Practical Generalization Bounds

Chicago 2005 Machine Learning Summer School

John Langford

TTI Chicago

Why study prediction theory?

1. Better methods for learning and verifying predictive ability
2. To gain insight into learning.

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.

Outline

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

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?

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.

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:

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

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

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

Model: basic quantities

We use the cumulative:

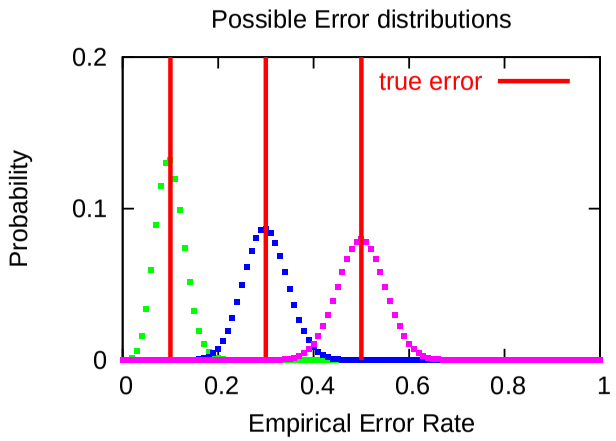
$$\begin{aligned} \text{Bin}\left(m,k,c_D\right) &= \Pr_{S\sim D^m}\left(\hat{c}_S \leq \frac{k}{m}\middle|\, c_D\right) \\ &= \sum_{i=0}^{k}\binom{m}{i} c_D^i (1-c_D)^{m-i} \end{aligned}$$

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

Model: Derived quantities

The thing we want to know:

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



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: Basic Observations

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

A: A Binomial.

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

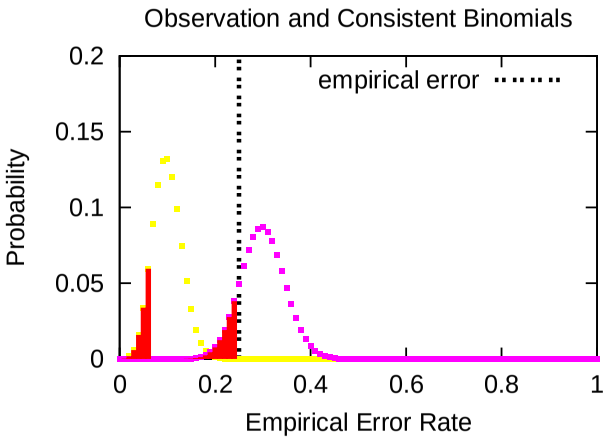
= probability of k heads (errors) in m flips of a coin with bias c_D.

Test Set Bound: Setting

Standard technique:

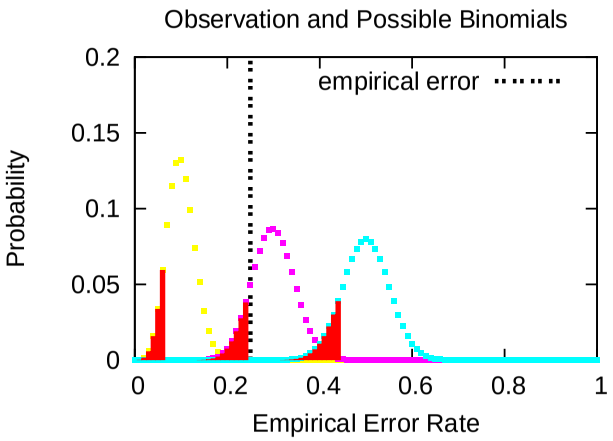
1. Cut the data into train set and test set
2. Train on the train set
3. Test on the test set

What do sample complexity say about this method?



Outline

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



Model: basic quantities

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

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

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

Test Set Bound: Theorem

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

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

World's easiest proof: (by contradiction).

Assume $\text{Bin}(m, k, c_D) \geq \delta$ (which is true with probability $1 - \delta$).

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

What does Test Set Bound mean?

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

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

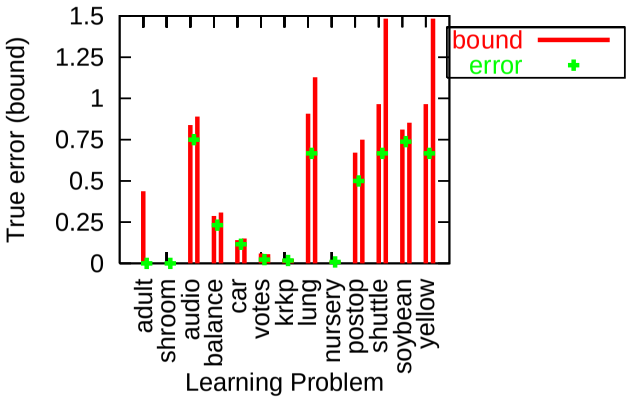
where $\mathrm{KL}(q||p) = q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}$ for $q < p$

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

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

Proof: Use the Chernoff approximation. Full details in the notes.

Test Set Bound vs. 2 Sigma Bound



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

Test Set Bound Comparison: Empirical "confidence" intervals

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

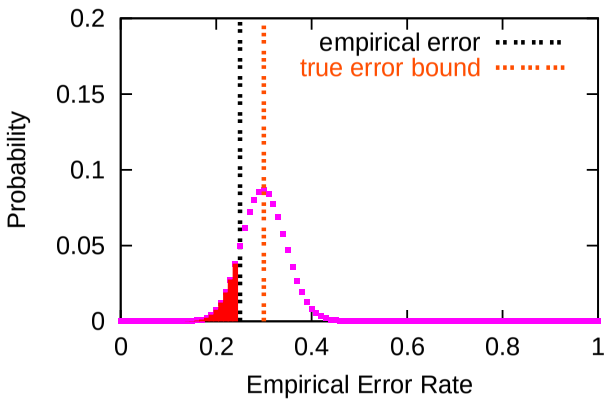
$\mu = \frac{k}{m}$

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

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

How do they compare?

True Error Bound



Test Set Bound: Example

Suppose $\delta = 0.1$

Suppose $m = 100$

Suppose $\hat{c}_S = 2$

Square root Chernoff bound: $\Rightarrow c_D \in [-0.102, 0.142]$

Exact calculation $\Rightarrow c_D \in [0.0045, 0.0616]$

K-fold Cross Validation

Divide m examples into K subsets.

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

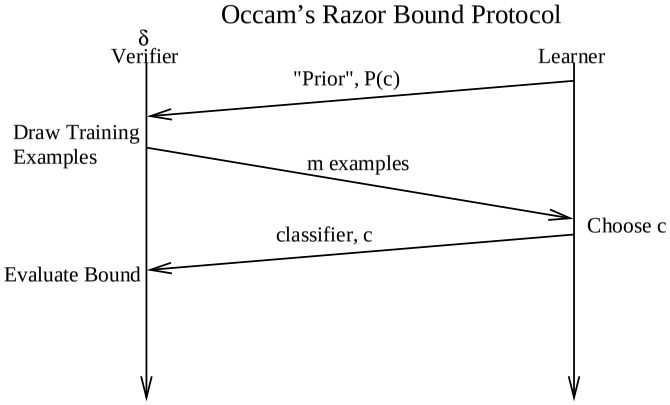
Not well understood theoretically. (Big open problem!)

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

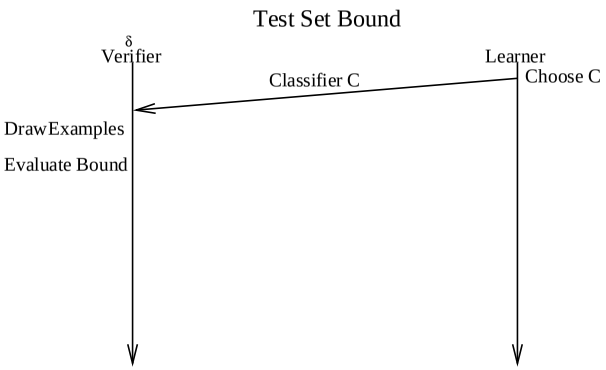
⇒ leave-one-out cross validation very prone to overfitting.

⇒ Some people fool themselves with overconfidence in Cross Validation.

Occam's Razor Bound Protocol



Interpretation: Interactive Proof of Learning



Training Set Bounds in General

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

⇒Train set bounds

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

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

Outline

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

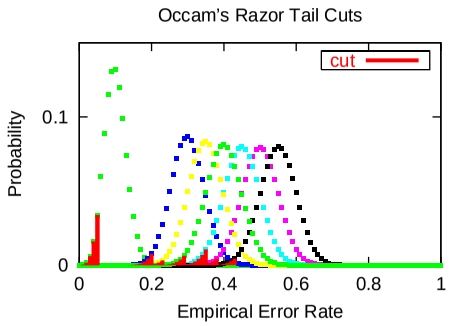
Occam's Razor Bound: Proof

Test set bound $\Rightarrow$

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

Negate to get:

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



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

With high probability no error falls in any tail.

Occam's Razor Bound: Proof

Test set bound $\Rightarrow$

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

Occam's Razor Bound: Proof

Test set bound $\Rightarrow$

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

Negate to get:

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

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

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

Negate again to get proof.

Next: Graphical proof

Occam's Razor Bound

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

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

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

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

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

Occam's Razor Bound: Proof

Test set bound $\Rightarrow$

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

Negate to get:

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

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

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

Occam's Razor Bound: Example

Suppose $\delta = 0.1$

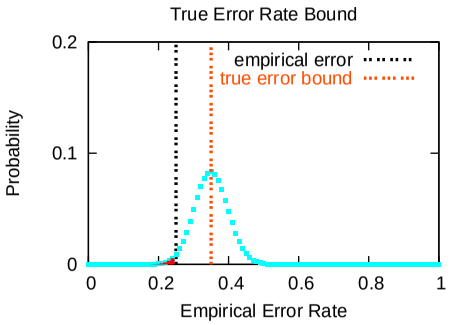
Suppose $m = 100$

Suppose $P(c) = 0.1$

Suppose $\hat{c}_S = 2$

Square root Chernoff $\Rightarrow c_D \in [-0.143, 0.183]$

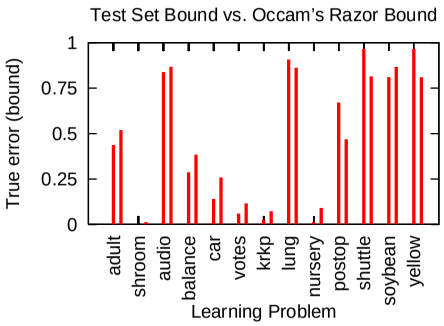
Exact calculation $\Rightarrow c_D \in [0.001, 0.089]$



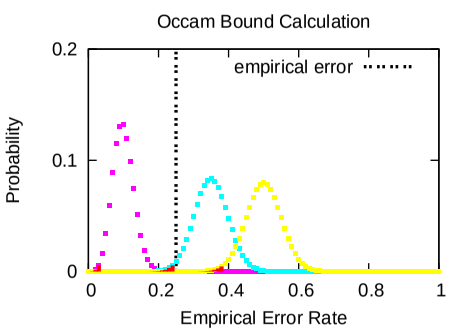
Bound = the largest true error rate for which the observation is not in the tail.

Outline

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



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



The chosen classifier has an unknown true error rate.

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

PAC-Bayes Bound: Theorem

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

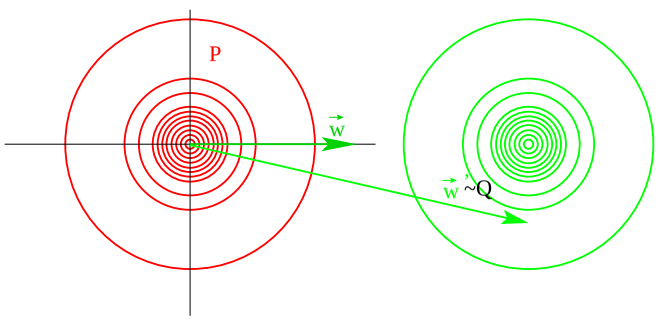
$$\Pr_{S\sim D^m}\left(\forall Q(c):\ \mathrm{KL}\left(\hat{Q}_S||Q_D\right) \leq \frac{\mathrm{KL}(Q||P) + \ln\frac{m+1}{\delta}}{m}\right) \geq 1-\delta$$

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 $\delta \in (0,1]$:

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

PAC-Bayes Margin Bound: Intuition



Isotropic Gaussian prior and posterior

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 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 $\vec{w}$

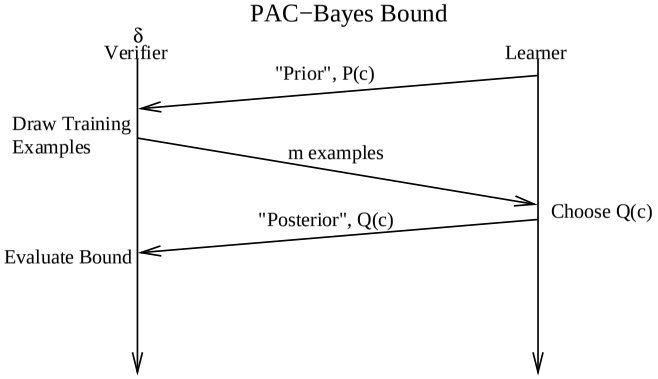
$\gamma(\vec{x},y) = \frac{y\vec{w}\cdot\vec{x}}{||\vec{w}||||\vec{x}||}$ = normalized margin

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



PAC-Bayes Bound: Application

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

Application: true error bounds for Support Vector Machines.

Classifier form:

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

Also note: Work by Matthias Seeger for Gaussian Processes.

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 $\Rightarrow$

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

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

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

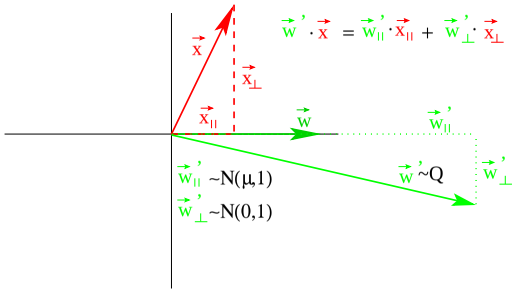
$\Rightarrow$ Corollary

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 at:

http://hunch.net/~jl/projects/prediction_bounds/bound/bound.html

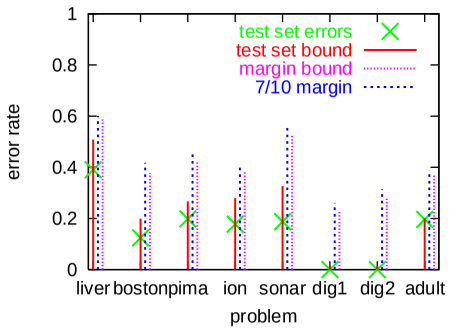


$$\hat{Q}(\vec{w},\mu)_S = E_{\vec{x},y\sim S,\vec{w}'\sim Q(\vec{w},\mu)} I\left(y \neq \text{sign}\left(\vec{w}'\cdot\vec{x}\right)\right)$$

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

Use properties of Gaussians to finish proof

PAC-Bayes Margin Bound Results



PAC-Bayes Margin Bound: Proof

Start with PAC-Bayes bound:

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

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

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

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

PAC-Bayes: Application to SVM

SVM classifier:

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

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

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

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

$\Rightarrow$ Margin bound applies to support vector machines.