Basic Sample Complexity

John Langford

http://www.cs.cmu.edu/~jcl/research/tutorial/tutorial.ps
http://www.cs.cmu.edu/~jcl/presentation/tutorial/index.htm (Need MathML. Use mozilla.)

GOTO 1

Help

Goals

1. How can we measure learning?
2. How can we design training algorithms?

GOTO
2

Help

Goals

1. Sample Complexity Model
2. Test Set Bound
3. Application: Learning Measurement
4. Train Set Bound
5. Application: viability and learning algorithm design

GOTO
3

Help

Model

- $X \equiv$ Input space to Classifier
- $Y \equiv \{0,\ 1\} \equiv$ Output space of Classifier
- $D \equiv$ Distribution over $X \times Y$
- $S \equiv$ A set of examples drawn i.i.d. from D.
- $m \equiv |S| \equiv$ the number of examples.
- $c : X \rightarrow Y \equiv$ a classifier.

GOTO
4

Help

Model Notes

Independence is a fundamental assumption.

Model like information theory: $D \sim X \times Y$ rather than $D \sim X$.

GOTO
5

Help

Example

Goal: Predict Rain or Not Rain given sensor data

X = Barometric Pressure, Cloudy/Not Cloudy
Y = Rain or Not Rain
D = Distribution over weather
S = 100 examples i.i.d from D
$c : X \rightarrow Y$ function predicting rain/not rain

GOTO
6

Help

Derived Quantities

True error = $e_D(c) \equiv \Pr_D(c(x) \neq y)$
(= Generalization error)

Empirical error = $\widehat{e}_S(c) \equiv \Pr_{U(S)}(c(x) \neq y)$
(= $\frac{1}{m}\sum_{i=1}^{m} I(c(x_i) \neq y_i)$ = observed error)

GOTO
7

Help

Example (continued)

Suppose: $\forall\ x \quad c(x) =$ Not Rain

Empirical error = $\hat{e}_S(c) = \frac{38}{100}$

True error unknown, maybe 0.5.

GOTO 8

Help

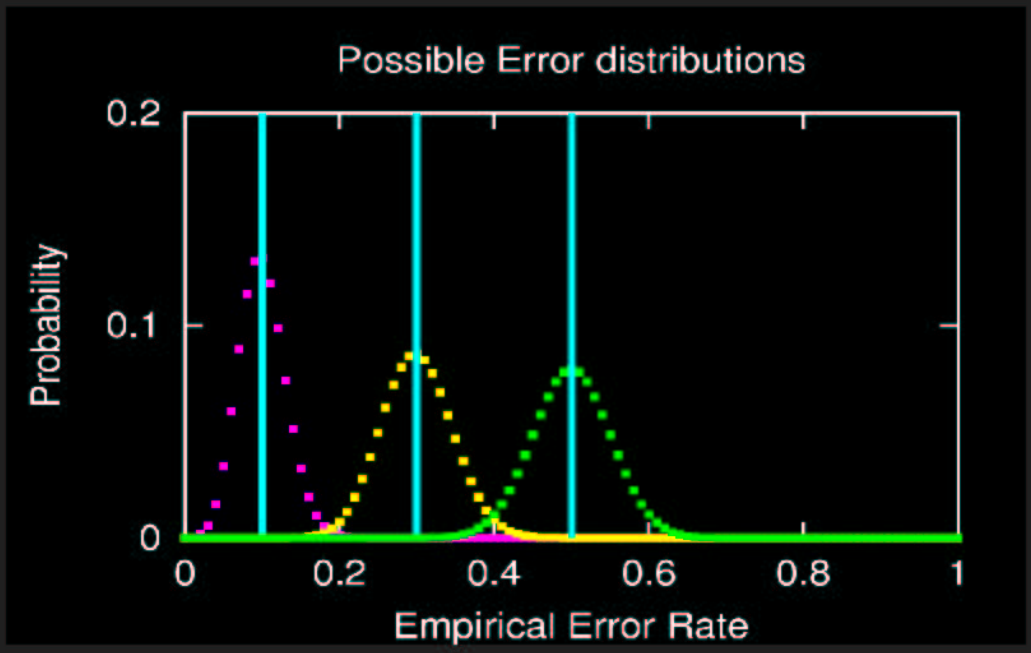
Goals

1. Sample Complexity Model
2. Test Set Bound
3. Application: Learning Measurement
4. Train Set Bound
5. Application: viability and learning algorithm design

GOTO 9

Help

What is the distribution of $\hat{e}_S(c)$?



GOTO
10

Help

Test Set Bound

$\mathrm{Bin}\left(m, \frac{k}{m}, p\right) = \Pr_{X_1, \ldots, X_m}\left(\sum_{i=1}^{m} X_i \leq k \middle| p\right)$
(= Binomial cumulative distribution)

$\overline{\mathrm{Bin}}\left(\frac{k}{m}, \delta\right) = \max_p \{p : \quad \mathrm{Bin}\left(m, \hat{e}_S(c), p\right) = \delta\}$

GOTO
11

Help

Theorem (Test Set Bound):

$$\Pr_{S \sim D^m}\left(e_D(c) \geq \overline{\mathrm{Bin}}\left(\hat{e}_S(c), \delta\right)\right) \leq \delta$$

(note randomization)

Test Set Notes

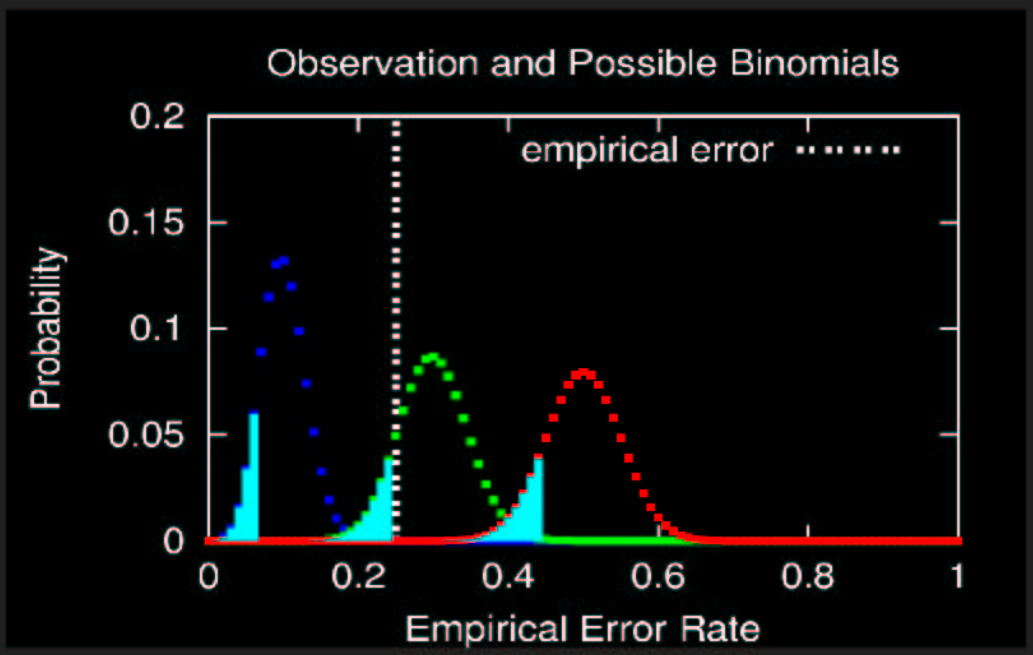
"Perfectly" tight: there exist true error rates which achieve the bound.

Test Set lower bound of the same form.

GOTO
12

Help

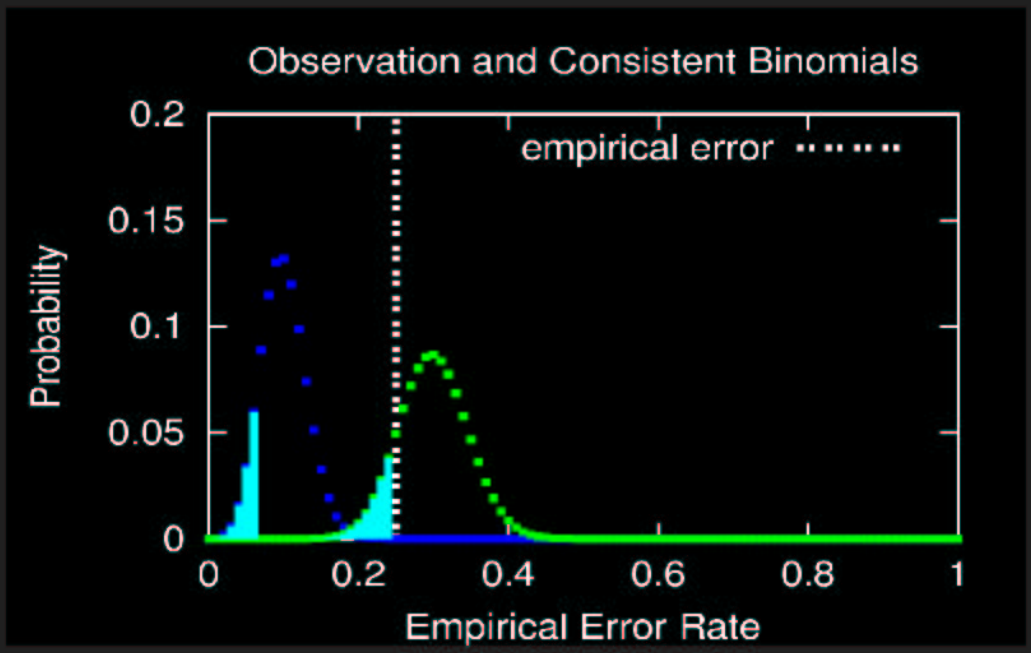
Proof (in pictures)



GOTO
13

Help

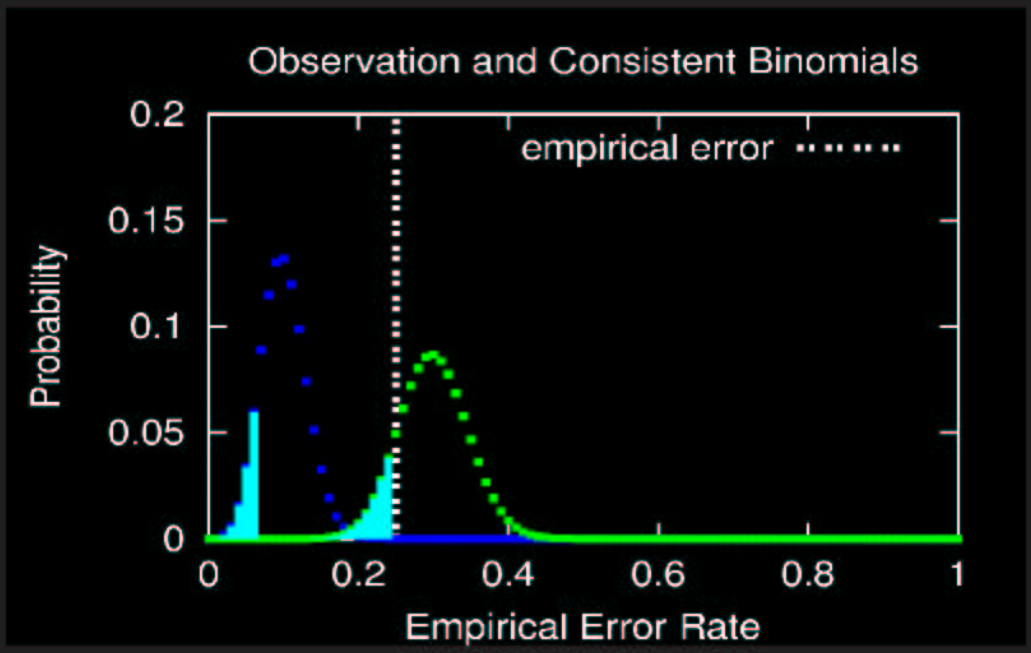
Proof (in pictures)



GOTO
14

Help

Proof (in pictures)



GOTO
15

Help

Test Set Bound Approximations

Corollary: (Realizable Test Set Bound)

$$\Pr_{S \sim D^m}\left(\hat{e}_S(c) = 0 \,\middle|\, e_D(c) \geq \frac{\ln \frac{1}{\delta}}{m}\right) \leq \delta$$

Proof: specialization

Corollary: (Agnostic Test Set Bound)

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

Proof: Hoeffding approximation to Binomial tail bound.

GOTO
16

Help

Goals

1. Sample Complexity Model
2. Test Set Bound
3. Application: Learning Measurement
4. Train Set Bound
5. Application: viability and learning algorithm design

GOTO 17

Help

Test Set Use

Primary use: verification of succesful learning

GOTO
18

Help

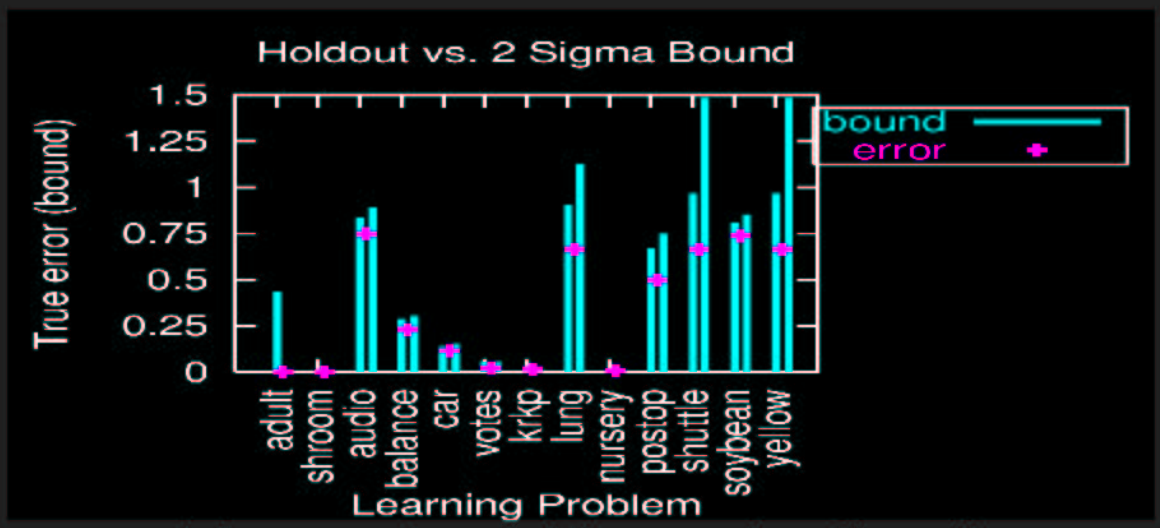
The Sigma Approach

1. $\widehat{\mu} \leftarrow \widehat{e}_{S_{\text{textrm test}}}(h) = \frac{1}{m}\sum_{i=1}^{m} I(h(x_i) \neq y_i)$
2. $\widehat{\sigma}^2 = \frac{1}{m-1}\sum_{i=1}^{m} (I(h(x_i) \neq y_i) - \widehat{\mu})^2$
3. Pretend data is drawn from $N(\widehat{\mu}, \widehat{\sigma})$ gaussian, calculate confidence intervals.

GOTO
19

Help

Test Set vs. Sigma



left = test set bound, right = two sigma "bound"

Learning = seeking "bad" bias.

GOTO
20

Help

Test Set vs. Sigma Notes

Sigma approach (and related) = relic of noncomputerized days.

1. Test set approach *never* overoptimistic.
2. Test Set approch works well for all true error rates.

Bound program available at:
http://www.cs.cmu.edu/~jcl/programs/bound/bound.html

GOTO
21

Help

Example (continued)

Suppose we have:
$m = 100$ test examples
$\hat{e}_S(c) = \frac{38}{100}$
$\delta = 0.1$.

Hoeffding approximation:

$$e_D(c) \in [0.26,\ 0.50]$$

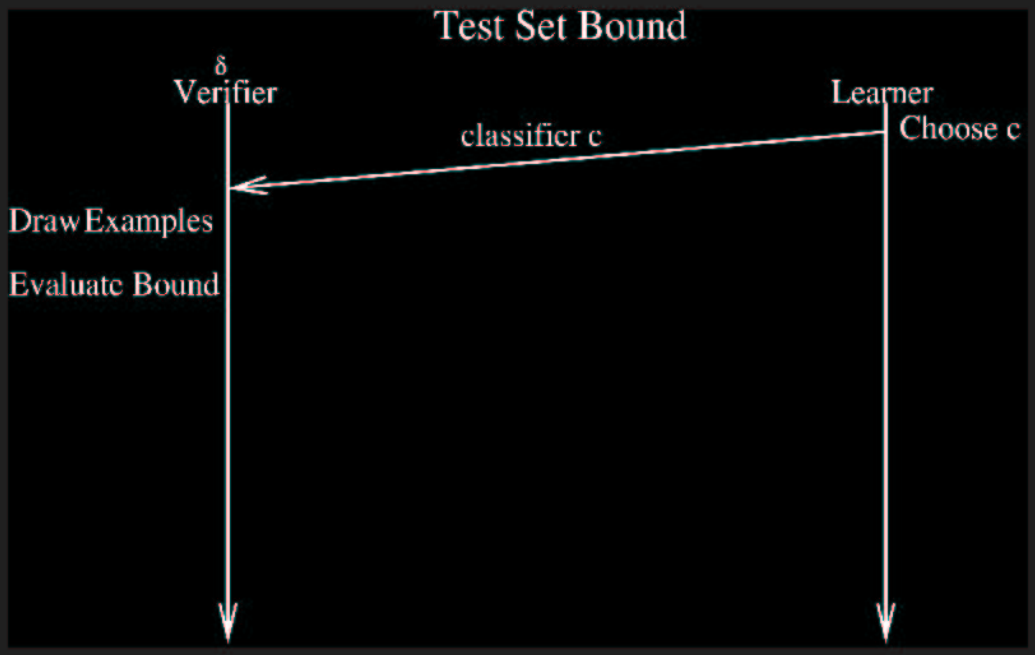
Exact calculation:

$$e_D(c) \in [0.30,\ 0.47]$$

GOTO
22

Help

Test Set Protocol



GOTO
23

Help

The state of the art (test sets)

- Test set bound very well understood.
- Cross validation techniques somewhat understood.
  - Understood for Nearest Neighbor.
  - Weak results for arbitrary algorithms.
- Many others not understood.

GOTO
24

Help

Goals

1. Sample Complexity Model
2. Test Set Bound
3. Application: Learning Measurement
4. Train Set Bound
5. Application: viability and learning algorithm design

GOTO
25

Help

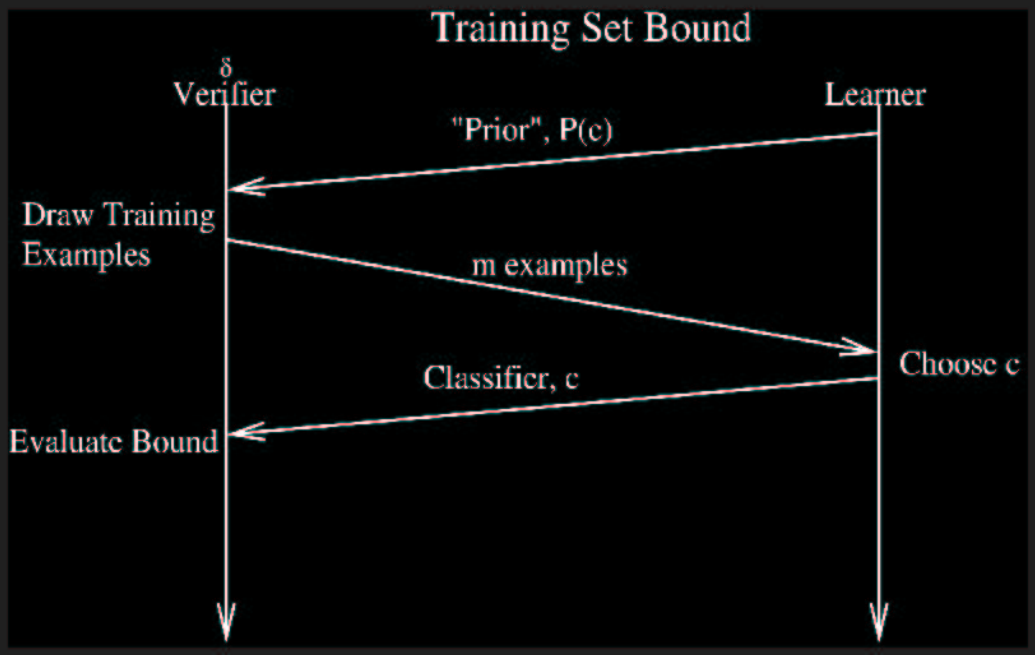
Why Train set bounds?

1. Sometimes an extra few examples are *critical.*
2. Train set bounds can be used in the learning algorithm.
3. Train set bounds teach us about ways to learn.

GOTO
26

Help

Train Set Protocol



GOTO
27

Help

Train Set Bound

Theorem: (Occam's Razor Bound) $\forall\ p(c)$

$$\Pr_{S \sim D^m}\left(\exists\ c:\ \ e_D(c) \geq \overline{\mathrm{Bin}}\left(\hat{e}_S(c), \delta\, p(c)\right)\right) \leq \delta$$

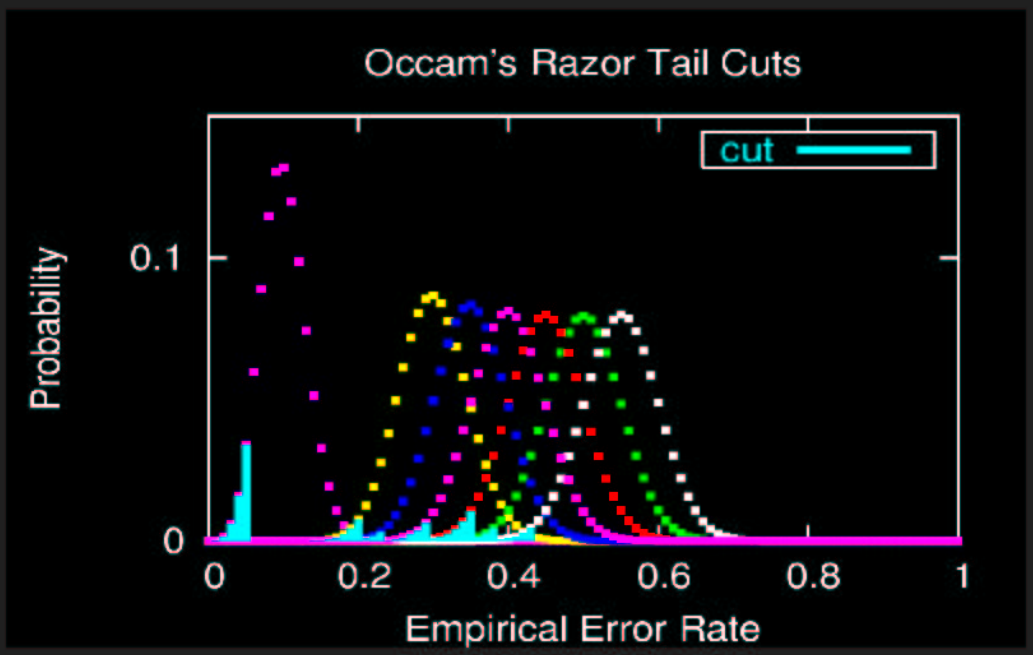
Corollary: $\forall\ p(c)$

$$\Pr_{S \sim D^m}\left(\exists\ c:\ \ e_D(c) \geq \hat{e}_S(c) + \sqrt{\frac{\ln \frac{1}{p(c)} + \ln \frac{1}{\delta}}{2m}}\right) \leq \delta$$

GOTO 28

Help

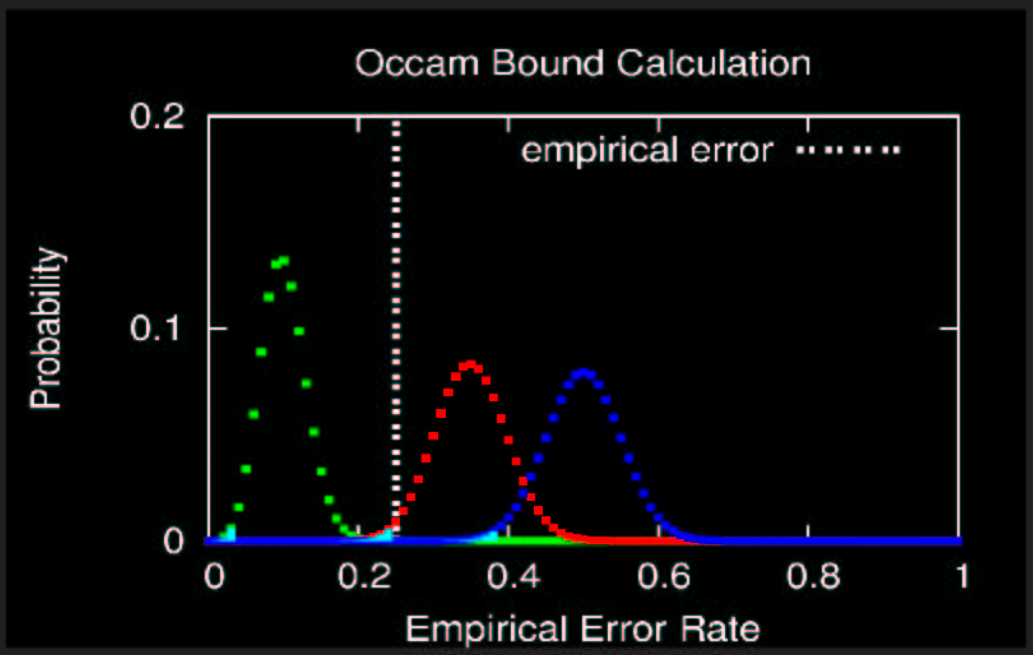
Proof (in pictures)



GOTO
29

Help

Proof (in pictures)



GOTO
30

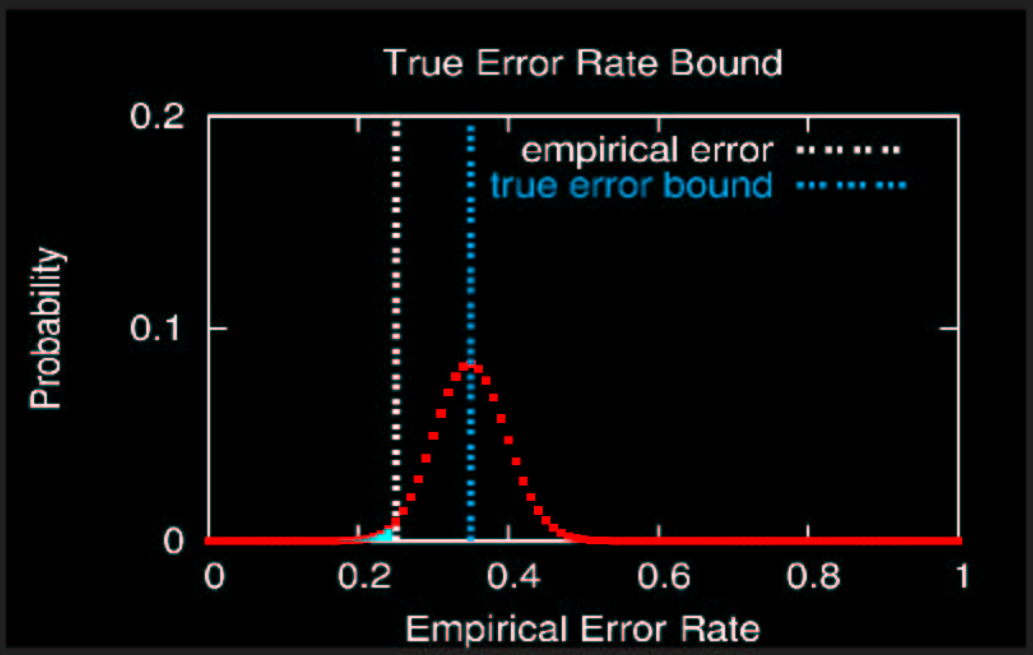
Help

Proof (in pictures)

GOTO
31

Help

Proof (in pictures)



GOTO
32

Help

Goals

1. Sample Complexity Model
2. Test Set Bound
3. Application: Learning Measurement
4. Train Set Bound
5. Application: viability and learning algorithm design

GOTO
33

Help

Train Set setup

- Discrete classification problems.
- Decision tree algorithm.
- d = description length = description of data-dependent choices made to construct tree.
- "prior" = 2^{-d}

GOTO
34

Help

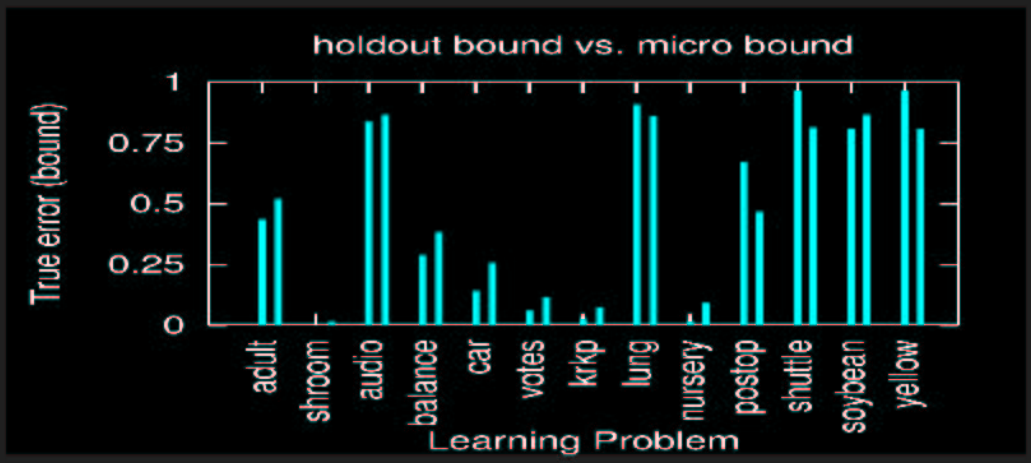
Train Set Bound Uses

- Verification of learning
- Decision tree pruning criteria
- Method for learning

GOTO
35

Help

Train Set Verification



- Left = Test set bound in 80/20 train/test split
- Right = Train set bound in 100/0 train/test split

GOTO
36
Help

Example

Suppose $p(c) = 0.1$ with $m = 100$ train examples and error rate $\widehat{e}_S(c) = \frac{38}{100}$.

Hoeffding approximation:

$$e_D(c) \in [0.22,\ 0.54]$$

Exact calculation:

$$e_D(c) \in [0.26,\ 0.51]$$

GOTO
37

Help

Train Set Induction

Bound $\Rightarrow$ learning algorithm (= Optimize bound)

Note learning algorithm is often *bad*.

1. Many bounds too loose to justify use. Let $f(k/m) > 0$

$$\Pr_{S \sim D^m}\left[e_D(c) \leq 1 + f\left(\hat{e}_S(c)\right)\right] = 1$$

2. Learning algorithm = minimize worst case.
   1. Optimize for best case?
   2. Optimized for average case?

GOTO 38

Help

The State of the Art (Train Sets)

Much work

- VC bounds (reinterpreted...)
- PAC bounds (reinterpreted...)
- PAC-Bayes bounds
- Shell bounds
- Combined Train and Test Bounds

GOTO
39

Help

Conclusion

1. Test set bound = "right" way to verify learning.
2. Train set bound:
    1. more developmental
    2. Can be useful now
    3. suggest learning algorithms

http://www.cs.cmu.edu/~jcl/research/tutorial/tutorial.ps

GOTO
40

Help