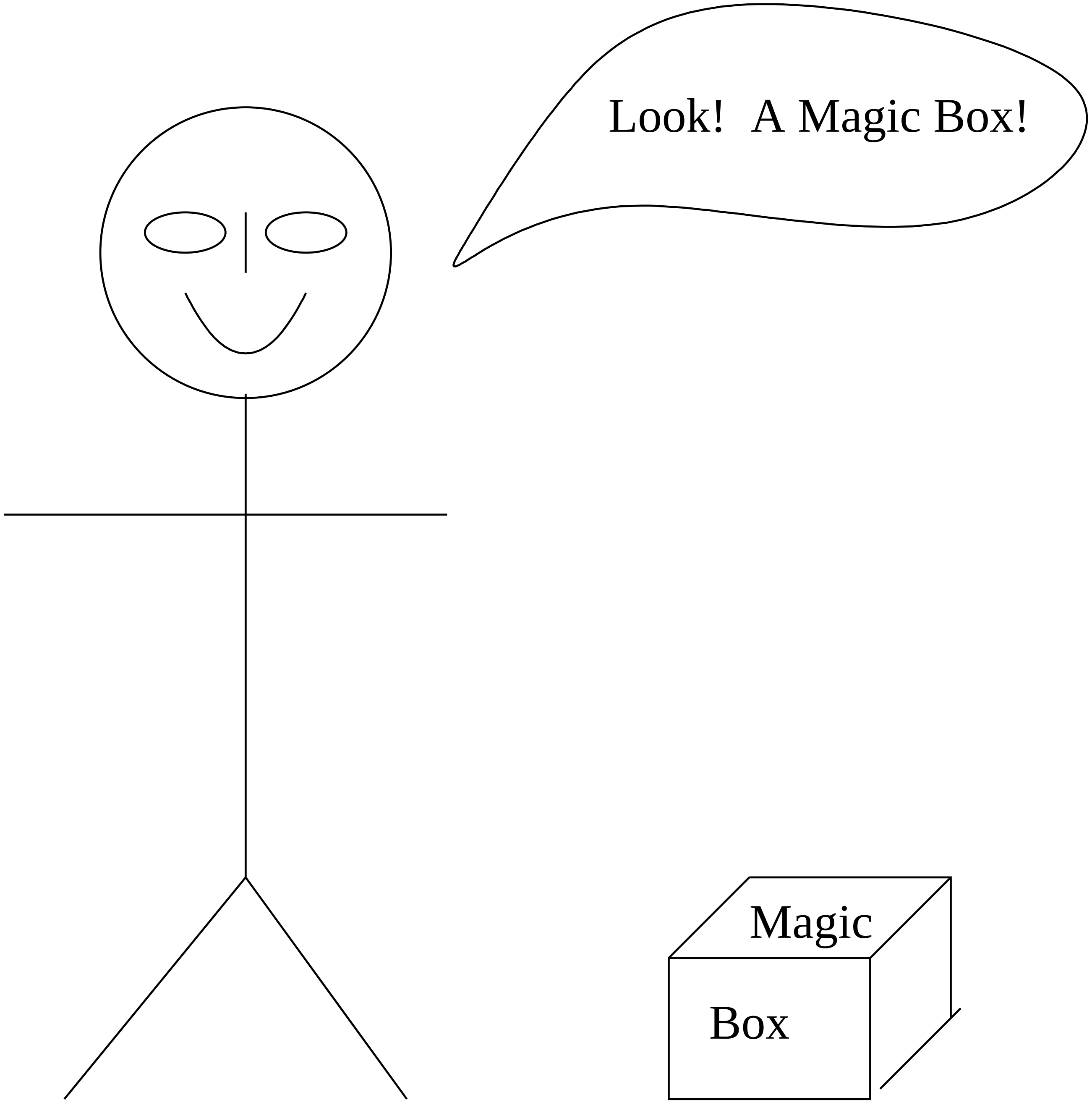
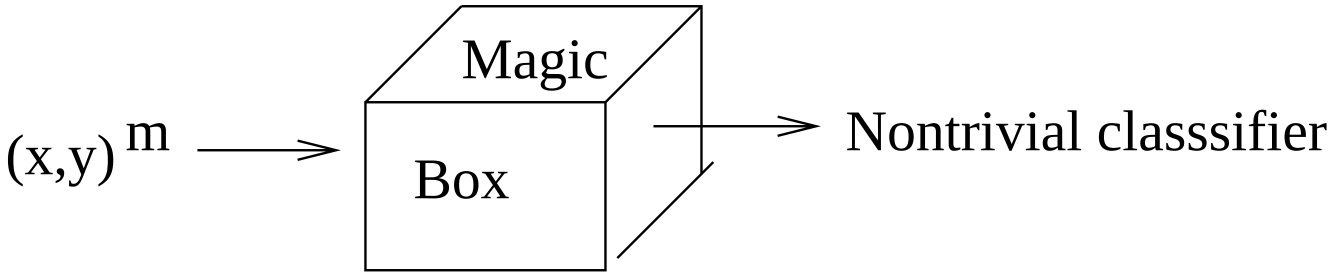
A Little Magic goes a Long Ways

John Langford


Look! A Magic Box!
Magic
Box

Our Magic Box



.. but only barely nontrivial classifier.

$\exists \epsilon > 0 : \ \mathsf{Pr}_{(x,y)\sim D}(c(x) \neq y) \leq \frac{1}{2} - \epsilon$

How can we use the magic box to solve difficult problems?

Outline

1. Algorithms & Theory
   (a) Adaboost: a boosting algorithm
   (b) The boosting property
   (c) Sample complexity issues

2. Empirical evaluation vs. other approaches

Boosting

Given:

1. labeled examples, $S = (x, y)^m$

2. An importance weighted classifier learning algorithm, LEARN

Output:

Highy accurate classifier.

Possible Learning algorithms

1. Many algorithms can be easily modified to use importances
   (a) Decision trees (or Decision "stumps")
   (b) Neural network classifier
   (c) Naive Bayes classifier
2. All classification algorithms can be made to work using "costing" (aka repeated rejection sampling) according to importances.

Next: the Adaboost algorithm

1. $D_1(x, y) = \frac{1}{|S|}$

2. For $t = 1, ..., T$

   (a) $c_t = \text{LEARN}(D_t, S)$

   (b) $\epsilon_t = \text{E}_{x,y \sim D_t} I(c_t(x) \neq y)$

   (c) $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$

   (d) $D_{t+1}(x, y) = D_t(x, y) \frac{e^{-y\alpha_t c_t(x)}}{Z_t}$ (where $Z_t = \text{E}_{x,y \sim D_t} e^{-y\alpha_t c_t(x)}$)

3. Output $c(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t c_t(x)\right)$

Adaboost Analysis

Theorem: (Train set boosting) If LEARN produces a classifier c with loss rate:

$$\Pr_{x,y \sim D_t}(c(x) \neq y) \leq \epsilon$$

then the train error rate of the average is at most $e^{-2T\left(\frac{1}{2}-\epsilon\right)^2}$.

Note: Training set error. There are variants which apply to the "true error" rate.

Boosting Proof:

Let $f(x) = \sum_t \alpha_t c_t(x)$ (so $c(x) = \mathsf{Sign}(f(x))$)

$\hat{c}_S = \mathsf{E}_{x,y\sim S}\, I(c(x) \neq y)$ (Definition)

$\leq \mathsf{E}_{x,y\sim S}\, e^{-yf(x)}$ (since $e^{-yf(x)} \geq I(y \neq c(x))$)

$= \Pi_t Z_t$ (Lemma 1)

$= \Pi_t 2\sqrt{\epsilon_t(1-\epsilon_t)}$ (Lemma 2)

$\leq e^{-2\sum_t (\frac{1}{2}-\epsilon_t)^2}$ (Taylor expansion)

$\leq e^{-2T(\frac{1}{2}-\epsilon)^2}$ (Since $\epsilon_t \leq \epsilon$ by magic)

Lemma 1: $\mathsf{E}_{x,y\sim S}\, e^{-yf(x)} = \Pi_t Z_t$

$$\mathsf{E}_{x,y\sim S}\, e^{-yf(x)}$$

$$= \mathsf{E}_{x,y\sim S}\, e^{-y\sum_t \alpha_t c_t(x)}$$

$$= \mathsf{E}_{x,y\sim S}\, \Pi_t e^{-y\alpha_t c_t(x)}$$

$D_{T+1}(x,y) = \frac{\Pi_t e^{-y\alpha_t c_t(x)}}{|S|\Pi_t Z_t}$ (since $D_{t+1}(x,y) = D_t(x,y)\frac{e^{-y\alpha_t c_t(x)}}{Z_t}$)

$\Rightarrow \mathsf{E}_{x,y\sim S}\, \Pi_t e^{-y\alpha_t c_t(x)} = \Pi_t Z_t$ (since $\sum_{x,y} D_{T+1}(x,y) = 1$)

Lemma 2: $\forall t \;\; Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$

$$Z_t = \mathsf{E}_{x,y\sim D_t}\, e^{-\alpha_t y c_t(x)}$$

$$= \mathsf{E}_{x,y\sim D_t}\, e^{-y c_t(x)\frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}}$$

$$= \mathsf{E}_{x,y\sim D_t}\left(\tfrac{\epsilon_t}{1-\epsilon_t}\right)^{\frac{1}{2} y c_t(x)}$$

$$= \epsilon_t\, \mathsf{E}_{x,y\sim D_t}\left(\tfrac{1-\epsilon_t}{\epsilon_t}\right)^{\frac{1}{2}} + (1-\epsilon_t)\, \mathsf{E}_{x,y\sim D_t}\left(\tfrac{\epsilon_t}{1-\epsilon_t}\right)^{\frac{1}{2}}$$

$$= 2\sqrt{(1-\epsilon_t)\epsilon_t}$$

Boosting side notes

Variants for

1. real-valued outputs
2. multiclass classification
3. different update functions
4. DAG structures

MANY Variants (boosting.org)

Sample Complexity Issues

Boosting performance NOT explained by Occam's Razor bound.

... but there are many other train set bounds, many of which motivate averages.

1. Margin Bound (Schapire, Freund, Bartlett, Lee, 1998)

2. PAC-Bayes Bound (McAllester, 1999)

3. Stochastic Margin Bound (Langford and Shawe-Taylor, 2002)

4. (many others...)

Outline

1. Algorithms and Theory

2. Empirical evaluation vs. other approaches

   (a) Bagging: a strawman alternative

   (b) Empirical results

An alternative algorithm: Bagging

Given: training examples S, Learning algorithm Learn

1. Repeat $t = 1...T$ times
   (a) $S' = |S|$ samples drawn with replacement from S
   (b) $c_t = \text{Learn}(S)$
2. Return $c(x) = \text{sign}\left(\sum_t c_t(x)\right)$

Bagging: Analysis

Question: How many unique examples are in S'?

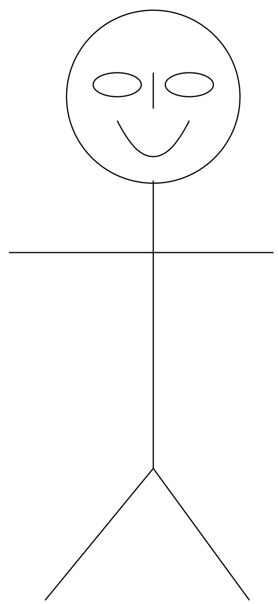
Answer: about $1 - \frac{1}{e}$

Bagging: Analysis

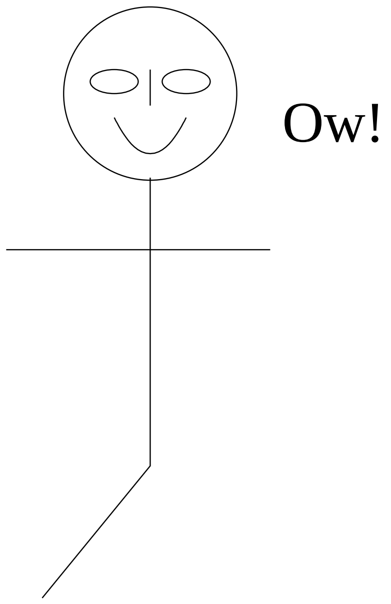
Question: What is the effect of duplicates?

Answer: They can weaken complexity control

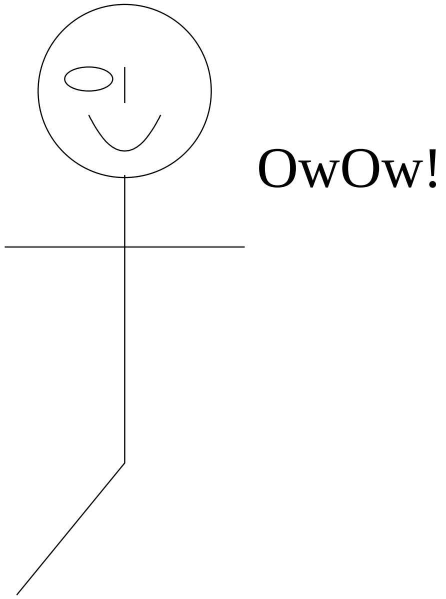
A Learning Algorithm



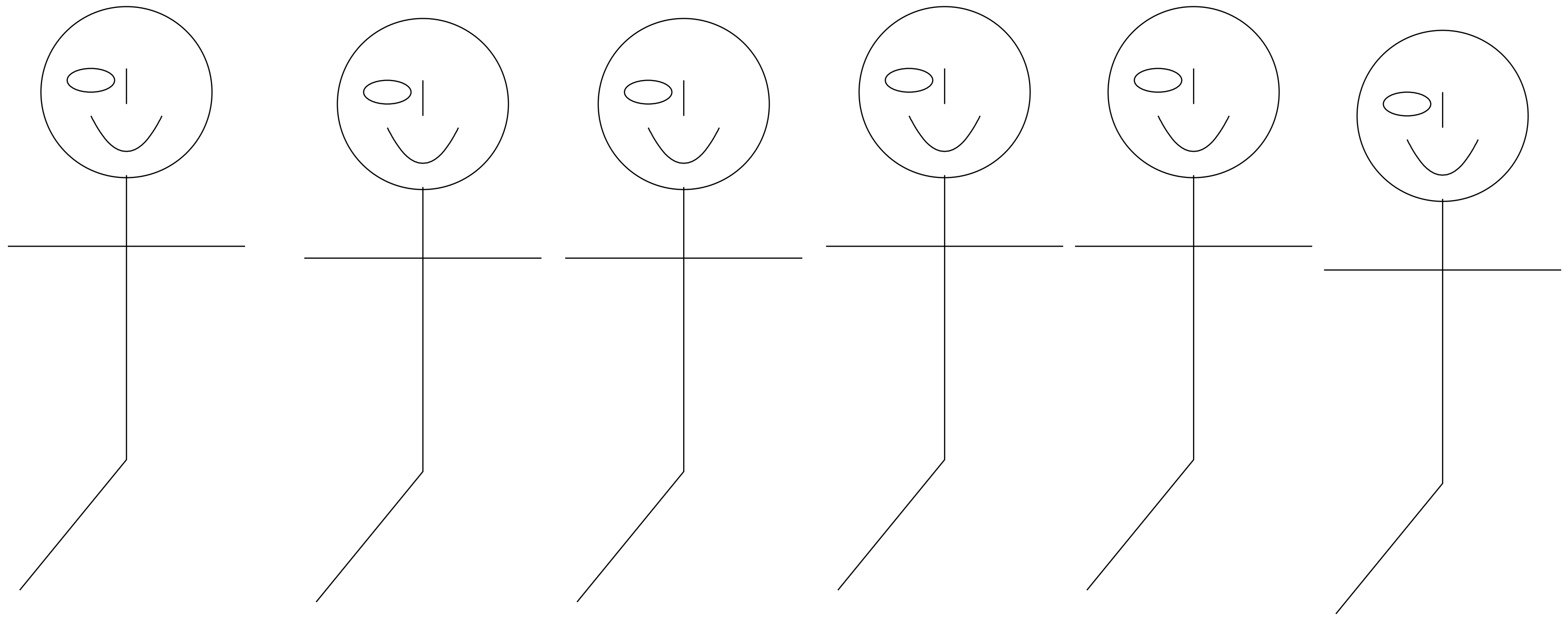
A Learning Algorithm missing $\frac{1}{e}$ of all examples

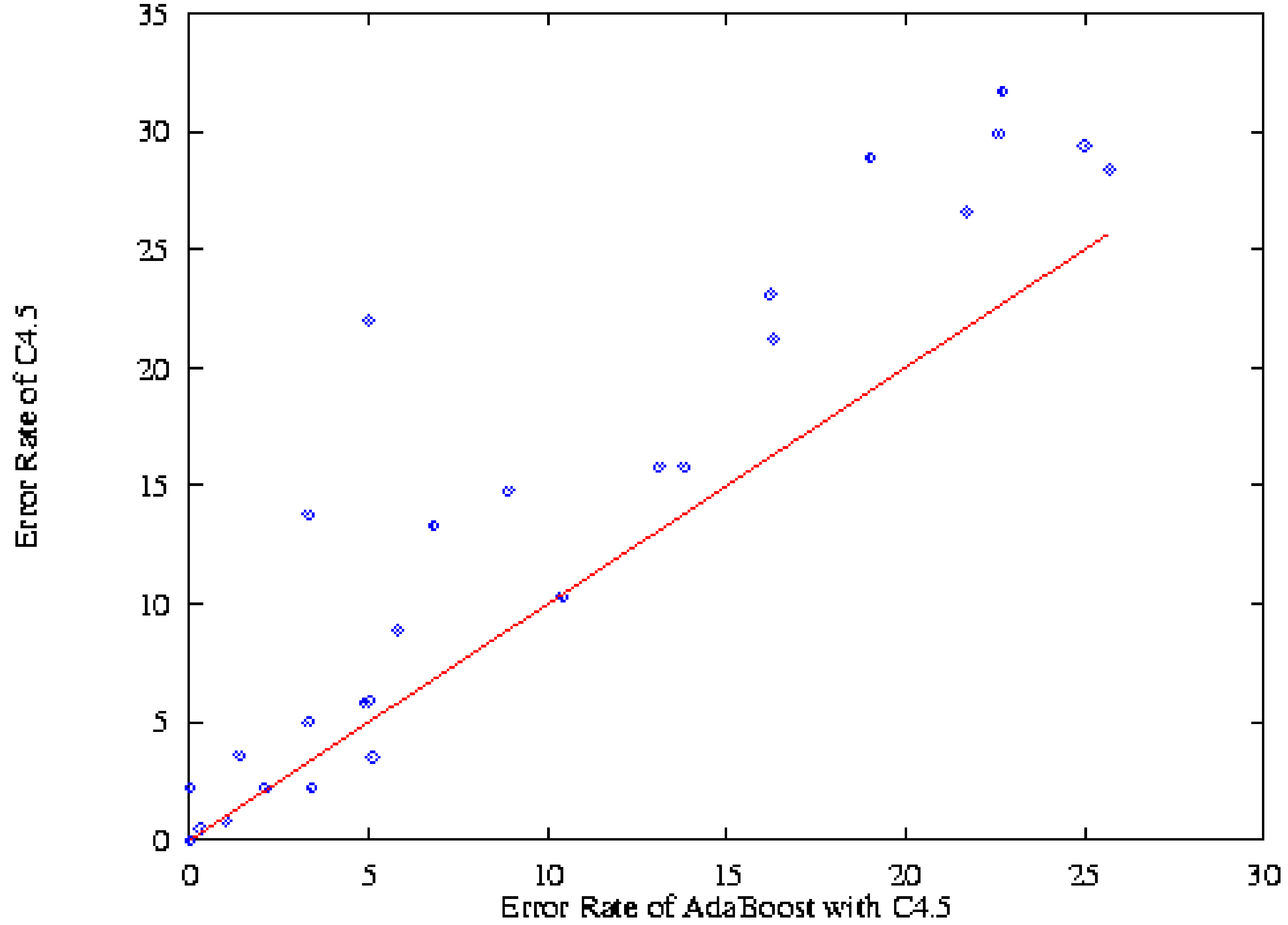


A Learning Algorithm missing $\frac{1}{e}$ examples and with duplicates

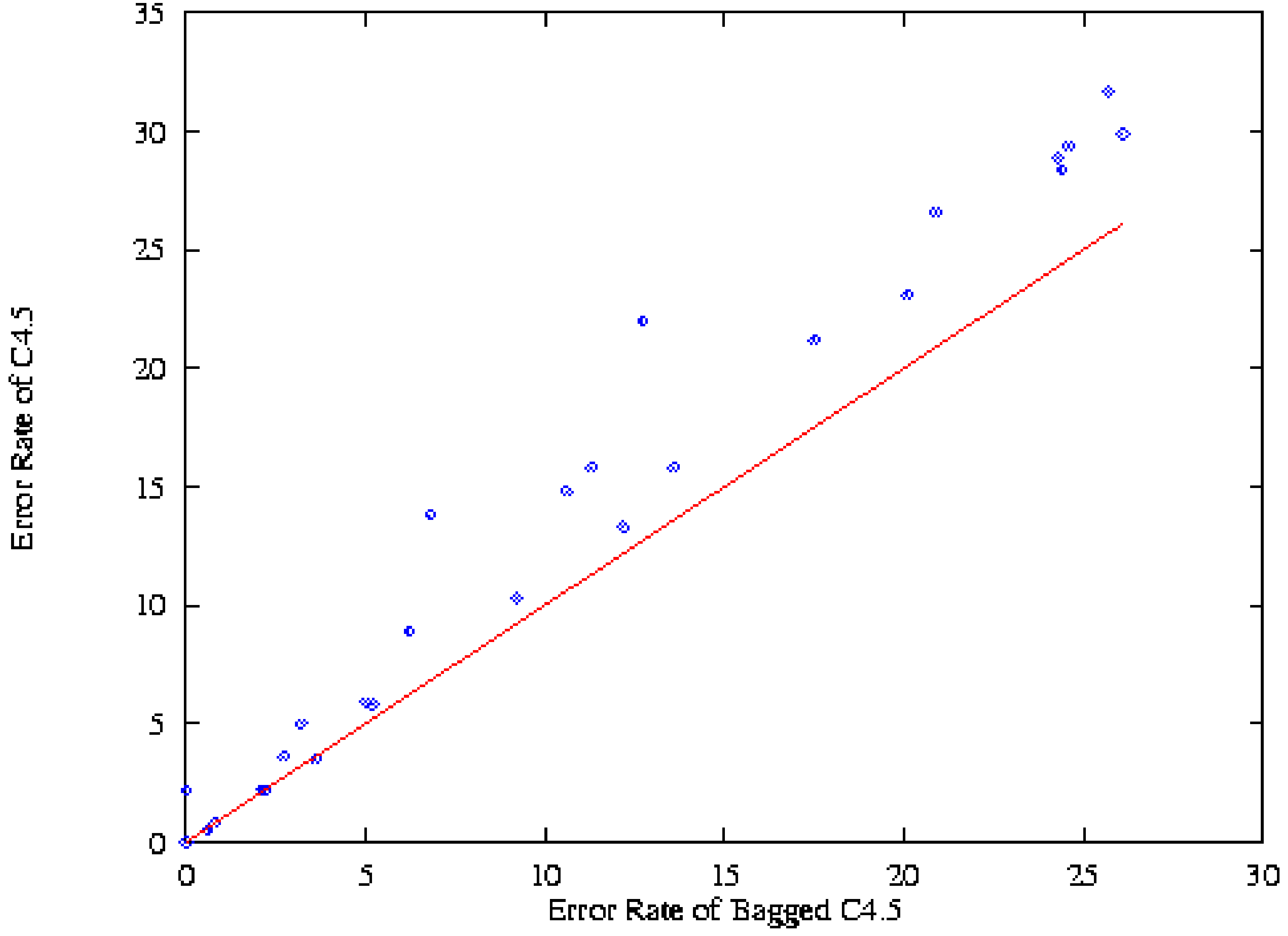


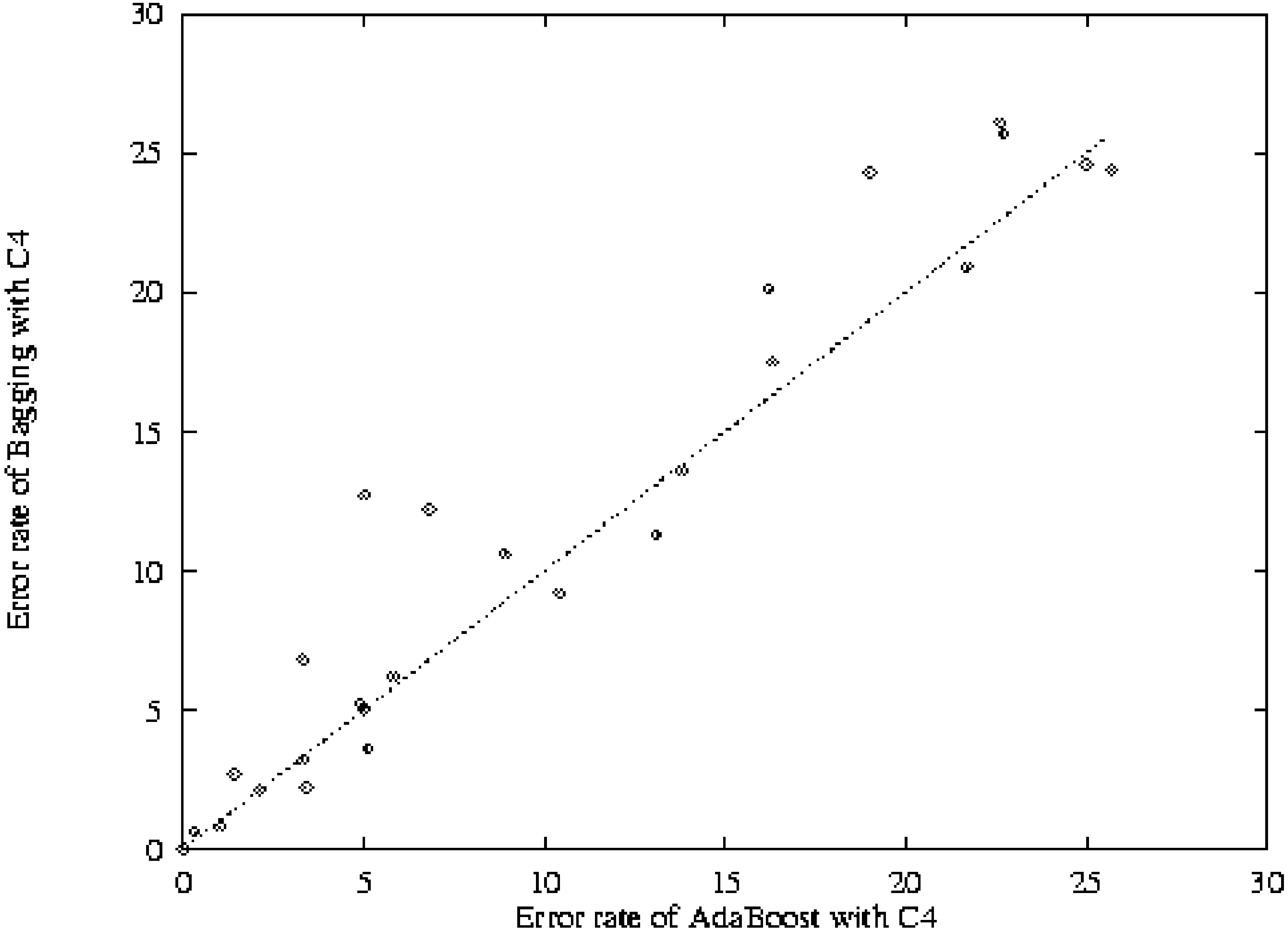
Bagging: Learning algorithm loses $\frac{1}{e}$ examples, gains duplicates, and is averaged





Error Rate of C4.5
Error Rate of AdaBoost with C4.5
0
5
10
15
20
25
30
35


Error Rate of C4.5
Error Rate of Bagged C4.5
0
5
10
15
20
25
30
35


Error rate of Bagging with C4
Error rate of AdaBoost with C4
0
5
10
15
20
25
30

Conclusion

Averaging techniques *dominate* in supervised classification learning.

Boosting has more motivation than many others.

A little magic goes a long ways.