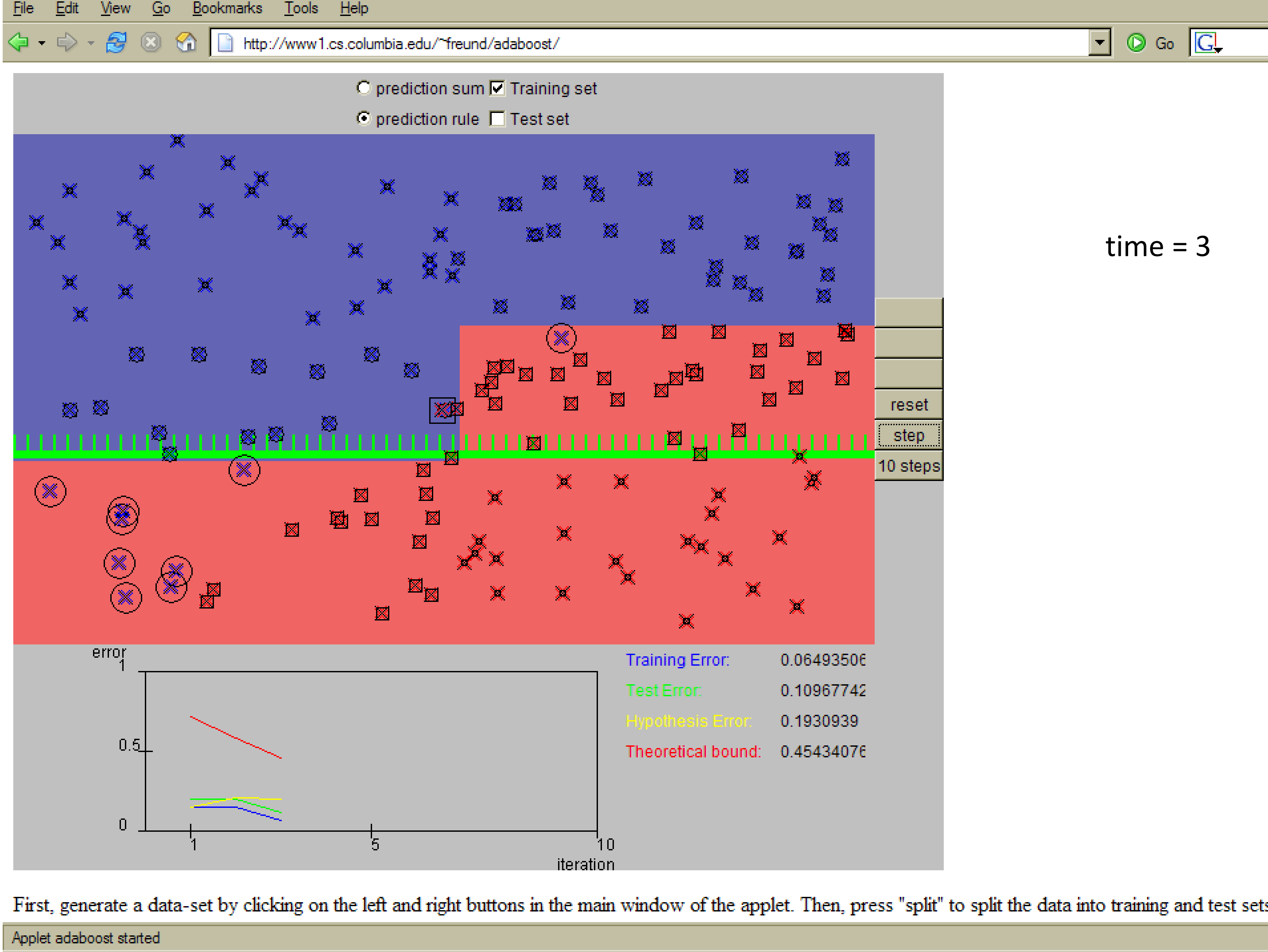
this hypothesis has 15% error

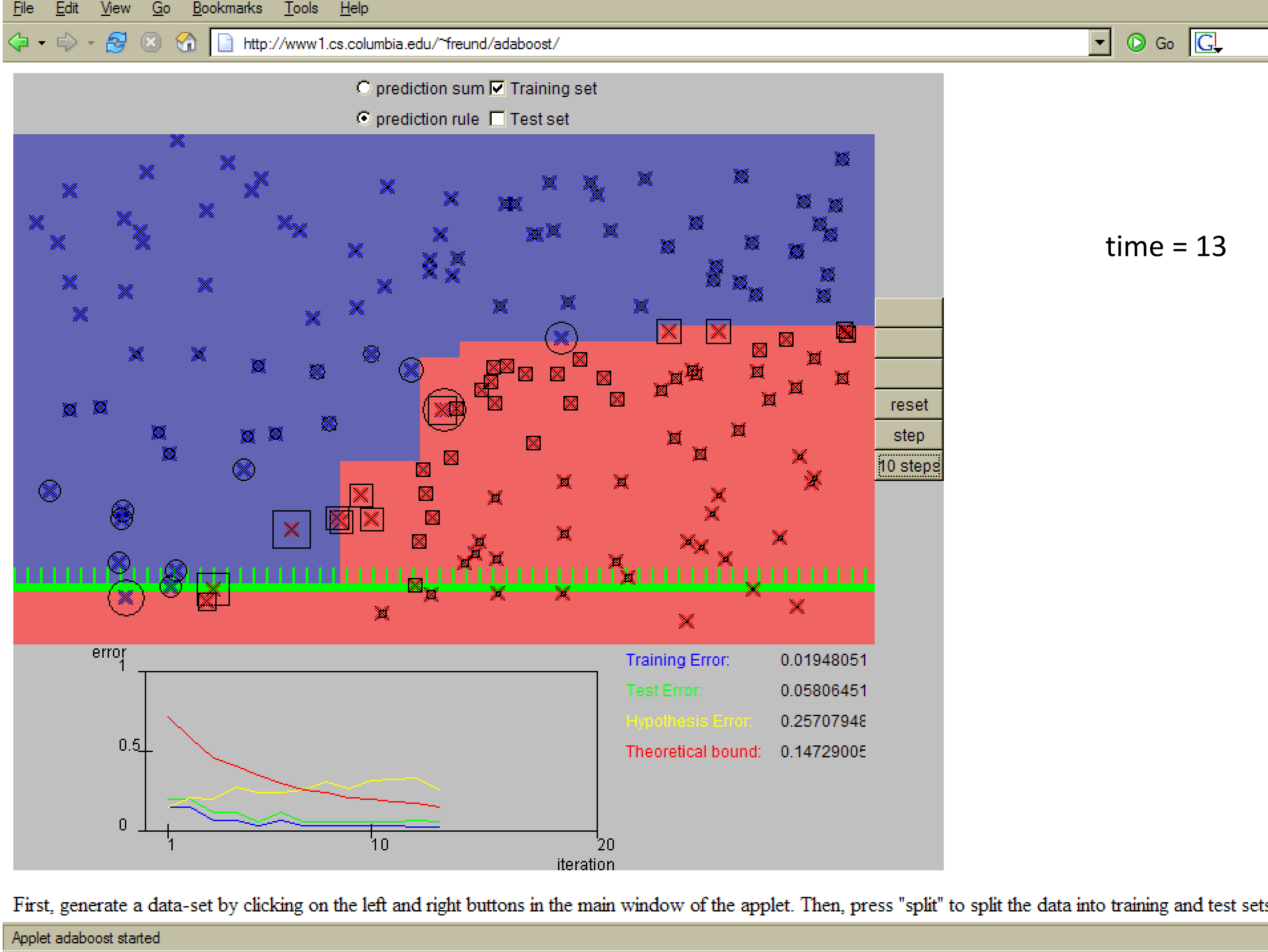
and so does this ensemble, since the ensemble contains just this one hypothesis

First, generate a data-set by clicking on the left and right buttons in the main window of the applet. Then, press "split" to split the data into training and test sets

Applet adaboost started


File Edit View Go Bookmarks Tools Help
http://www1.cs.columbia.edu/~freund/adaboost/
Go
prediction sum
Training set
prediction rule
Test set
reset
step
10 steps
error
1
0.5
0
1
5
10
iteration
Training Error: 0.14935064
Test Error: 0.2
Hypothesis Error: 0.20494479
Theoretical bound: 0.575514
time = 2
First, generate a data-set by clicking on the left and right buttons in the main window of the applet. Then, press "split" to split the data into training and test sets
Applet adaboost started

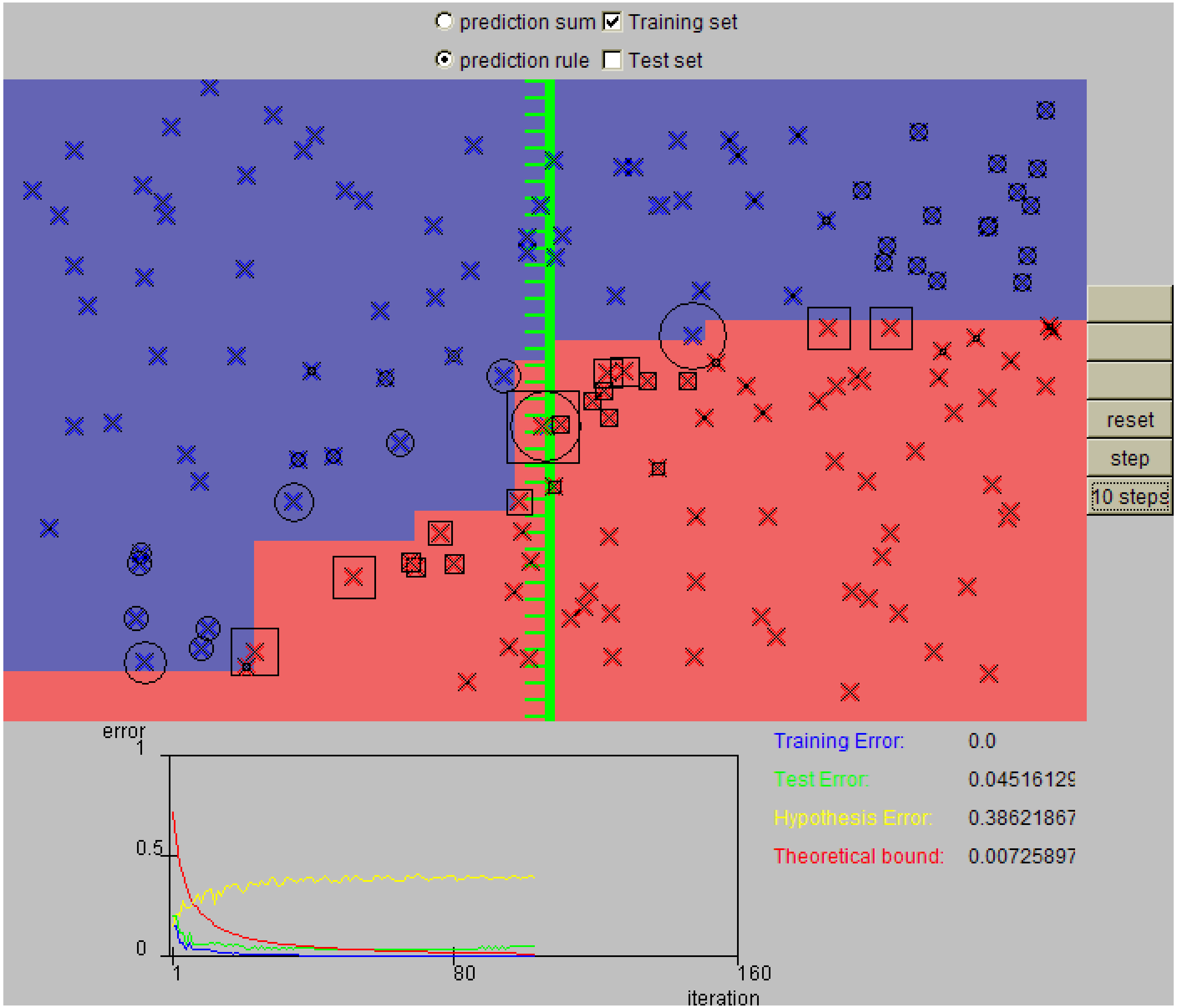

File Edit View Go Bookmarks Tools Help
http://www1.cs.columbia.edu/~freund/adaboost/
Go
prediction sum
Training set
prediction rule
Test set
reset
step
10 steps
time = 3
error
1
0.5
0
1
5
10
iteration
Training Error: 0.0649350
Test Error: 0.10967742
Hypothesis Error: 0.1930939
Theoretical bound: 0.4543407
First, generate a data-set by clicking on the left and right buttons in the main window of the applet. Then, press "split" to split the data into training and test sets
Applet adaboost started


File Edit View Go Bookmarks Tools Help
http://www1.cs.columbia.edu/~freund/adaboost/
Go
prediction sum
Training set
prediction rule
Test set
reset
step
10 steps
error
1
0.5
0
1
10
20
iteration
Training Error: 0.01948051
Test Error: 0.05806451
Hypothesis Error: 0.25707948
Theoretical bound: 0.14729005
First, generate a data-set by clicking on the left and right buttons in the main window of the applet. Then, press "split" to split the data into training and test sets
Applet adaboost started

time = 13

File Edit View Go Bookmarks Tools Help

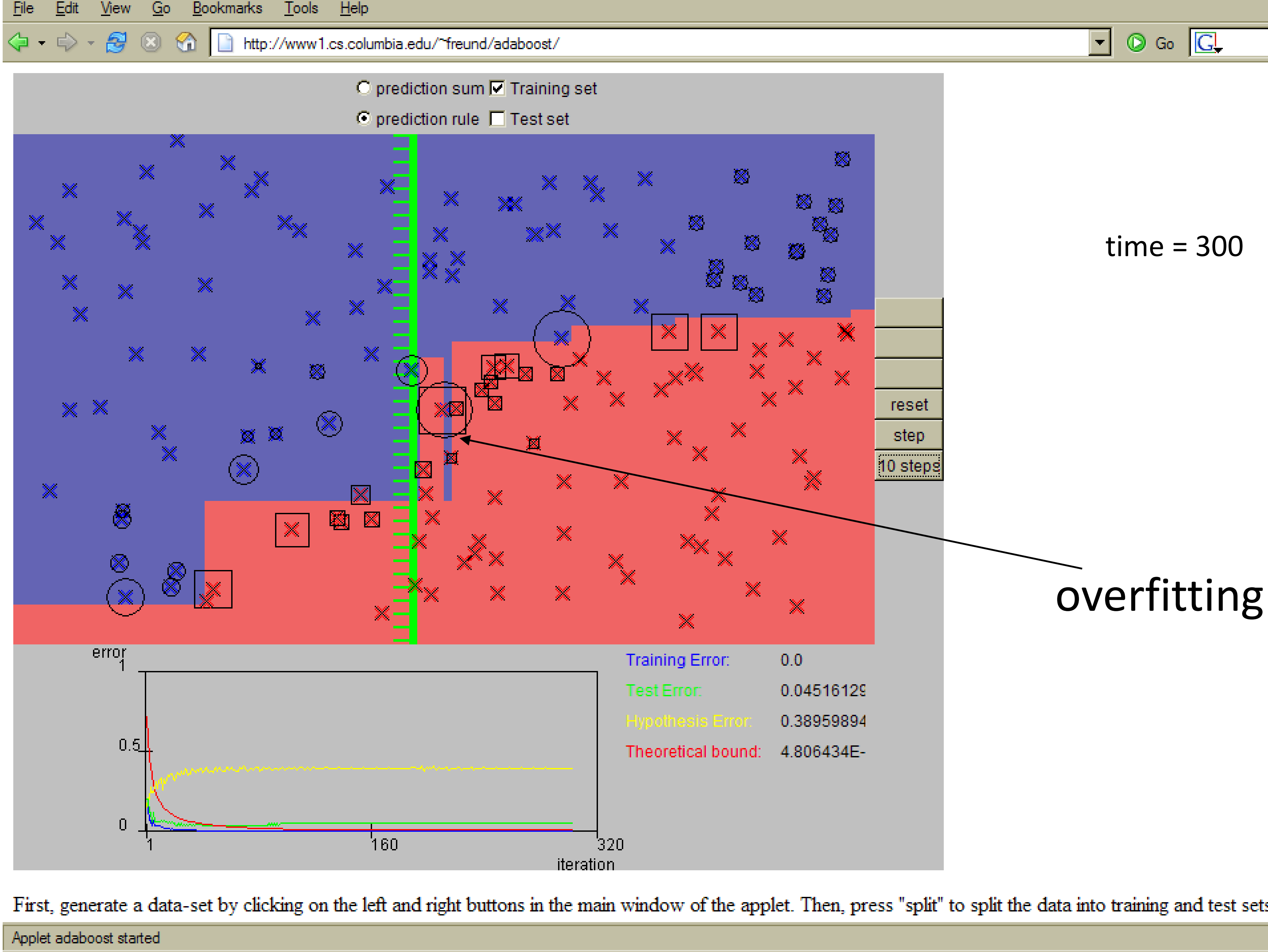
http://www1.cs.columbia.edu/~freund/adaboost/



time = 100

First, generate a data-set by clicking on the left and right buttons in the main window of the applet. Then, press "split" to split the data into training and test sets

Applet adaboost started


File Edit View Go Bookmarks Tools Help
http://www1.cs.columbia.edu/~freund/adaboost/
Go
prediction sum
Training set
prediction rule
Test set
time = 300
reset
step
10 steps
overfitting
error
1
0.5
0
1
160
320
iteration
Training Error: 0.0
Test Error: 0.04516129
Hypothesis Error: 0.38959894
Theoretical bound: 4.806434E-
First, generate a data-set by clicking on the left and right buttons in the main window of the applet. Then, press "split" to split the data into training and test sets
Applet adaboost started