Simple Sample Complexity for Machine Learning

NYU, 2007

John Langford

Full Tutorial writeup at:

hunch.net/~jl/projects/prediction_bounds/tutorial/langford05a.ps

Quiz

For a dataset with 1000 binary features, how many examples are sufficient to learn:

1. A 100 node decision tree?

2. A 100 node neural network?

3. A 100 support vector machine with margin 0.1?

Outline

1. The Basic Model
2. Occam's Razor Bound
3. PAC-Bayes Bound
4. ... for margins

Model: Definitions

X = input space

$Y = \{-1, 1\}$ = 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:

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

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

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

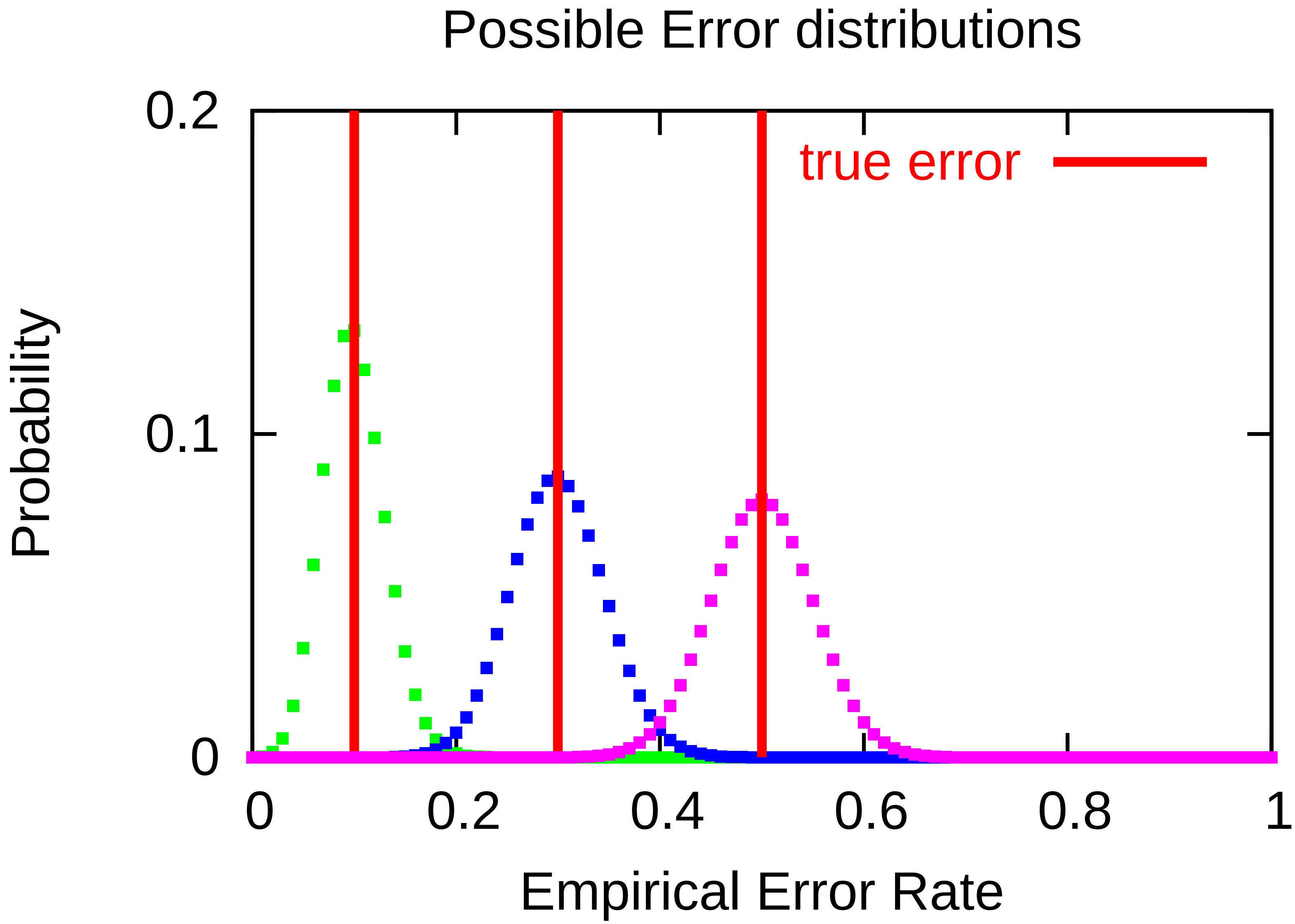
Model: Basic Observations

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

A: A Binomial.

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

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


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

Model: basic quantities

We use the cumulative:

$$\begin{aligned}\mathsf{Bin}\left(m,k,c_D\right) &= \mathsf{Pr}_{S\sim D^m}\left(\hat{c}_S \le k \middle|\, c_D\right)\\ &= \sum_{i=0}^{k} \binom{m}{i} c_D^i (1-c_D)^{m-i}\end{aligned}$$

= 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

$$\overline{\mathsf{Bin}}\,(m,k,\delta) = \max\{p : \mathsf{Bin}\,(m,k,p) \geq \delta\}$$

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

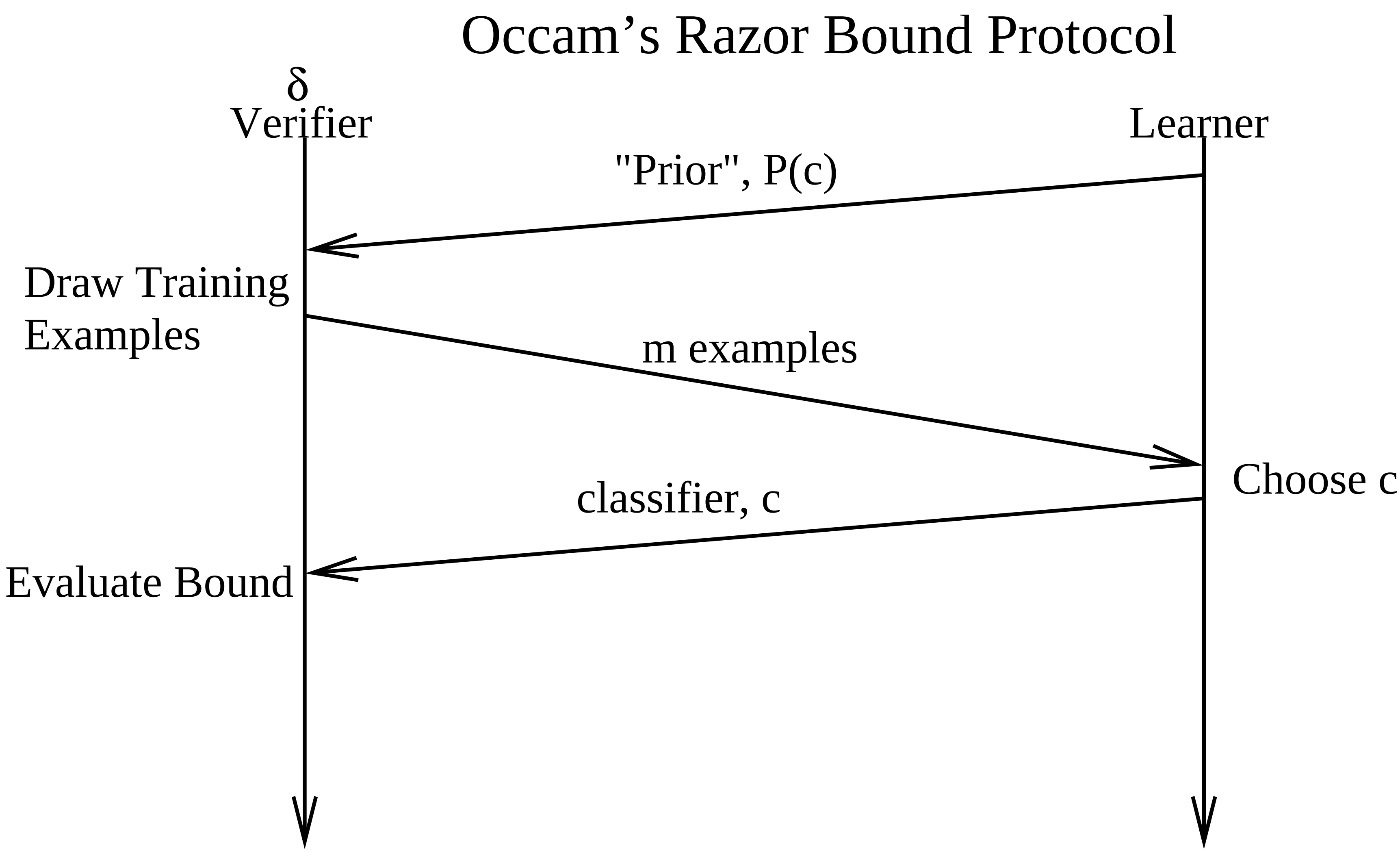
Outline

1. The Basic Model

2. Occam's Razor Bound

3. PAC-Bayes Bound

4. ... for margins


Occam's Razor Bound Protocol
δ
Verifier
Learner
"Prior", P(c)
Draw Training
Examples
m examples
Choose c
classifier, c
Evaluate Bound

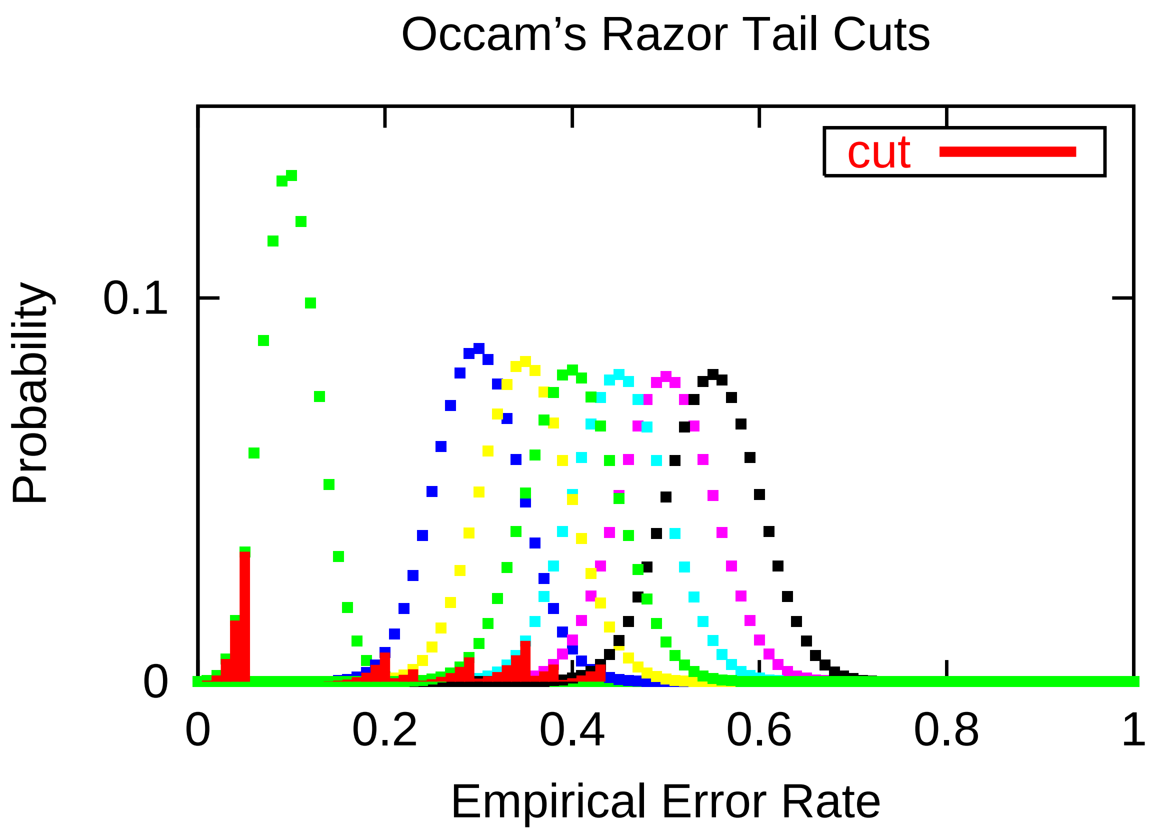
Occam's Razor Bound

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

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

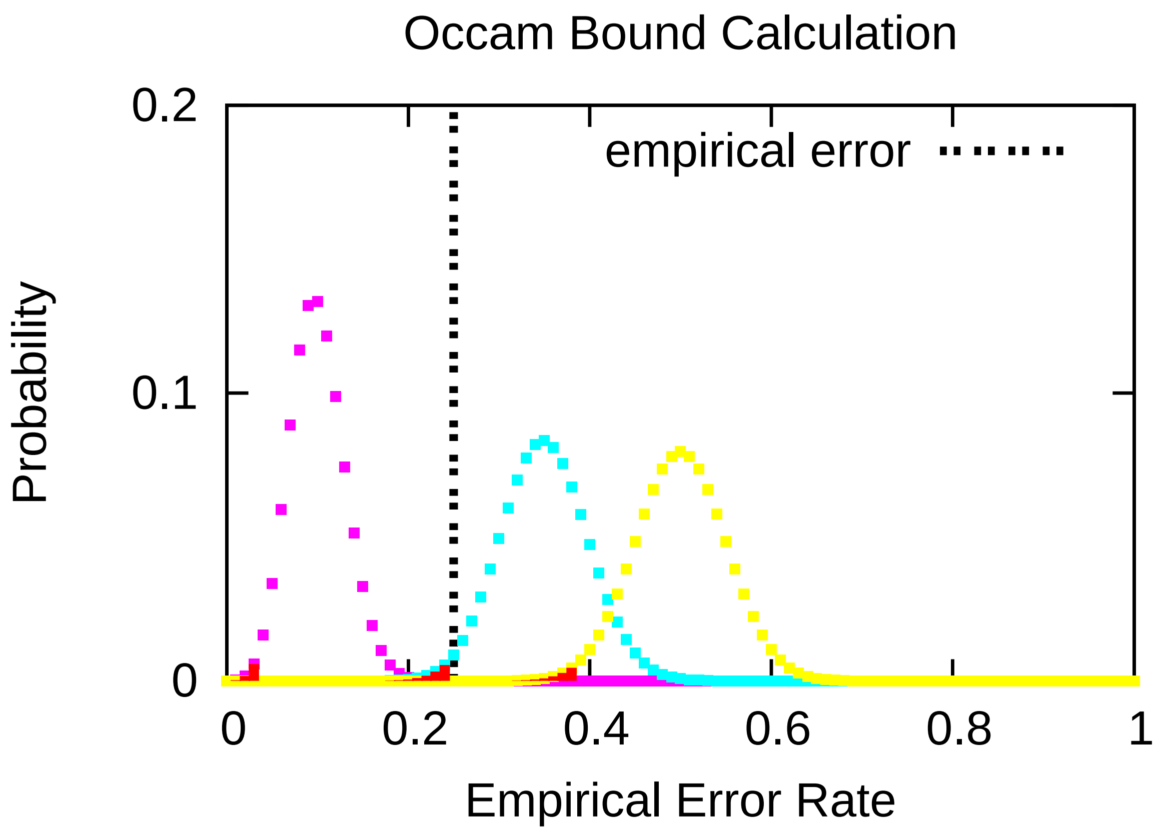
Corollary: For all $P(c)$, for all D, for all $\delta \in (0,1]$:

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

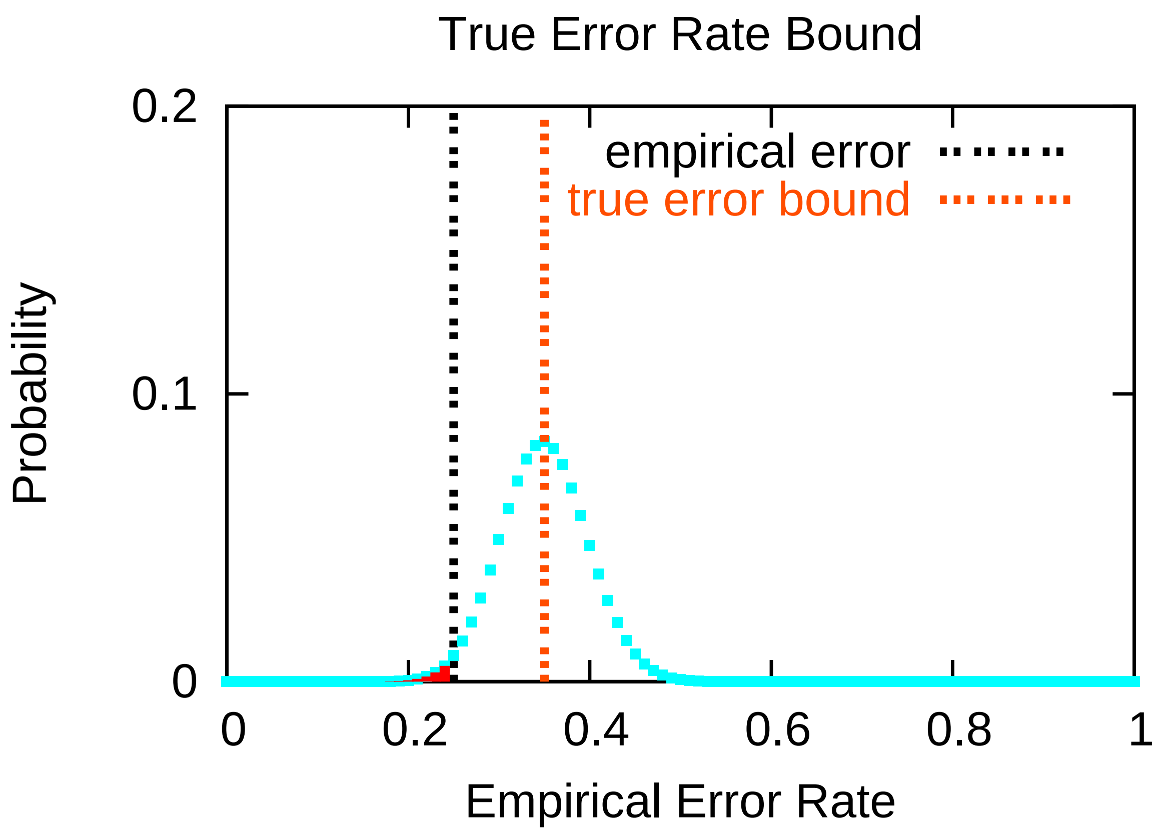


Each classifier is a Binomial with a different size tail cut.

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.

Quiz

For a dataset with 1000 binary features, how many examples are sufficient to learn:

1. A 100 node decision tree?

2. A 100 node neural network?

3. A 100 support vector machine with margin 0.1?

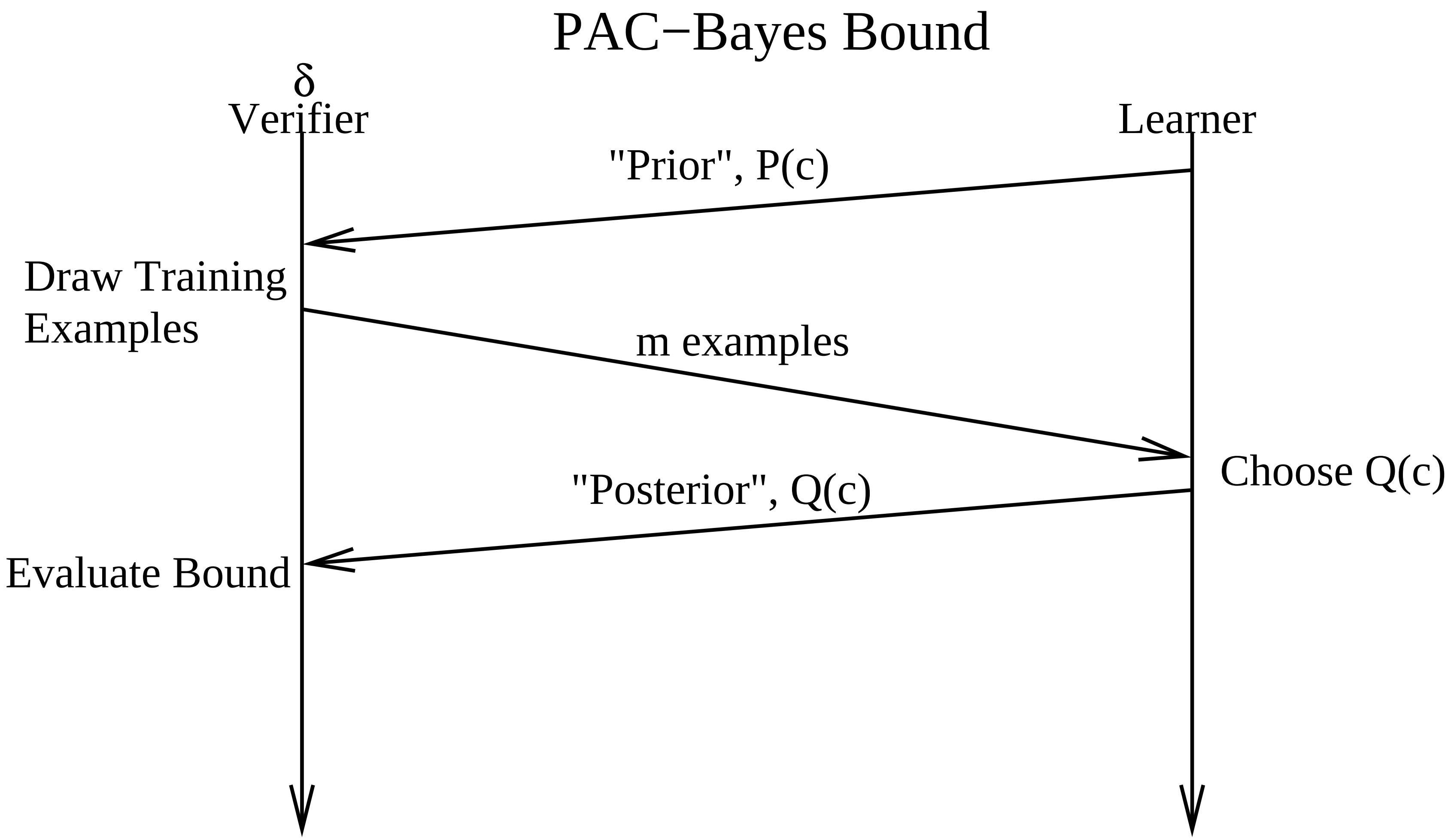
Outline

1. The Basic Model

2. Occam's Razor Bound

3. PAC-Bayes Bound

4. ... for margins


PAC−Bayes Bound
δ
Verifier
Learner
"Prior", P(c)
Draw Training
Examples
m examples
Choose Q(c)
"Posterior", Q(c)
Evaluate Bound

PAC-Bayes Bound: Basic quantities

$Q_D \equiv E_{c \sim Q}[c_D]$ = average true error

$\hat{Q}_S \equiv E_{c \sim Q}\left[\frac{\hat{c}_S}{m}\right]$ = average train error

PAC-Bayes Bound: Theorem

Theorem: (PAC-Bayes Bound) For all "priors" $P(c)$ over the classifiers c, for all D, for all $\delta \in (0, 1]$:

$$\Pr_{S \sim D^m}\left(\forall Q(c):\ \mathrm{KL}\left(\hat{Q}_S || Q_D\right) \leq \frac{\mathrm{KL}(Q||P) + \ln \frac{m+1}{\delta}}{m}\right) \geq 1 - \delta$$

where: $\mathrm{KL}(Q||P) = E_{c \sim Q} \ln \frac{Q(c)}{P(c)}$

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

$$\Pr_{S \sim D^m}\left(\forall Q(c):\ Q_D \leq \hat{Q}_S + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln \frac{m+1}{\delta}}{2m}}\right) \geq 1 - \delta$$

Outline

1. The Basic Model
2. Occam's Razor Bound
3. PAC-Bayes Bound
4. ... for margins

PAC-Bayes Bound: Application

Is the PAC-Bayes bound tight enough to be useful?

Application: true error bounds for Support Vector Machines.

Classifier form:

$$c(x) = \mathrm{sign}\left(\vec{w} \cdot \vec{x}\right)$$

PAC-Bayes Margin bound

$\bar{F}(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ = cumulative distribution of a Gaussian

$Q(\vec{w}, \mu) = N(\mu, 1) \times N(0, 1)^{n-1}$ where first direction parallel to $\vec{w}$

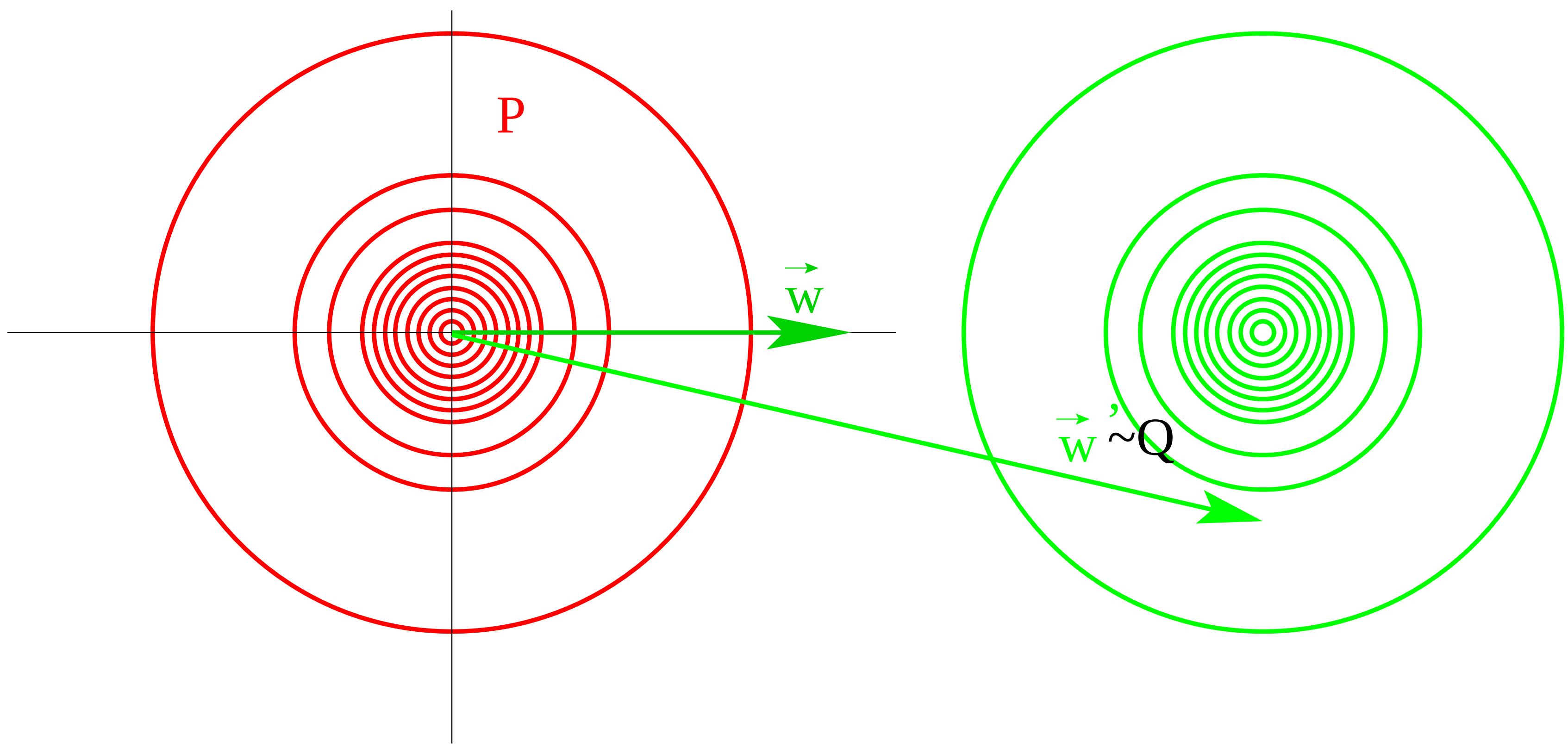
$\gamma(\vec{x}, y) = \frac{y\vec{w} \cdot \vec{x}}{||\vec{w}||||\vec{x}||}$ = normalized margin

$\hat{Q}(\vec{w}, \mu)_S = E_{\vec{x}, y \sim S} \bar{F}\left(\mu\gamma(\vec{x}, y)\right)$ = stochastic error rate

Corollary: (PAC-Bayes Margin Bound) For all distributions D, for all $\delta \in (0, 1]$:

$$\Pr_{S \sim D^m}\left(\forall \vec{w}, \mu > 0 : \ \mathrm{KL}\left(\hat{Q}(\vec{w}, \mu)_S || Q(\vec{w}, \mu)_D\right) \leq \frac{\frac{\mu^2}{2} + \ln \frac{m+1}{\delta}}{m}\right) \geq 1 - \delta$$

PAC-Bayes Margin Bound: Intuition



Isotropic Gaussian prior and posterior

PAC-Bayes Margin Bound: Proof

Start with PAC-Bayes bound:

$$\forall P(c) \quad \Pr_{S \sim D^m}\left(\forall Q(c) :\ \mathrm{KL}\left(\hat{Q}_S || Q_D\right) \leq \frac{\mathrm{KL}(Q||P) + \ln \frac{m+1}{\delta}}{m}\right) \geq 1 - \delta$$

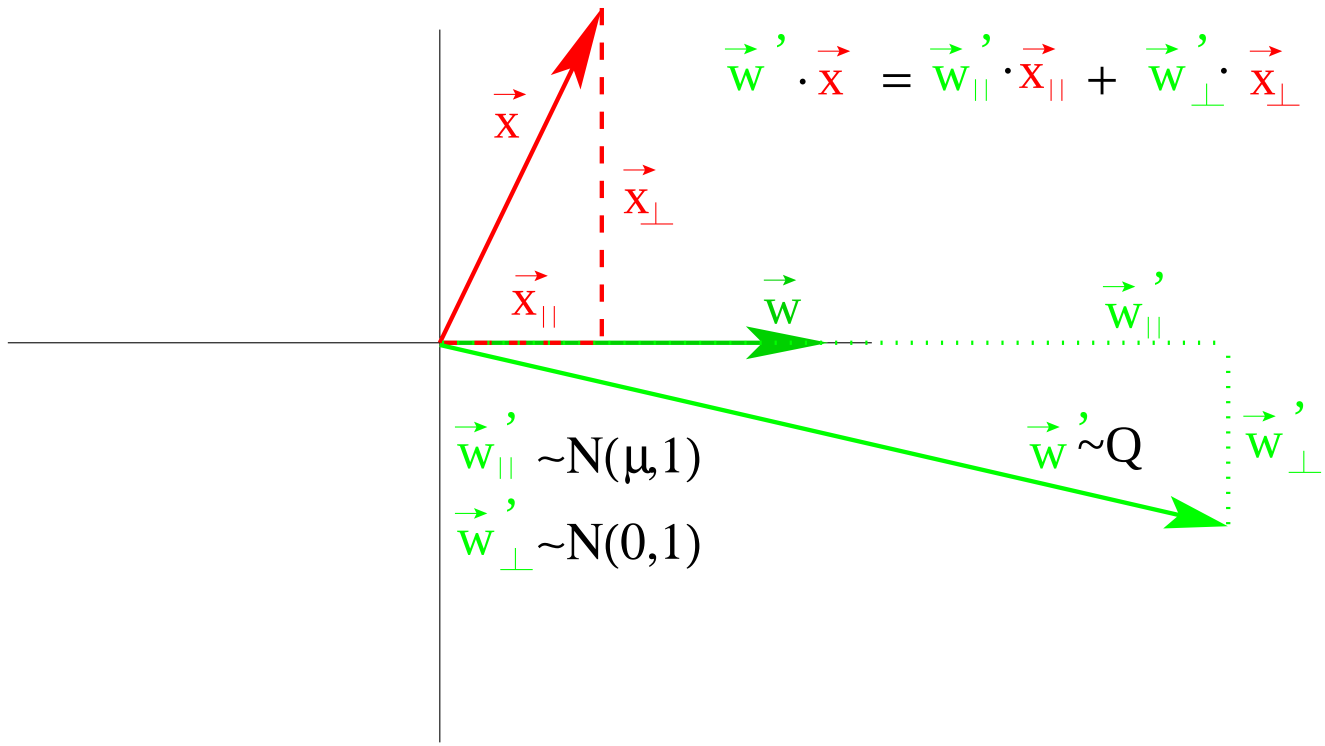
Set $P = N(0,1)^n$

$Q(\vec{w}, \mu) = N(\mu, 1) \times N(0,1)^{n-1}$ with first direction parallal to $\vec{w}$

Gaussian $\Rightarrow$ coordinate system reorientable

$$\Rightarrow \mathrm{KL}(Q||P) = \mathrm{KL}(N(0,1)^{n-1}||N(0,1)^{n-1}) + \mathrm{KL}(N(\mu,1)||N(0,1))$$

$$= \frac{\mu^2}{2}$$



$$\hat{Q}(\vec{w},\mu)_S = E_{\vec{x},y\sim S,\vec{w}'\sim Q(\vec{w},\mu)} I\left(y \neq \mathrm{sign}\left(\vec{w}' \cdot \vec{x}\right)\right)$$

$$= E_{\vec{x},y\sim S} E_{w'_{||}\sim N(\mu,1)} E_{w'_{\perp}\sim N(0,1)} I\left(y(w'_{||}x_{||} + w'_{\perp}x_{\perp}) \leq 0\right)$$

Use properties of Gaussians to finish proof

PAC-Bayes Margin proof: the end

$$= E_{\vec{x},y\sim S} E_{z'\sim N(0,1)} E_{w'_{\perp}\sim N(0,1)} I\left(y\mu \leq -yz' - yw'_{\perp}\frac{x_{\perp}}{x_{||}}\right)$$

The sum of two Gaussians is a Gaussian $\Rightarrow$

$$= E_{\vec{x},y\sim S} E_{v\sim N\left(0,1+\frac{x_{\perp}^2}{x_{||}^2}\right)} I\left(y\mu \leq -yv\right)$$

$$= E_{\vec{x},y\sim S} E_{v\sim N\left(0,\frac{1}{\gamma(\vec{x},y)^2}\right)} I\left(y\mu \leq -yv\right)$$

$$= E_{\vec{x},y\sim S} \bar{F}\left(\mu\gamma(\vec{x},y)\right)$$

$\Rightarrow$ Corollary

PAC-Bayes: Application to SVM

SVM classifier:

$$c(x) = \mathsf{sign}\left(\sum_{i=1}^{m} \alpha_i k(x_i, x)\right)$$

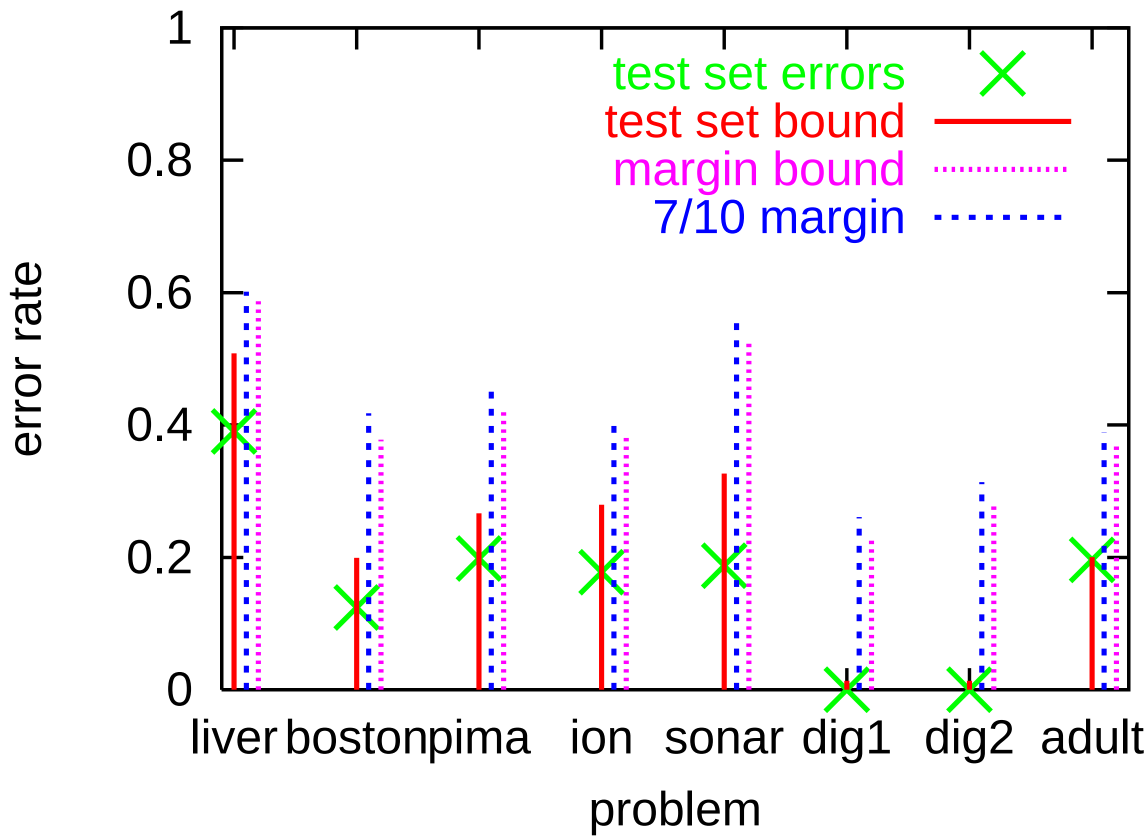
k is a kernel $\Rightarrow \exists \vec{\Phi} : \quad k(x_i, x) = \vec{\Phi}(x_i) \cdot \vec{\Phi}(x)$ so:

$\vec{w} \cdot \vec{x} = \sum_{i=1}^{m} \alpha_i k(x_i, x)$ $\qquad\qquad$ $\vec{w} \cdot \vec{w} = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)$

$$\Rightarrow \gamma(x, y) = \frac{y \sum_{i=1}^{m} \alpha_i k(x_i, x)}{\sqrt{k(x, x) \sum_{i,j=1,1}^{m,m} \alpha_i \alpha_j k(x_i, x_j)}}$$

$\Rightarrow$ Margin bound for kernelized SVM also.

PAC-Bayes Margin Bound Results



My view of things

1. Occam's Razor bound is very useful for high altitude algorithm choosing, but it's loose.

2. PAC-Bayes fixes much of this looseness, with very little tightness loss. PAC-Bayes says you pay only for the bits of precision which actually matter in making a decision.

3. Less elemental bounds (VC, Rademacher, etc...) don't appear to add much to tightness and are often much looser.

4. The bounds can be Martingalized without substantially altering their form. ($\Rightarrow$ works for online learning)