Lessons From Statistical Learning Theory for Benchmark Design

John Langford

Toyota Technological Institute - Chicago

A Prototypical Result from Learning Theory

$D =$ distribution on $X \times \{0,1\}$

$S \sim D^m$ be m i.i.d. draws from D

$c : X \to \{0,1\}$ be a classifier

$\hat{c}_S = \Pr_{(x,y)\sim S}(c(x) \neq y) = \frac{1}{|S|} \sum_{(x,y)\in S} I(c(x) \neq y)$

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

Theorem: For *all* D, for *all* c, for *all* $\delta > 0$:

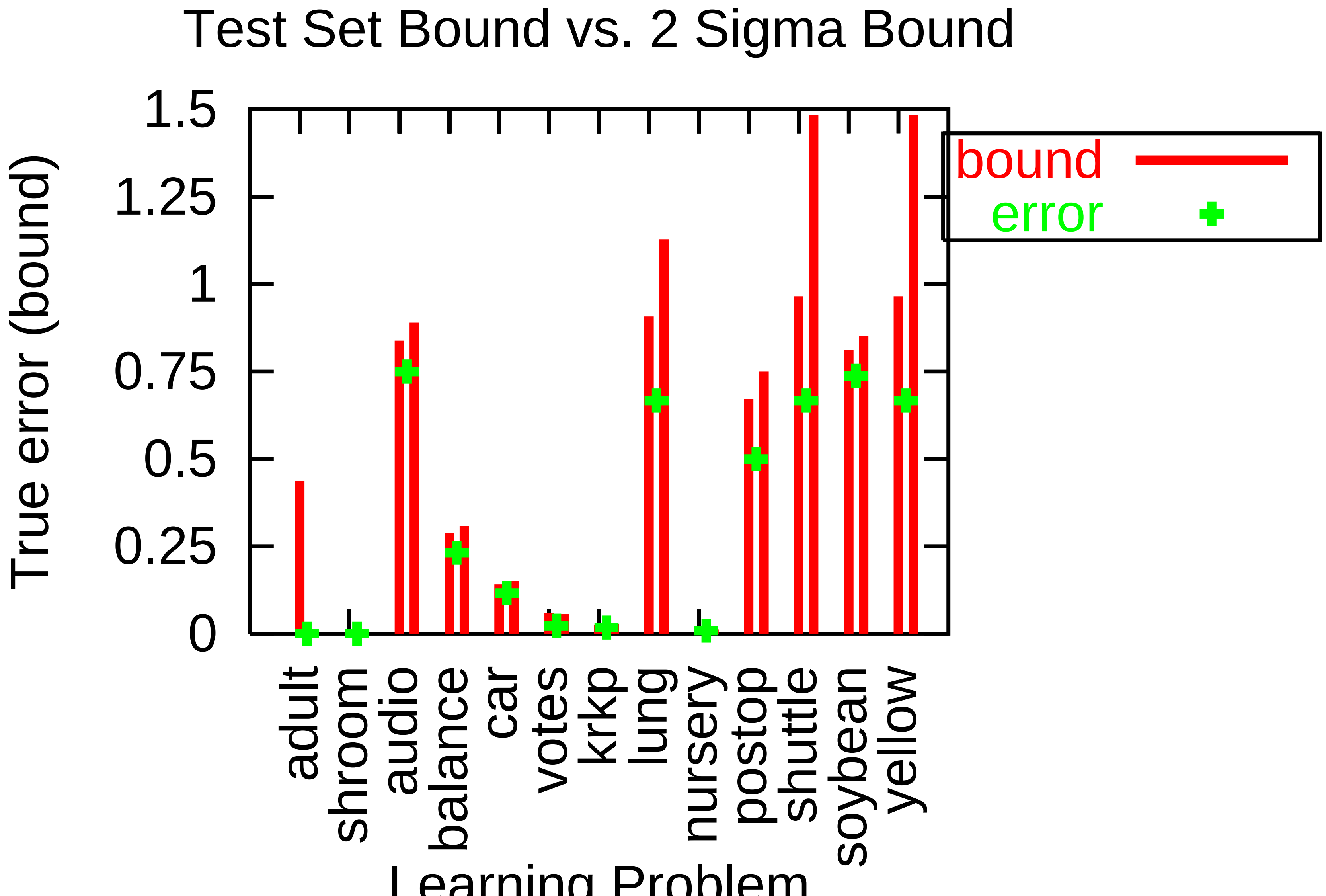
$$\Pr_{S\sim D^m}\left(c_D \leq \hat{c}_S + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}\right) \geq 1 - \delta$$

Note:

1. Very General (few assumptions => many applications)

2. Directly Applicable (demo)

Hidden here: even tighter results hold


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

Outline

1. Prediction Domains and Loss functions
   (a) Classification (Predict a bit)
   (b) Regression (Predict a real)
   (c) Density Estimation (Predict a measure)
2. Prediction Settings
3. Assumption Failure

Regression

$D =$ distribution on $X \times [0,1]$

$S \sim D^m$ be m i.i.d. draws from D

$r \colon X \rightarrow [0,1]$ be a regressor

$\hat{r}_S = E_{(x,y)\sim S}(r(x)-y)^2 = \frac{1}{|S|}\sum_{(x,y)\in S}(r(x)-y)^2$

$r_D = E_{(x,y)\sim D}(r(x)-y)^2$

Theorem: For *all* D, for *all* r, for *all* $\delta > 0$:

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

Regression Notes

1. Sometimes D on $(-\infty, \infty)$ $\Rightarrow$ Theorem fails!

2. Sometimes assume $D(y|x) = f(x) +$ normal noise $\Rightarrow$ similar theorem.

3. Hidden detail: Very tight bounds are harder than for classification.

Density Estimation

D on domain X

$p(x)$ = probability or probability density on x

$p_D = E_{x \sim D} \ln \frac{1}{p(x)}$

1. Impossible to make theorem statement given above.

2. Assume D normal $\Rightarrow$ theorem statement.

3. Bounded loss function$\Rightarrow$ theorem statement.

Outline

1. Prediction Domains and Loss functions
   (a) Classification (Cleanest analysis)
   (b) Regression (Reasonable analysis)
   (c) Density Estimation (Tricky)
2. Prediction Settings
3. Assumption Failure

Outline

1. Prediction Domains and Loss functions

2. Prediction Settings

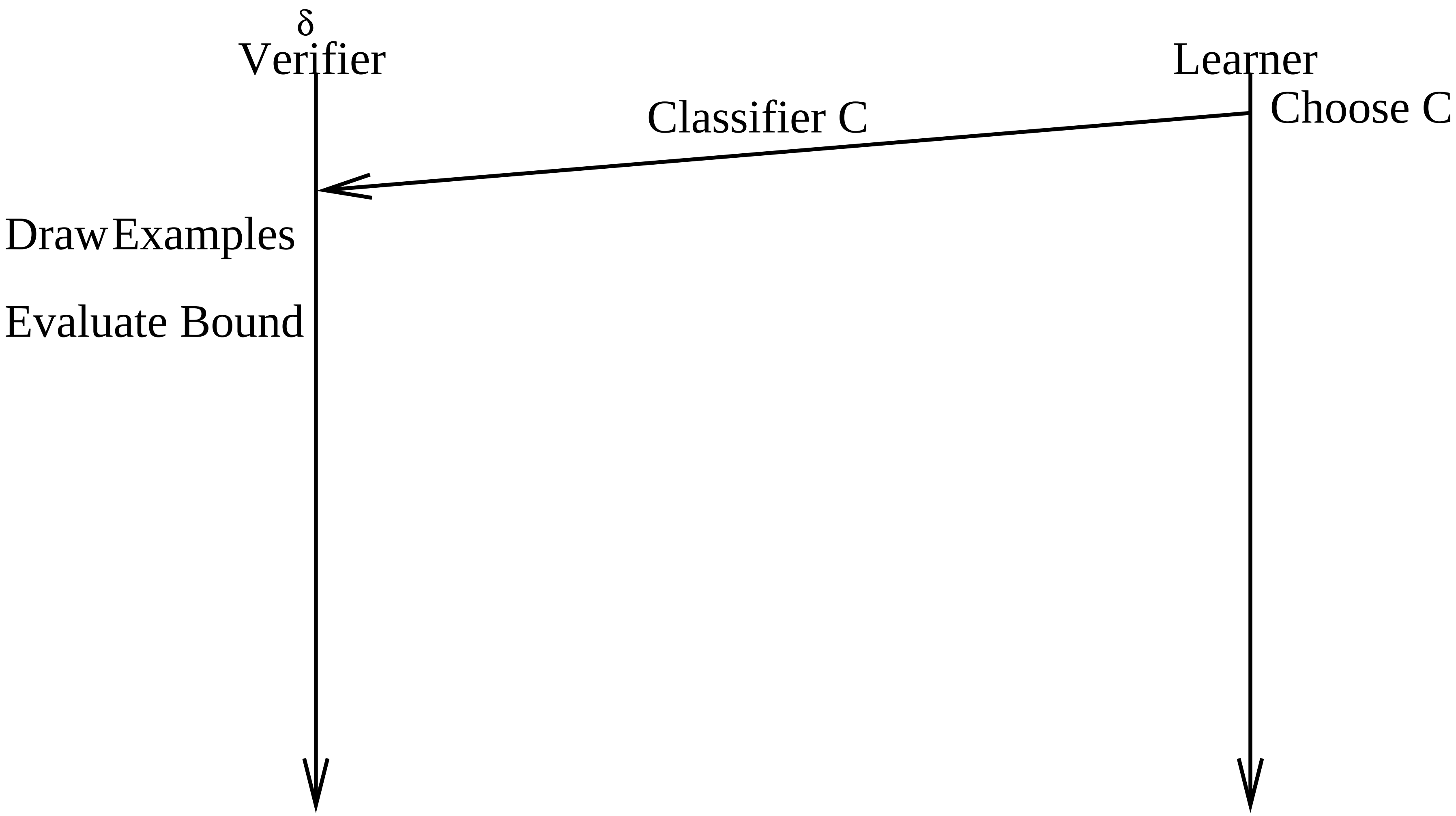
   (a) Batch: Train classifier, then evaluate on test set.

   (b) Online: Interactive train and test.

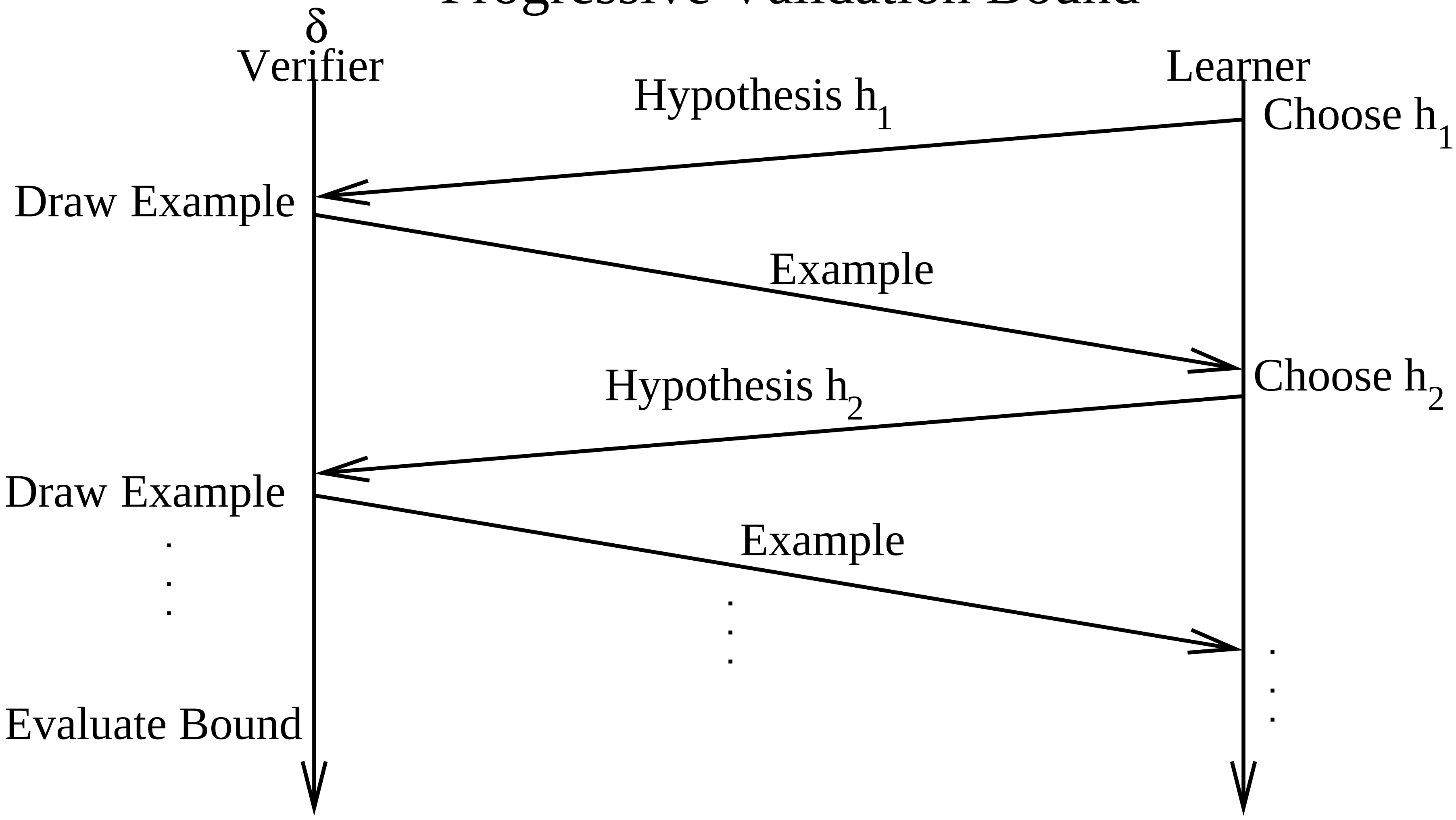
   (c) Pure Train: Train and test on the same sample set.

3. Assumption Failure

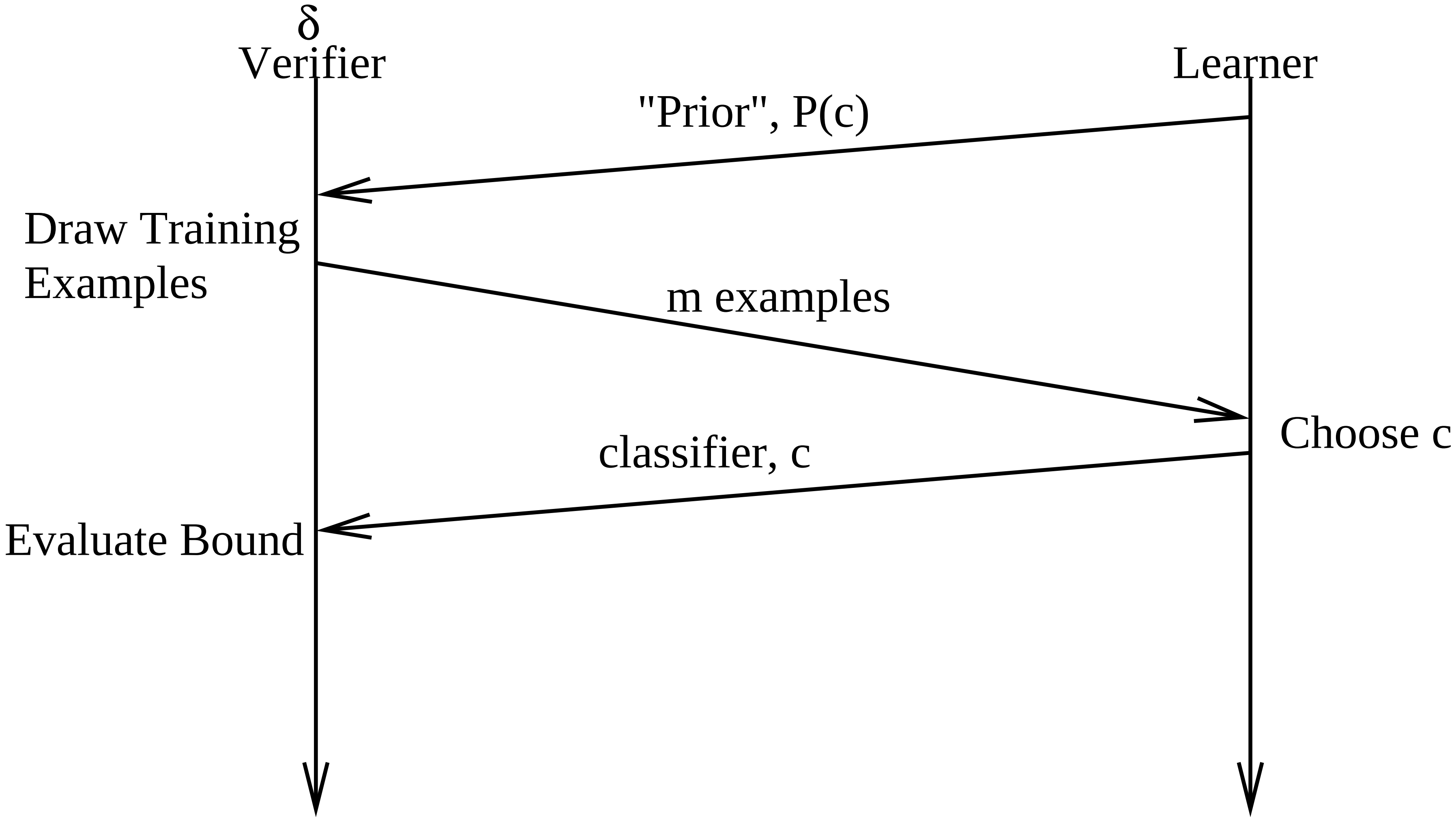
Test Set Bound



Progressive Validation Bound



Occam's Razor Bound Protocol



Outline

1. Prediction Domains and Loss functions

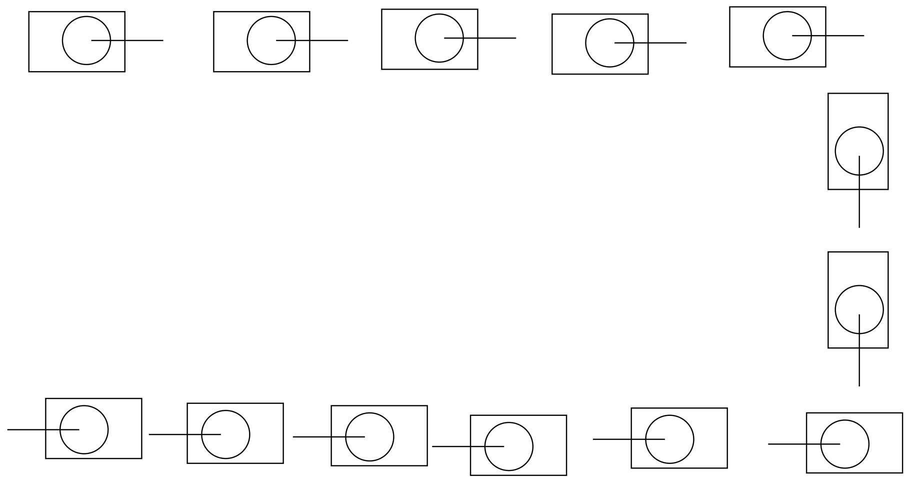
2. Prediction Settings

3. Assumption Failure: What do we do now? Design around it.

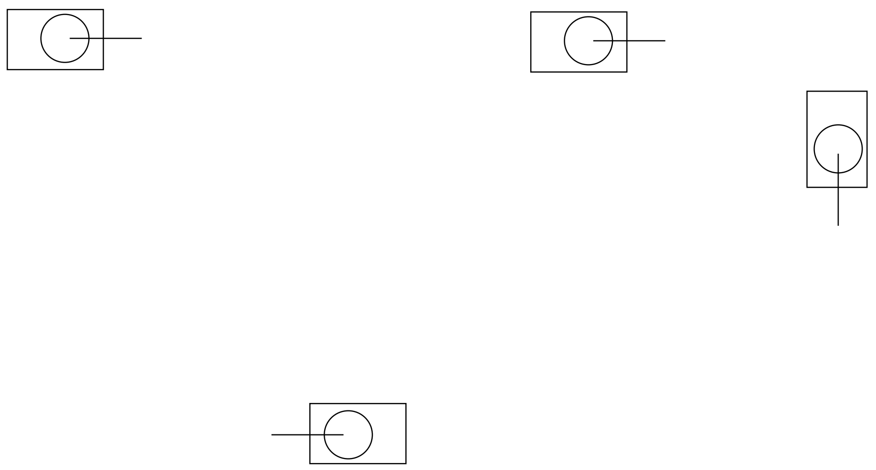
   (a) Correlated samples: "purify" by subsampling

   (b) Drifting distribution: Get lots of data so drift = correlation

Purification, before



Purification, after



Final Note: Classification is more adaptable than it looks