The Test Set Bound and the Occam's Razor Bound

John Langford

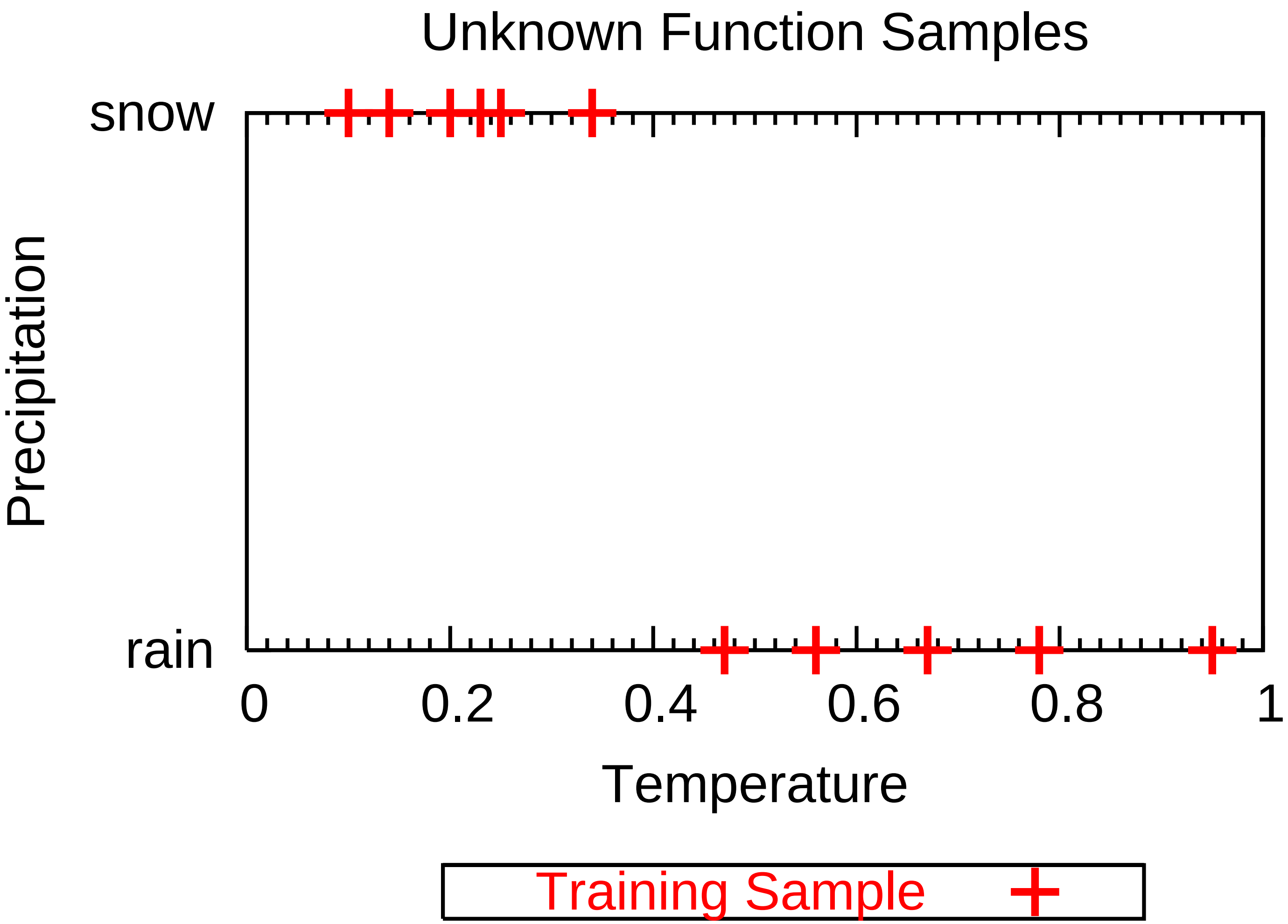
TTI-Chicago

The Classification Problem:

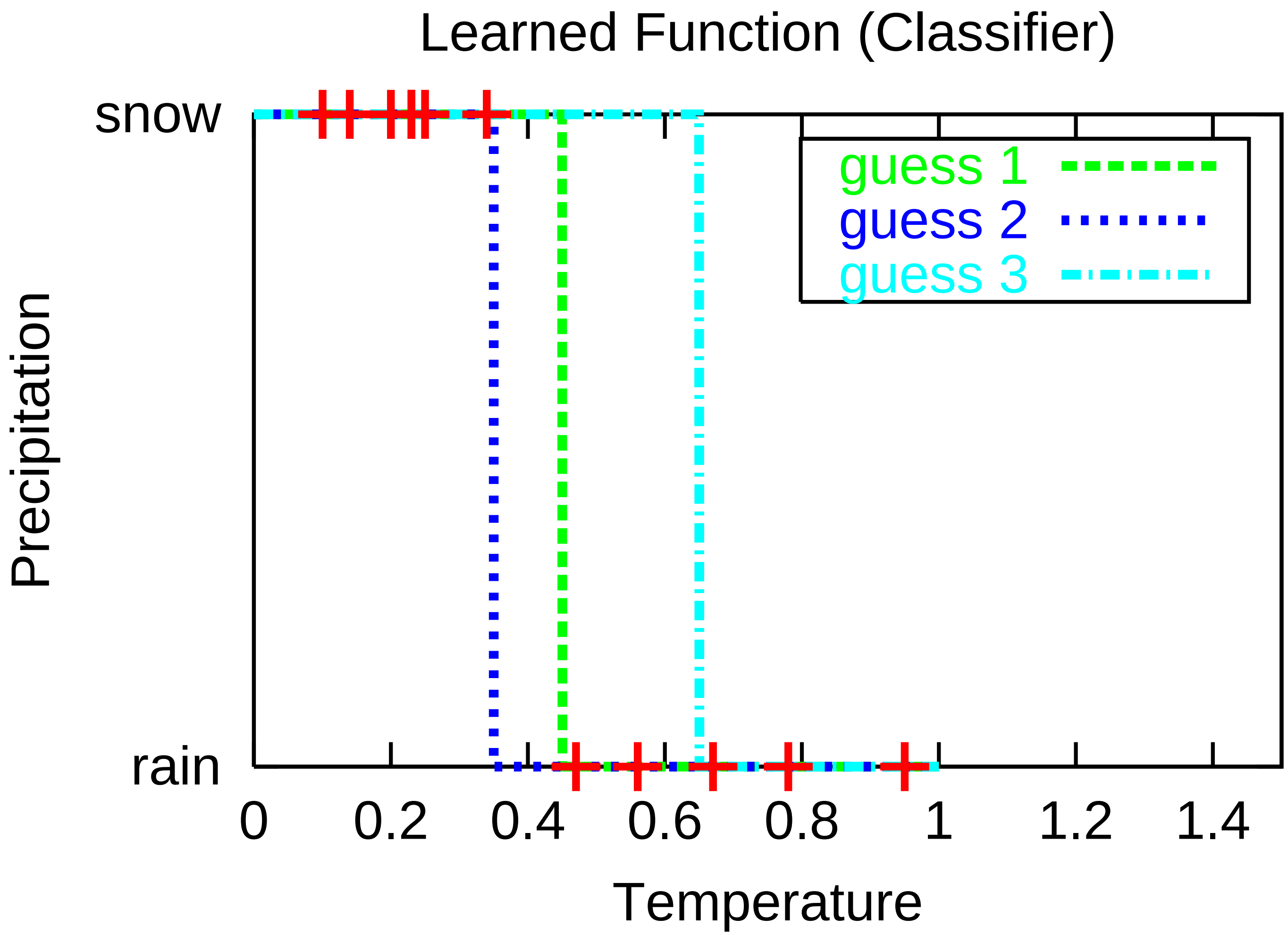
1. Get (input, class label) pairs from unknown function

2. Find the unknown function

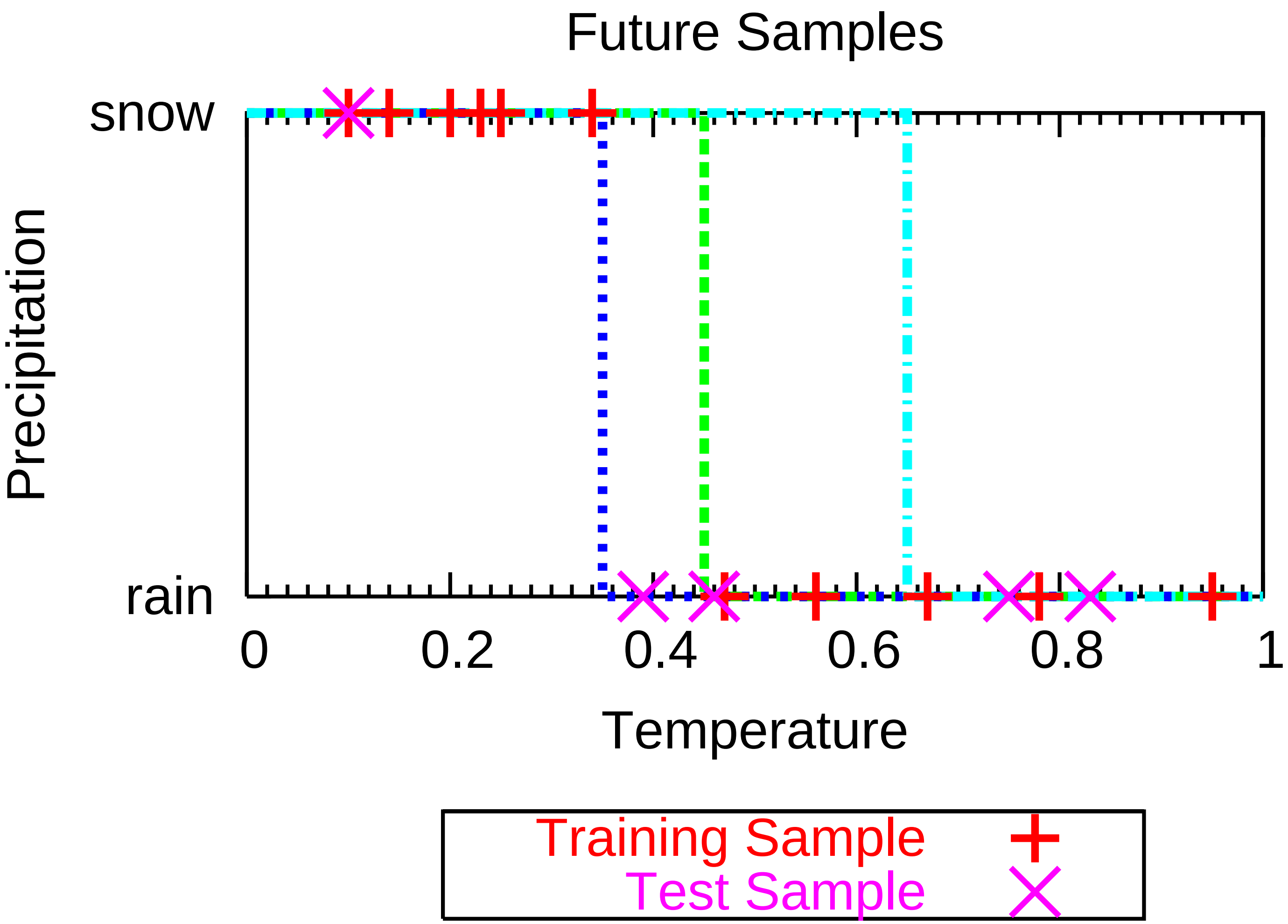
3. ... to minimize errors on future inputs



Unknown Function Samples
snow
rain
Precipitation
0
0.2
0.4
0.6
0.8
1
Temperature
Training Sample



Learned Function (Classifier)
snow
rain
Precipitation
0
0.2
0.4
0.6
0.8
1
1.2
1.4
Temperature
guess 1
guess 2
guess 3


Future Samples
snow
rain
Precipitation
0
0.2
0.4
0.6
0.8
1
Temperature
Training Sample
Test Sample

Learning = Prediction ability

- We can't expect any prediction ability, in general.

- We can expect prediction ability, if examples come independently, sometimes.

Here we study prediction ability, assuming indepedence.

Why study prediction theory?

1. Better methods for learning and verifying predictive ability

2. To gain insight into learning.

Better Methods for Learning & Verification

Standard technique:

1. Divide samples into train and test set
2. Train on train set
3. Test on test set

We can do better.

To gain insight into learning

1. Overfitting: sample complexity quantifies overfitting.
2. Learning algorithm design: What is a good pruning criterion? Why are large margins good? What other algorithms are likely to yield good results?

Outline

1. The Basic Model
2. The Test Set Bound
3. Occam's Razor Bound

Model: Definitions

X = input space

$Y = \{0, 1\}$ = output space

$c : X \to Y$ = classifier

Model: Basic Assumption

All samples are drawn independently from some unknown distribution $D(x, y)$.

$S = (x, y)^m \sim D^m$ is a sample set.

Model: Derived quantities

The thing we want to know:

$$c_D \equiv \Pr_{x,y \sim D}(c(x) \neq y) = \text{true error}$$

Model: Derived quantities

The thing we want to know:

$$c_D \equiv \Pr_{x,y\sim D}(c(x) \neq y) = \text{true error}$$

The thing we have:

$$\hat{c}_S \equiv \Pr_{x,y\sim S}(c(x) \neq y) = \frac{1}{m}\sum_{i=1}^{m} I\left[c(x) \neq y\right]$$

= "train error", "test error", or "observed error", depending on context.

(note: we identify the set S with the uniform distribution on S)

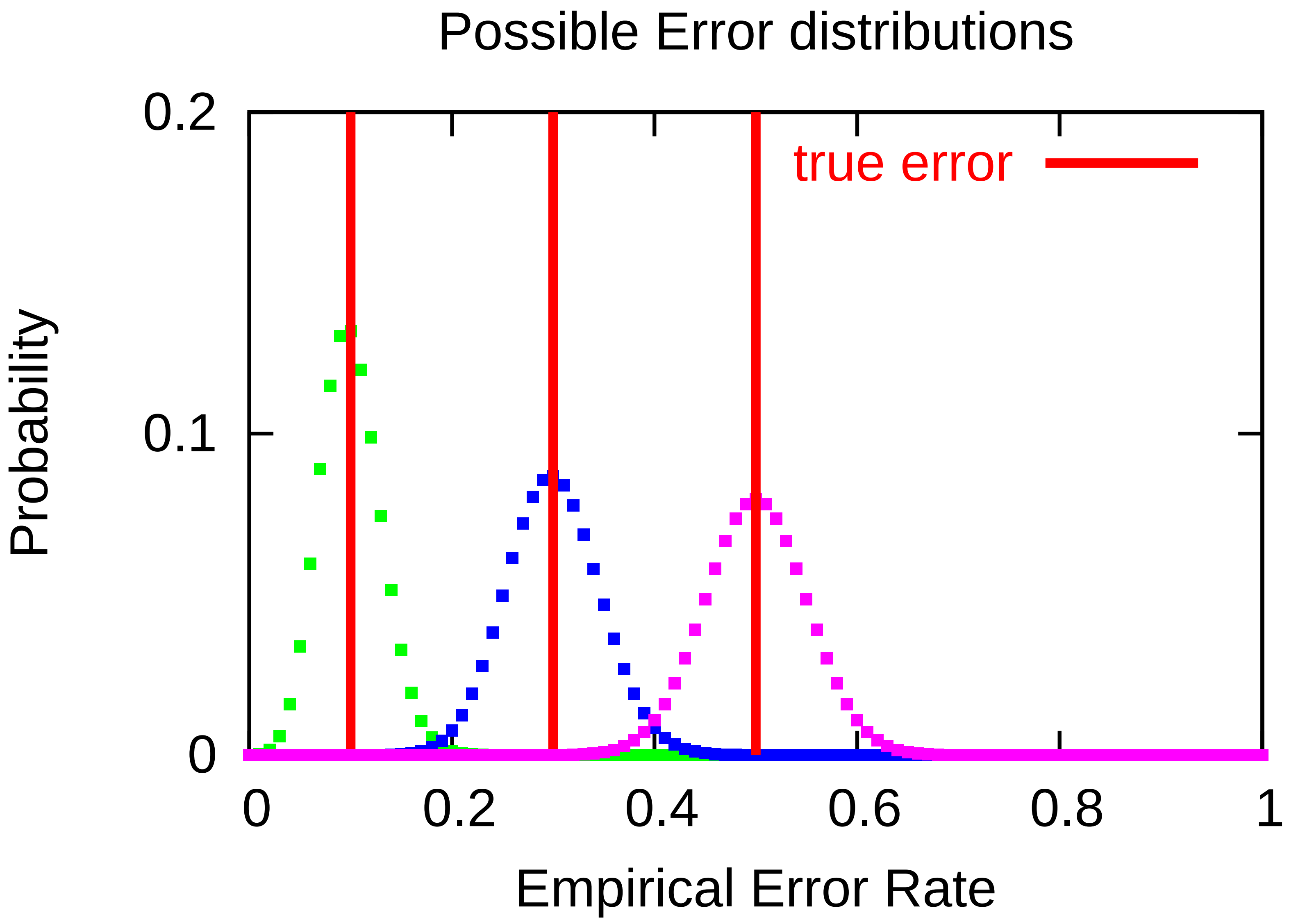
Model: Basic Observations

Q: What is the distribution of $\hat{c}_S$?

A: A Binomial.

$$\Pr_{S \sim D^m}\left(\hat{c}_S = \frac{k}{m} \middle| c_D\right) = \binom{m}{k} c_D^k (1 - c_D)^{m-k}$$

$=$ probability of k heads (errors) in m flips of a coin with bias c_D.



Possible Error distributions
true error
Probability
Empirical Error Rate
0
0.1
0.2
0
0.2
0.4
0.6
0.8
1

Model: basic quantities

We use the cumulative:

$$\begin{aligned}\mathrm{Bin}\left(\frac{k}{m}, c_D\right) &= \quad \Pr_{S\sim D^m}\left(\hat{c}_S \leq \frac{k}{m}\middle| \, c_D\right) \\ &= \sum_{i=0}^{k} \binom{m}{i} c_D^i (1-c_D)^{m-i}\end{aligned}$$

= probability of observing k or fewer "heads" (errors) with m coins.

Model: basic quantities

Need confidence intervals $\Rightarrow$ use the pivot of the cumulative instead

$$\overline{\mathrm{Bin}}\left(\frac{k}{m}, \delta\right) = \max\left\{p : \mathrm{Bin}\left(\frac{k}{m}, p\right) \geq \delta\right\}$$

= the largest true error such that the probability of observing k or fewer "heads" (errors) is at least δ.

Outline

1. The Basic Model
2. The Test Set Bound
3. Occam's Razor Bound

Test Set Bound: Setting

Standard technique:

1. Cut the data into train set and test set
2. Train on the train set
3. Test on the test set

Test Set Bound: Theorem

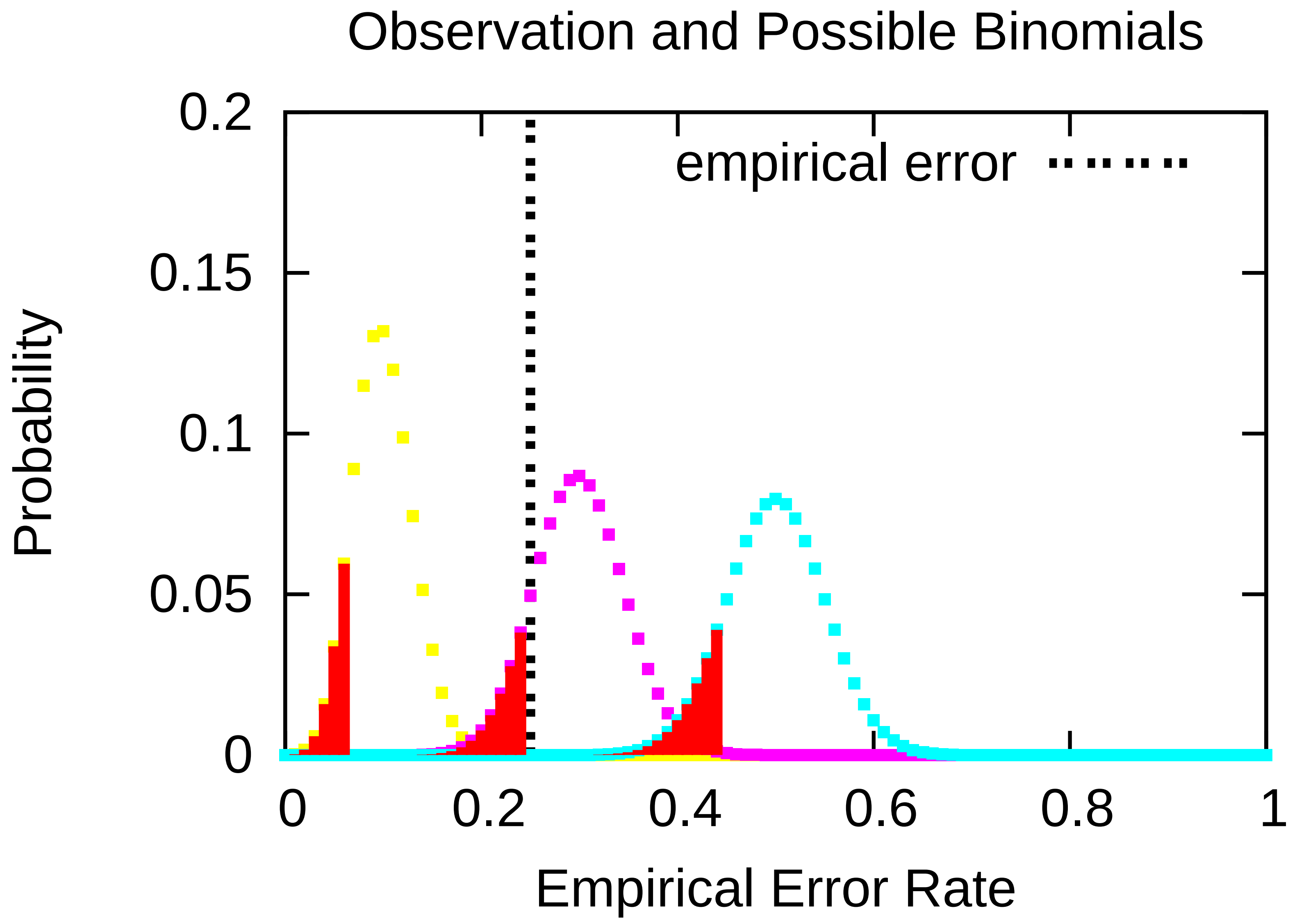
Theorem: (Test Set Bound) For all classifiers c, for all D, for all $\delta \in (0,1]$:

$$\Pr_{S \sim D^m}\left(c_D \leq \overline{\text{Bin}}\left(\hat{c}_S, \delta\right)\right) \geq 1 - \delta$$

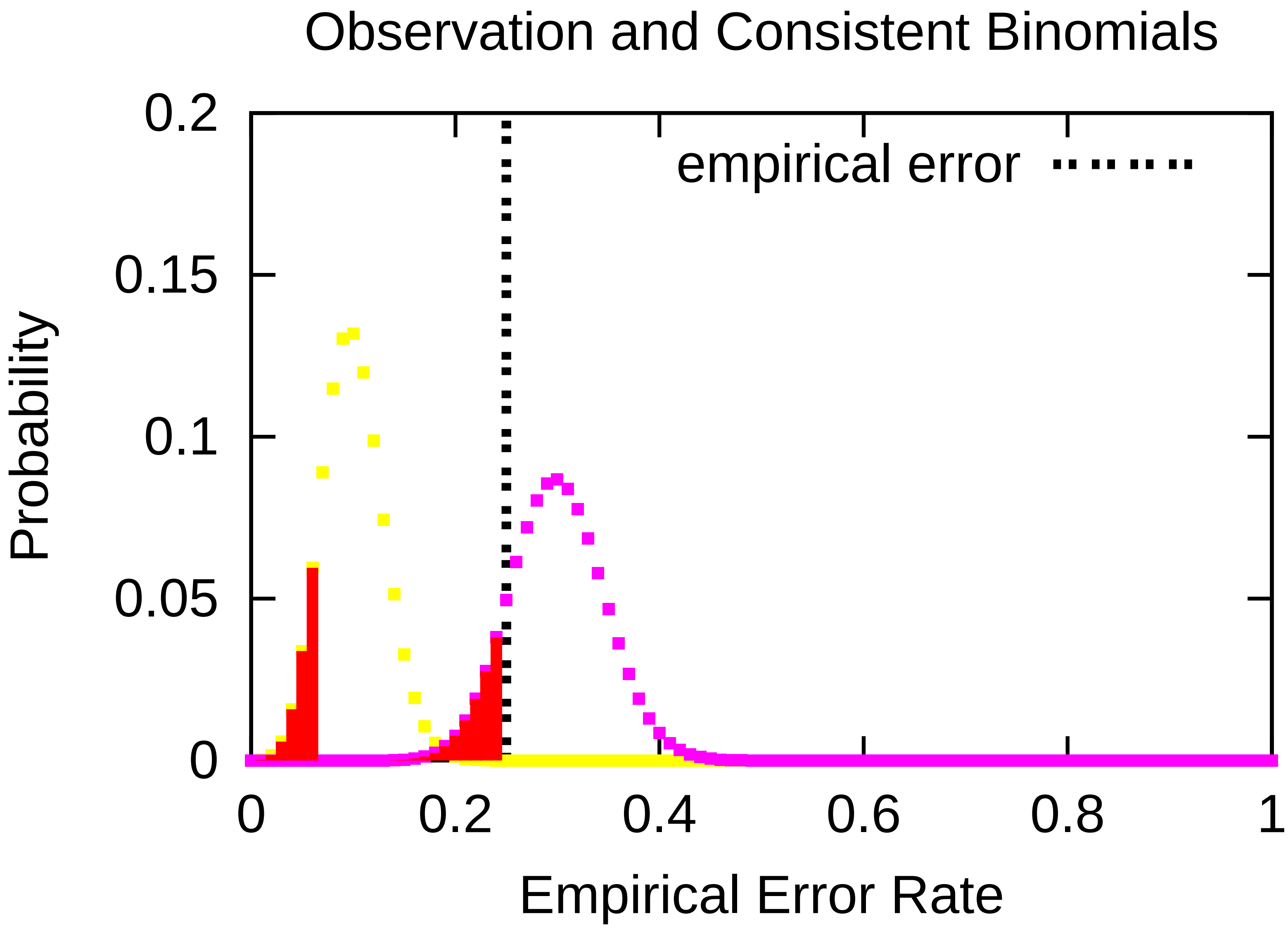
World's easiest proof:

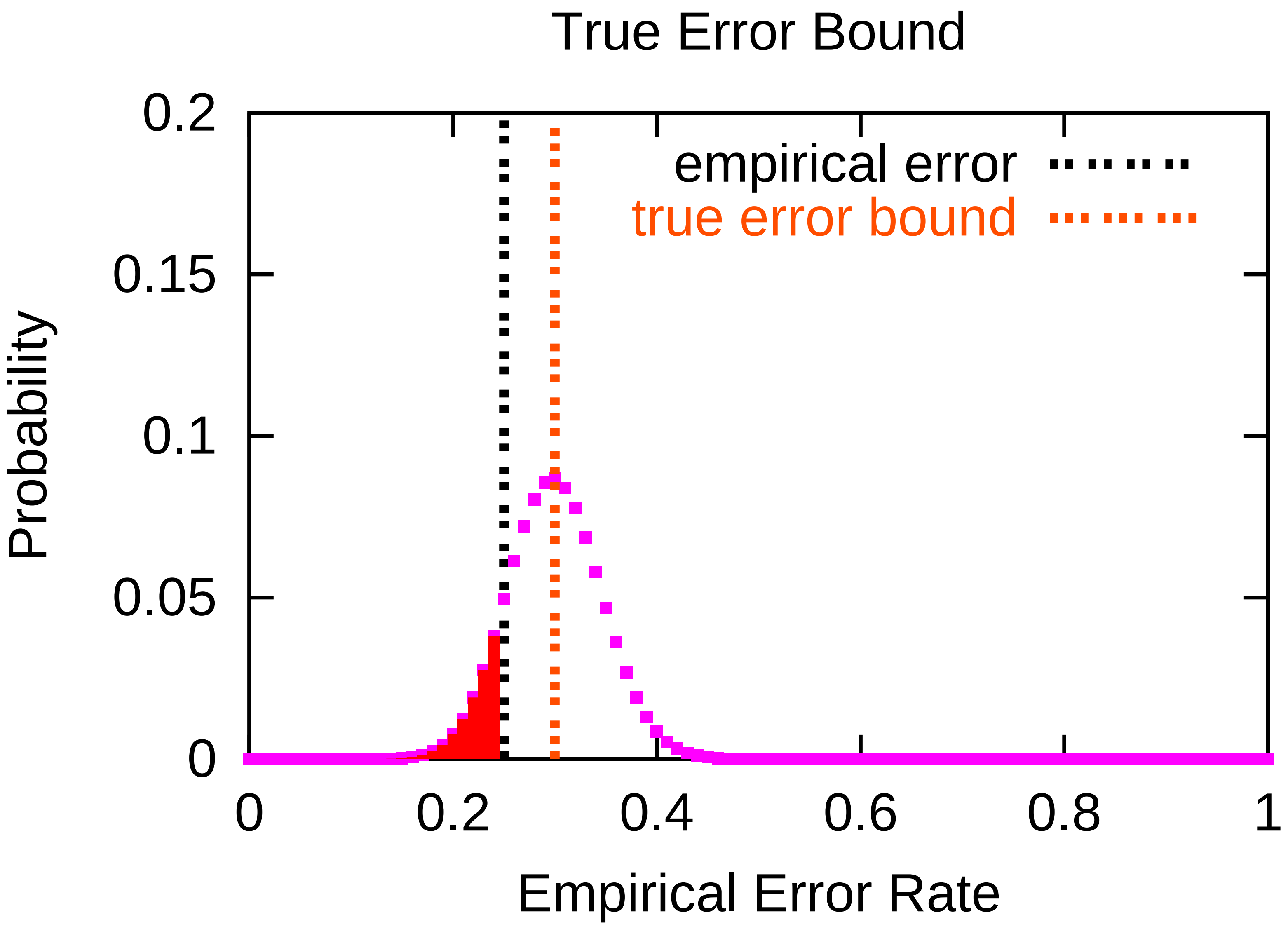
With probability $1-\delta$, $\text{Bin}\left(\frac{k}{m}, c_D\right) \geq \delta$

Then by definition, $\overline{\text{Bin}}\left(\hat{c}_S, \delta\right) \geq c_D$



Observation and Possible Binomials
empirical error
Probability
Empirical Error Rate
0.2
0.15
0.1
0.05
0
0
0.2
0.4
0.6
0.8
1


Observation and Consistent Binomials
empirical error
Probability
Empirical Error Rate
0
0.05
0.1
0.15
0.2
0
0.2
0.4
0.6
0.8
1


True Error Bound
empirical error
true error bound
Probability
Empirical Error Rate
0
0.05
0.1
0.15
0.2
0
0.2
0.4
0.6
0.8
1

Test Set Bound Notes

Perfectly tight: There exist true error rates achieving the bound

Lower bound of the same form.

Primary use: verification of succesful learning

What does Test Set Bound mean?

Corollary: For all classifiers c, for all D, for all $\delta \in (0,1]$:

$$\Pr_{S \sim D^m}\left(c_D \leq \hat{c}_S + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}} \right) \geq 1 - \delta$$

Proof: Use the Chernoff approximation,

$$\overline{\mathrm{Bin}}\left(\hat{c}_S, \delta\right) \leq \hat{c}_S + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}$$

Note: NOT tight when $\hat{c}_S$ near 0 (our goal!)

Test Set Bound: Example

Suppose $\delta = 0.1$

Suppose $m = 100$

Suppose $\hat{c}_S = \frac{38}{100}$

Square root Chernoff bound: $\Rightarrow c_D \in [0.26, 0.50]$

Exact calculation $\Rightarrow c_D \in [0.30, 0.47]$

Test Set Bound Comparison: Empirical "confidence" intervals

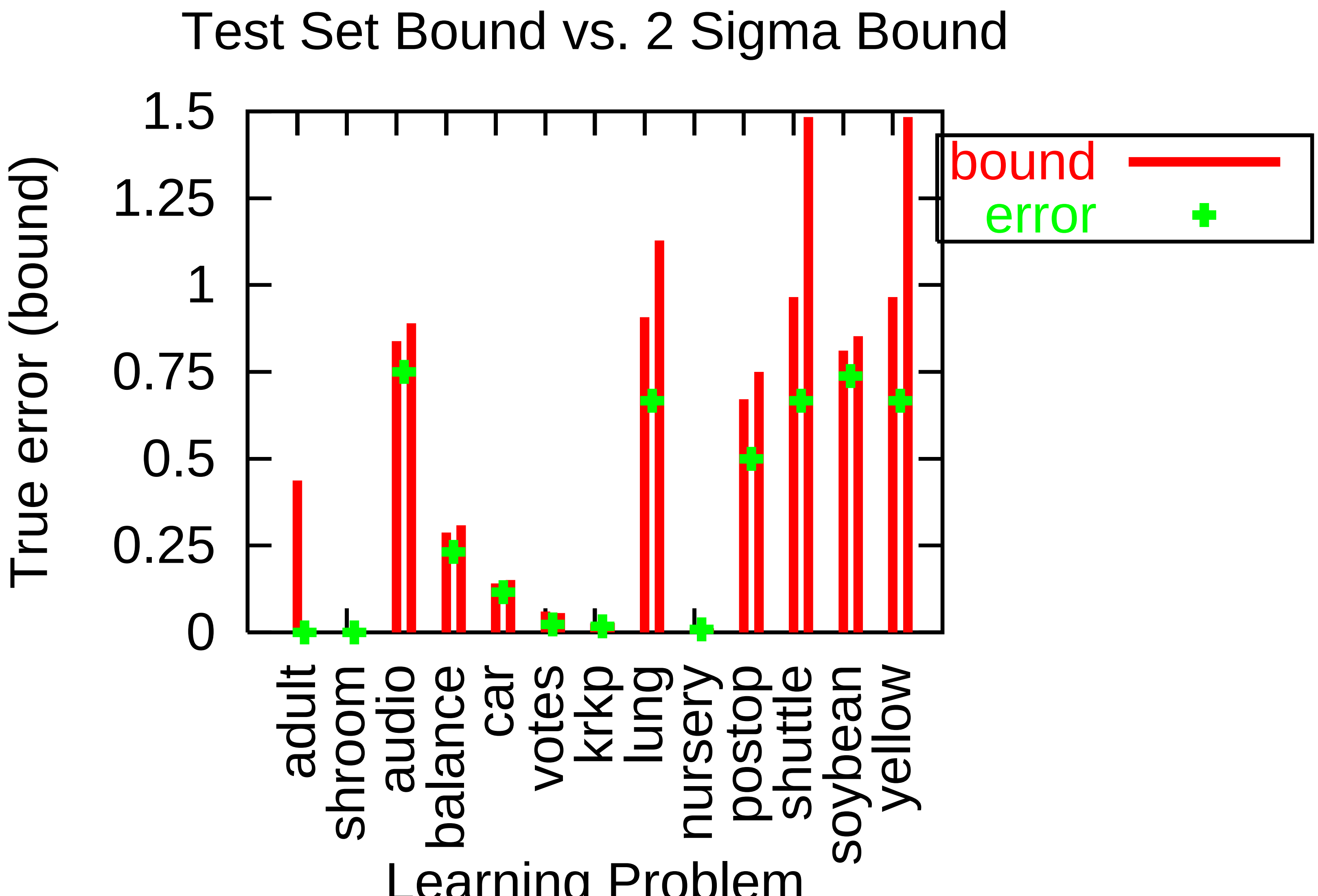
k = number of test errors, m = number of examples

$\mu = \frac{k}{m}$

$\sigma^2 = \frac{1}{m-1}\frac{1}{m}\sum_{i=1}^{m}\left(\mu - I\left[c(x_i) \neq y_i\right]\right)^2 = \frac{\mu(1-\mu)}{m-1}$

pick bound $= \frac{k}{m} + 2\sigma$

How do they compare?



Test Set Bound vs. 2 Sigma Bound
True error (bound)
1.5
1.25
1
0.75
0.5
0.25
0
bound
error
adult
shroom
audio
balance
car
votes
krkp
lung
nursery
postop
shuttle
soybean
yellow
Learning Problem

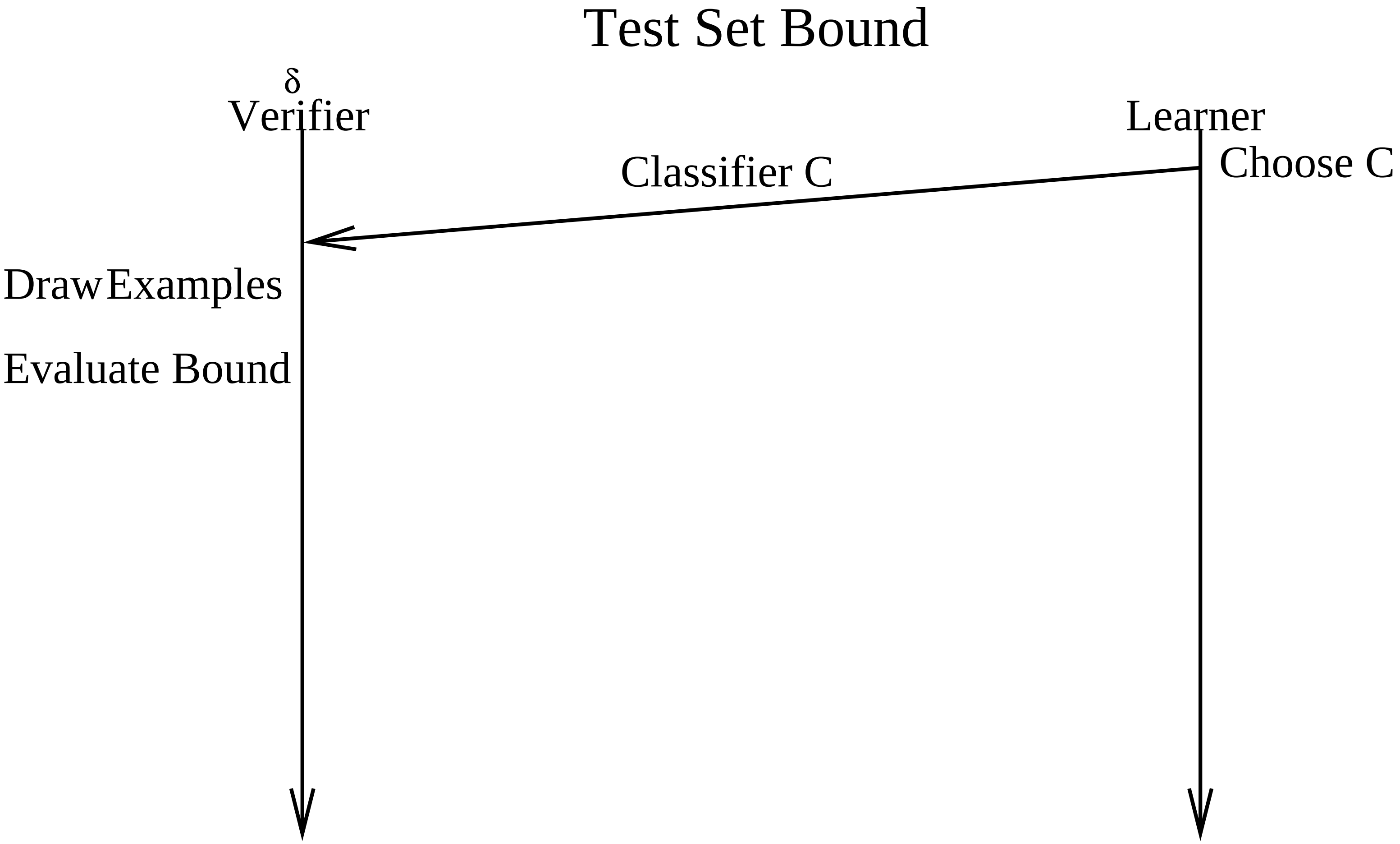
Test Set Bound vs Empirical Confidence Interval

1. empirical confidence intervals are sometimes pessimistic

2. empirical confidence intervals are sometimes optimistic

3. the test set bound always works

Interpretation: Interactive Proof of Learning



K-fold Cross Validation

Divide m examples into K subsets.

Repat K times: Train on $K-1$ subsets, test on heldout subset.

Not well understood theoretically.

Best Result: Confidence interval smaller than a test set of size $\frac{m}{K}$.

$\Rightarrow$ leave-one-out cross validation very prone to overfitting.

$\Rightarrow$ Some people fool themselves with overconfidence in Cross Validation.

Outline

1. The Basic Model
2. The Test Set Bound
3. Occam's Razor Bound

Training Set Bounds in General

- Sometimes the holdout set is *critical* for learning.
- Sometimes we want bounds to guide learning

$\Rightarrow$Train set bounds

Occam's Razor bound is the simplest train set bound.

A note on Practicality

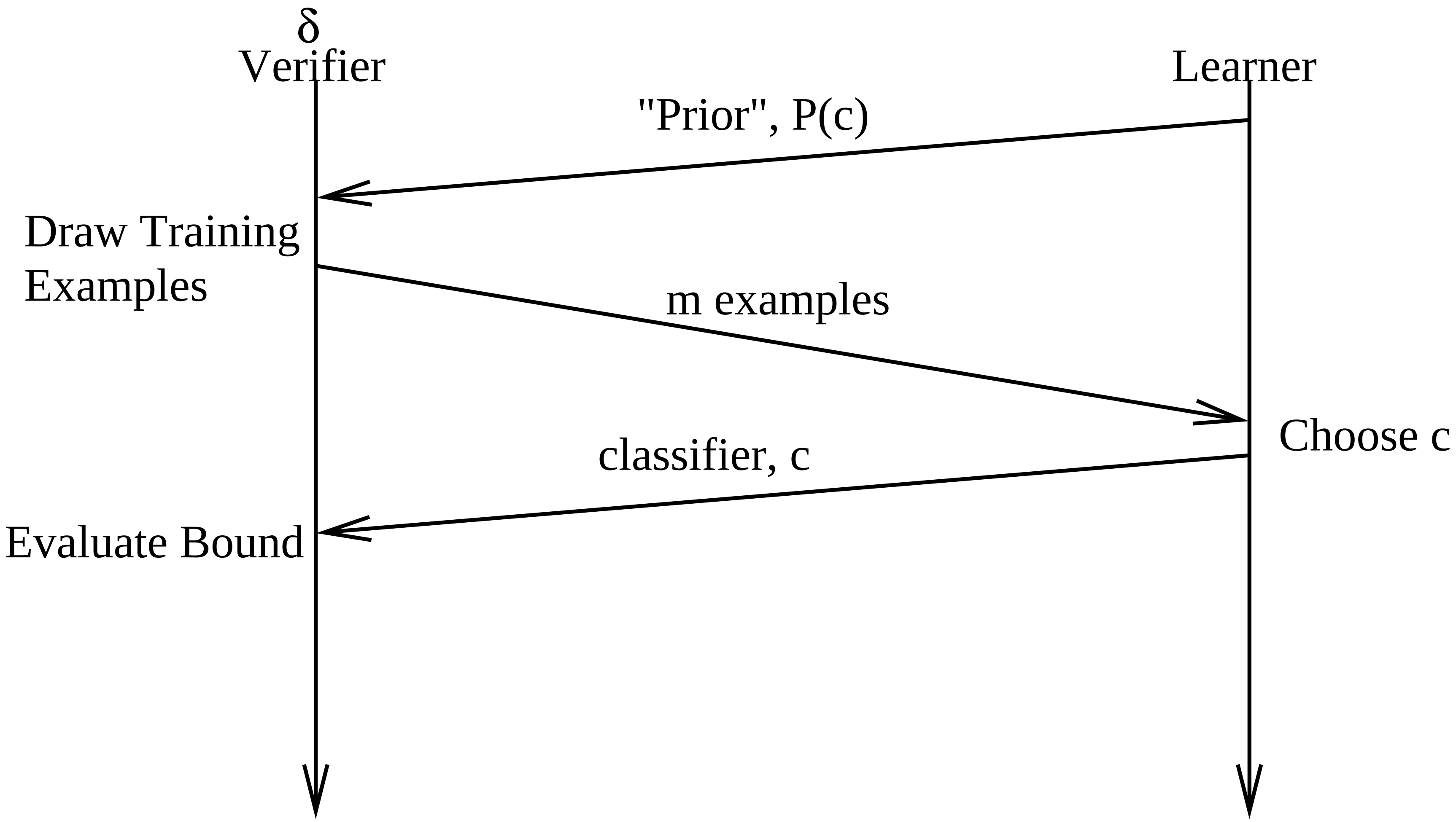
Many Train set bounds are not practical: They are too loose.

1. VC dimension bounds
2. Fat-shattering dimension bounds
3. Covering number bounds

We do not cover these here: the emphasis is on (in practice) tight bounds.

Despite lack of tightness, these other bounds can motivate learning algorithms.

Occam's Razor Bound Protocol



Occam's Razor Bound

Theorem: (Occam's Razor Bound) For all "priors" $P(c)$ over the classifiers c, for all D, for all $\delta \in (0,1]$:

$$\Pr_{S \sim D^m}\left(\forall c:\ c_D \leq \overline{\mathrm{Bin}}\left(\hat{c}_S, \delta P\left(c\right)\right)\right) \geq 1 - \delta$$

Compare with test set bound: $\delta \to \delta P(c)$.

Occam's Razor Bound: Proof

First note: $P(c) = 0 \Rightarrow \overline{\text{Bin}}\left(\hat{c}_S, 0\right) = 1$.

Test set bound $\Rightarrow$

$$\forall c \quad \Pr_{S \sim D^m}\left(c_D \leq \overline{\text{Bin}}\left(\hat{c}_S, \delta P(c)\right)\right) \geq 1 - \delta P(c)$$

Occam's Razor Bound: Proof

First note: $P(c) = 0 \Rightarrow \overline{\text{Bin}}\left(\hat{c}_S, 0\right) = 1$.

Test set bound $\Rightarrow$

$$\forall c \quad \Pr_{S \sim D^m}\left(c_D \leq \overline{\text{Bin}}\left(\hat{c}_S, \delta P(c)\right)\right) \geq 1 - \delta P(c)$$

Negate to get:

$$\forall c \quad \Pr_{S \sim D^m}\left(c_D > \overline{\text{Bin}}\left(\hat{c}_S, \delta P(c)\right)\right) < \delta P(c)$$

Occam's Razor Bound: Proof

First note: $P(c) = 0 \Rightarrow \overline{\text{Bin}}\,(\hat{c}_S, 0) = 1$.

Test set bound $\Rightarrow$

$$\forall c \quad \Pr_{S\sim D^m}\Big(c_D \leq \overline{\text{Bin}}\,(\hat{c}_S, \delta P(c))\Big) \geq 1 - \delta P(c)$$

Negate to get:

$$\forall c \quad \Pr_{S\sim D^m}\Big(c_D > \overline{\text{Bin}}\,(\hat{c}_S, \delta P(c))\Big) < \delta P(c)$$

Apply union bound: $\Pr(A \text{ or } B) \leq \Pr(A) + \Pr(B)$ repeatedly.

$$\Pr_{S\sim D^m}\Big(\exists c : \; c_D > \overline{\text{Bin}}\,(\hat{c}_S, \delta P(c))\Big) < \sum_c \delta P(c) = \delta$$

Occam's Razor Bound: Proof

First note: $P(c) = 0 \Rightarrow \overline{\mathsf{Bin}}(\hat{c}_S, 0) = 1$.

Test set bound $\Rightarrow$

$$\forall c \quad \Pr_{S \sim D^m}\left(c_D \leq \overline{\mathsf{Bin}}(\hat{c}_S, \delta P(c))\right) \geq 1 - \delta P(c)$$

Negate to get:

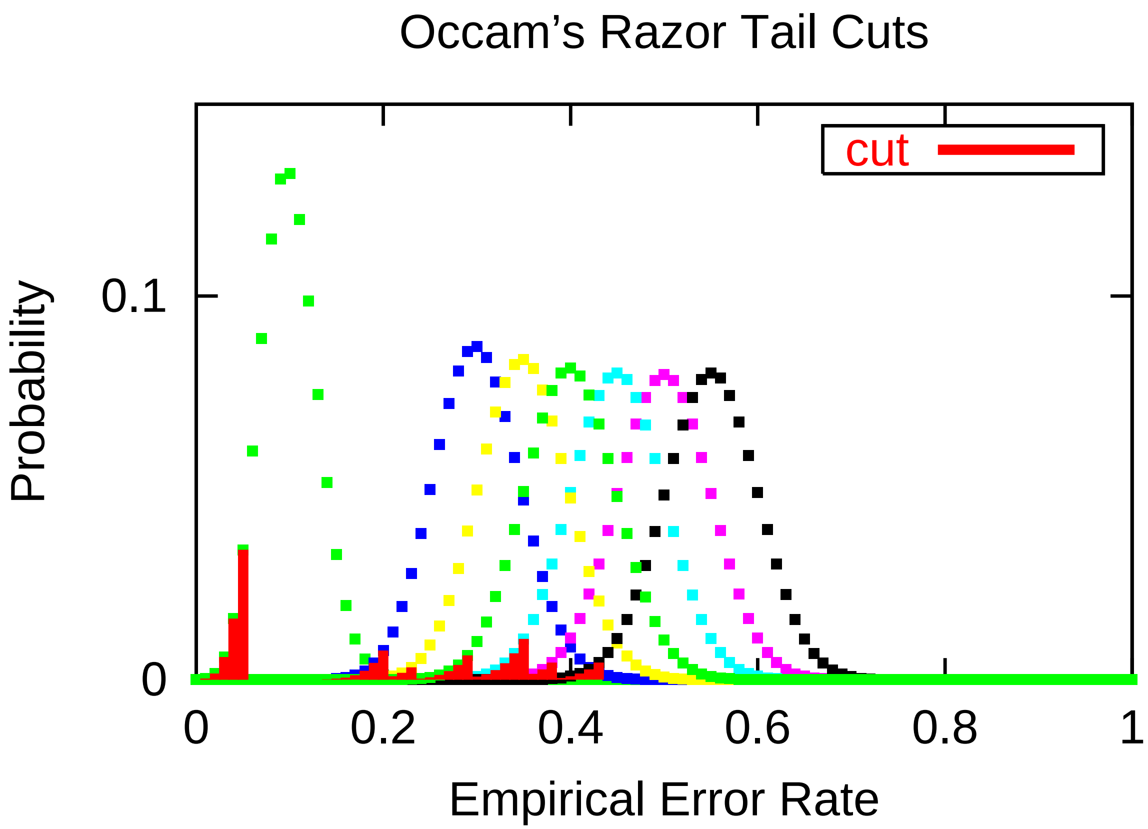
$$\forall c \quad \Pr_{S \sim D^m}\left(c_D > \overline{\mathsf{Bin}}(\hat{c}_S, \delta P(c))\right) < \delta P(c)$$

Apply union bound: $\Pr(A \text{ or } B) \leq \Pr(A) + \Pr(B)$ repeatedly.

$$\Pr_{S \sim D^m}\left(\exists c: \; c_D > \overline{\mathsf{Bin}}(\hat{c}_S, \delta P(c))\right) < \sum_{c : P(c) \neq 0} \delta P(c) = \delta$$

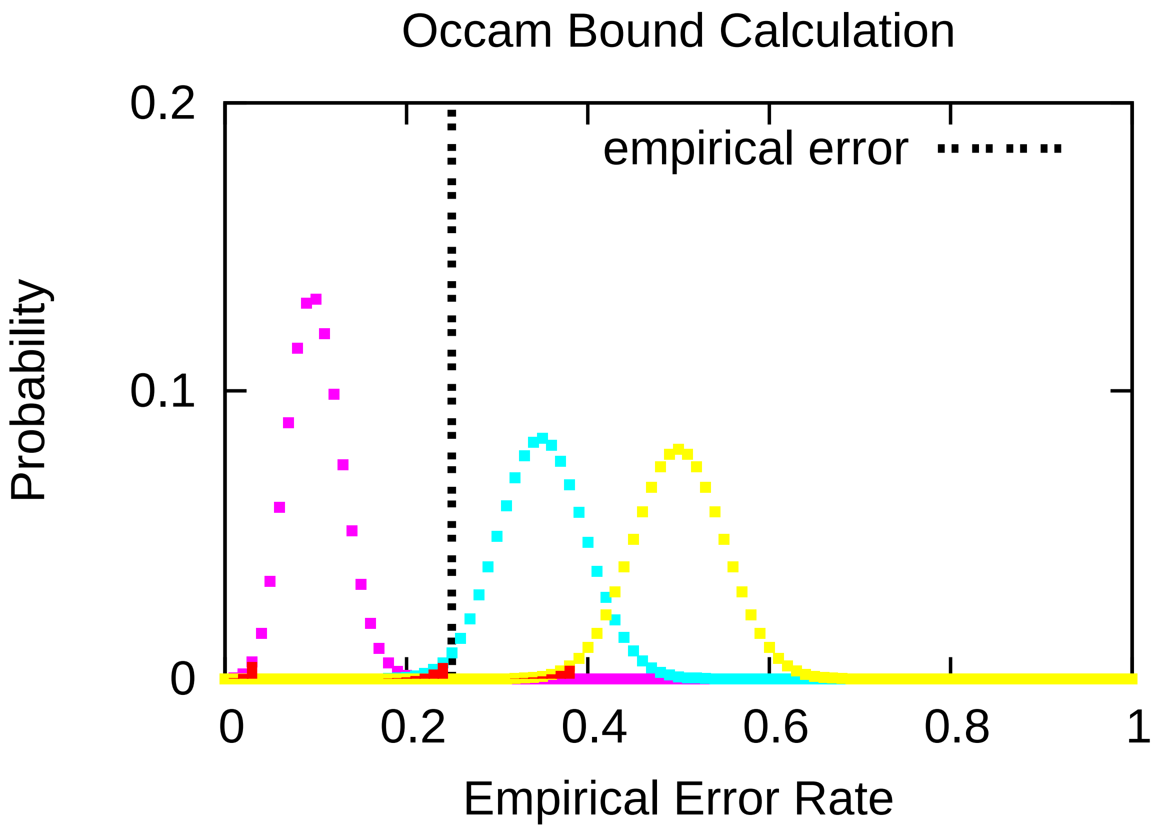
Negate again to get proof.

Next: Graphical proof

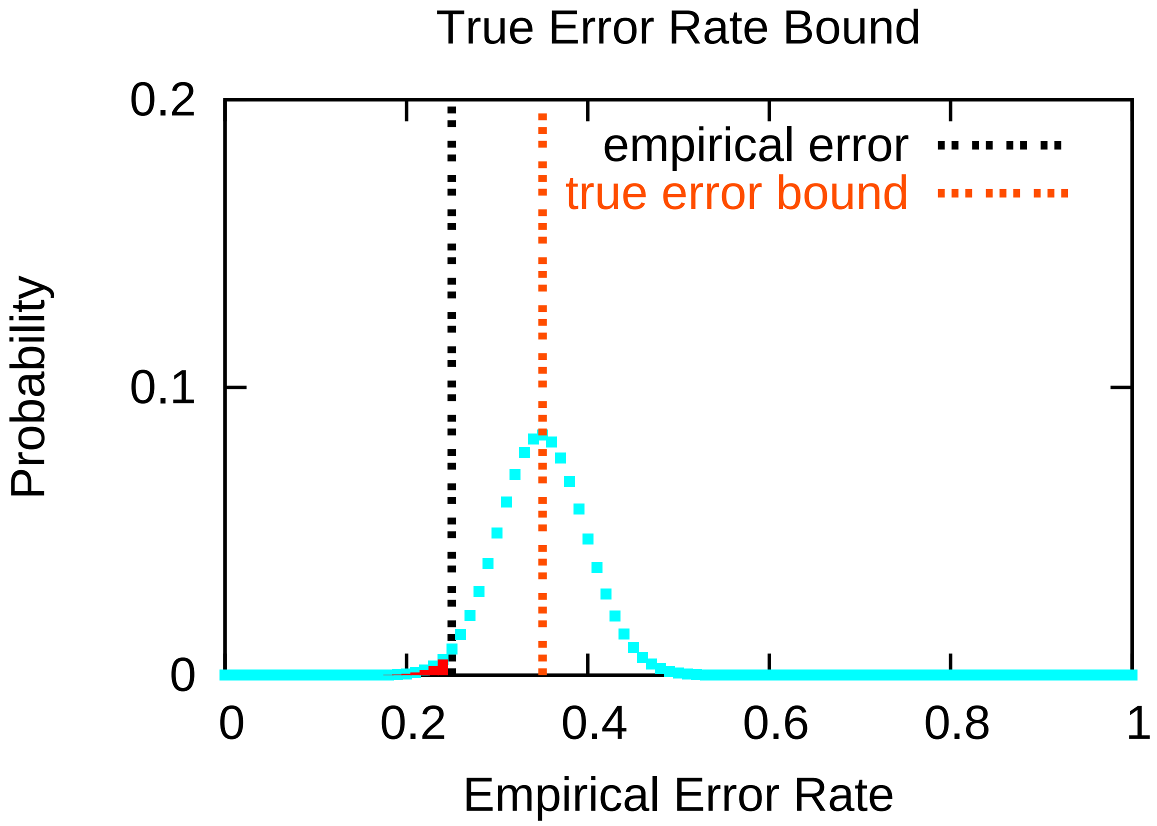


Each classifier is a Binomial with a different size tail cut.

With high probability no error falls in any tail.



The chosen classifier has an unknown true error rate.



Bound = the largest true error rate for which the observation is not in the tail.

Occam's Razor Bound: Example

Suppose $\delta = 0.1$

Suppose $m = 100$

Suppose $P(c) = 0.1$

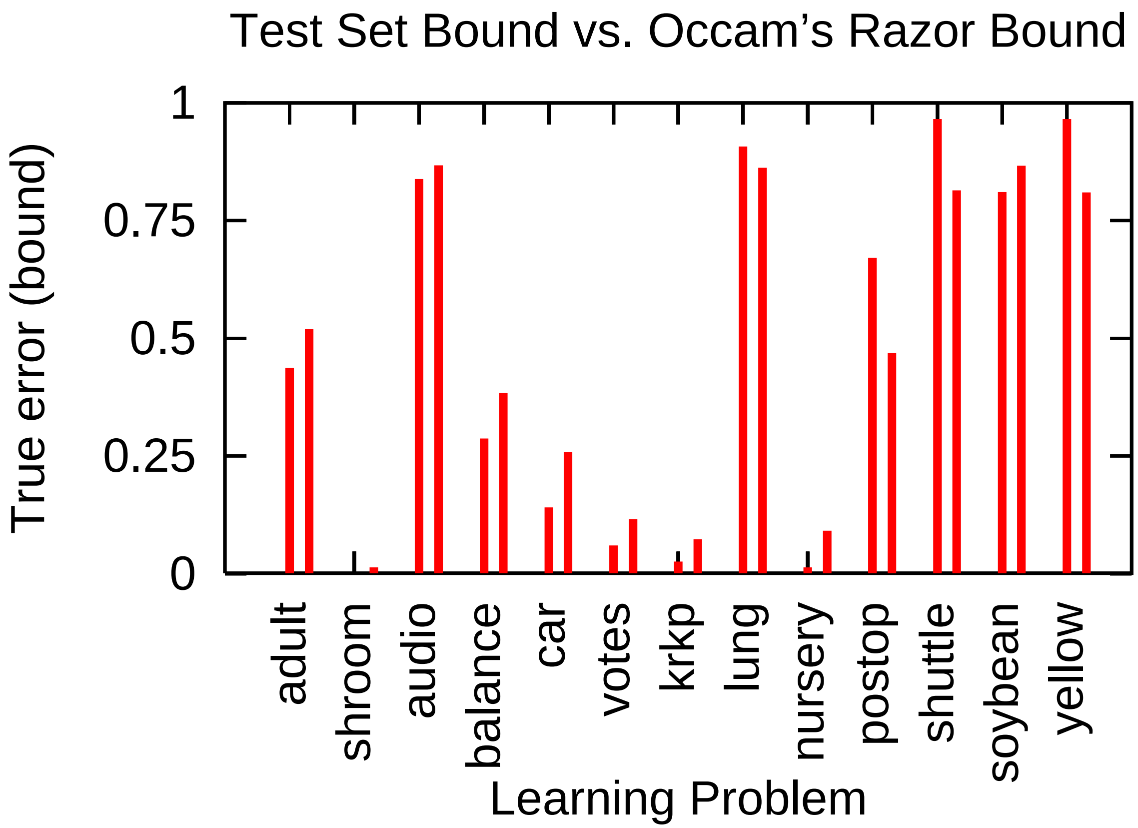
Suppose $\widehat{c}_S = \frac{38}{100}$

Square root Chernoff $\Rightarrow c_D \in [0.22, 0.54]$

Exact calculation $\Rightarrow c_D \in [0, 26, 0.51]$

Occam's Razor Bound Results Decision Trees

- ID3 decision tree + pruning
- probability of failure $= \delta = 0.1$
- Discrete problems from UCI database of Machine Learning problems.
- 100% of data used for training set bounds
- 80%/20% Train/Test split for test set bounds
- Minimal selection bias



Left bar = test set bound, right bar = Occam's Razor Bound

Conclusion

1. Use real confidence intervals to compare classifiers.

2. Test set bound very simple.

3. Train set bounds on the threshold of quantitatively useful.

Code for bound calculation on the class web page.