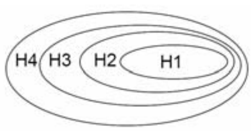
Machine Learning

10-701/15-781, Spring 2008

Computational Learning Theory II

Eric Xing

Lecture 12, February 25, 2008



Reading: Chap. 7 T.M book, and outline material

Last time: PAC and Agnostic Learning

- Finite H, assume target function c ∈ H

$$Pr(\exists h \in H,\ s.t.\ (error_{train}(h) = 0) \wedge (error_{true}(h) > \epsilon)\ ) \leq |H| e^{-\epsilon m}$$

  - Suppose we want this to be at most δ. Then *m* examples suffice:

$$m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$$

- Finite H, agnostic learning: perhaps c *not* in H

$$P(\exists h \in H,\ |\epsilon(h) - \hat{\epsilon}(h)| > \gamma) \quad = \quad 2k \exp(-2\gamma^2 m)$$

  - ➔ $$m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$
  - with probability at least (1-δ) every h in H satisfies

$$\epsilon(\hat{h}) \leq \left(\min_{h \in H} \epsilon(h) \right) + 2\sqrt{\frac{1}{m} \log \frac{2k}{\delta}}$$

What if H is not finite?

- Can't use our result for infinite H

- Need some other measure of complexity for H
  – Vapnik-Chervonenkis (VC) dimension!

What if H is not finite?

- Some Informal Derivation
  - Suppose we have an H that is parameterized by d real numbers. Since we are using a computer to represent real numbers, and IEEE double-precision floating point (double's in C) uses 64 bits to represent a floating point number, this means that our learning algorithm, assuming we're using double-precision floating point, is parameterized by 64d bits

$|H| = 2^{64d}$ $\quad m \le \frac{1}{\gamma^2} \log \frac{2k}{\delta}$

$m \le \frac{1}{\gamma^2} \log \frac{2^{64d}}{\delta} = \frac{64d}{\gamma^2} (1 - \log \frac{1}{\delta})$

- Parameterization

LR: θ^T

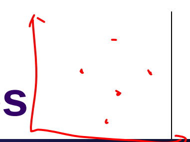
NB $\vec{\mu}_1 \sigma_1, \vec{\mu}_2 \sigma_2$

$(\cdot)$

How do we characterize "power"?

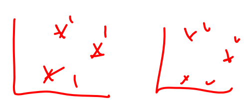
- Different machines have different amounts of "power".
- Tradeoff between:
  - More power: Can model more complex classifiers but might overfit.
  - Less power: Not going to overfit, but restricted in what it can model
- How do we characterize the amount of power?

Shattering a Set of Instances

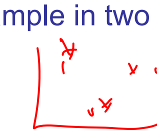


- *Definition*: Given a set $\mathcal{S} = \{x^{(1)}, \dots, x^{(d)}\}$ (no relation to the training set) of points $x^{(i)} \in X$, we say that $\mathcal{H}$ shatters $\mathcal{S}$ if $\mathcal{H}$ can realize any labeling on $\mathcal{S}$.

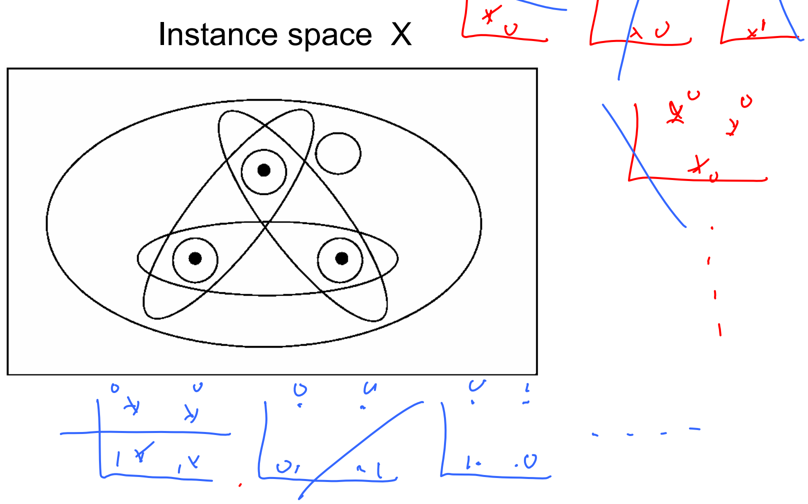
  I.e., if for any set of labels $\{y^{(1)}, \dots, y^{(d)}\}$, there exists some $h \in \mathcal{H}$ so that $h(x^{(i)}) = y^{(i)}$ for all $i = 1, \dots, d$.



- There are 2^L different ways to separate the sample in two sub-samples (a dichotomy)

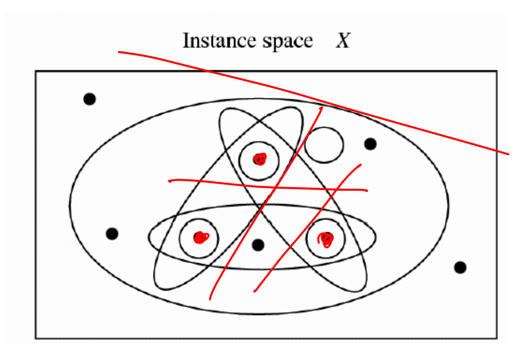


Three Instances Shattered



The Vapnik-Chervonenkis Dimension

- *Definition*: The **Vapnik-Chervonenkis dimension**, *VC*(*H*), of hypothesis space *H* defined over instance space *X* is the size of the *largest finite subset* of *X* shattered by *H* . If arbitrarily large finite sets of *X* can be shattered by *H* , then *VC*(*H*) ≡ ∞.



VC dimension: examples

Consider X = $\mathbb{R}$, want to learn c: X→{0,1}

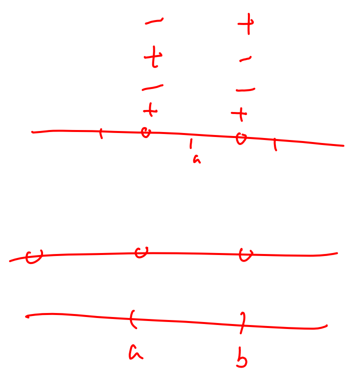
What is VC dimension of

- Open intervals:

  H1: if x>a, then y=1 else y=0

- Closed intervals:

  H2: if a<x<b, then y=1 else y=0



VC dimension: examples

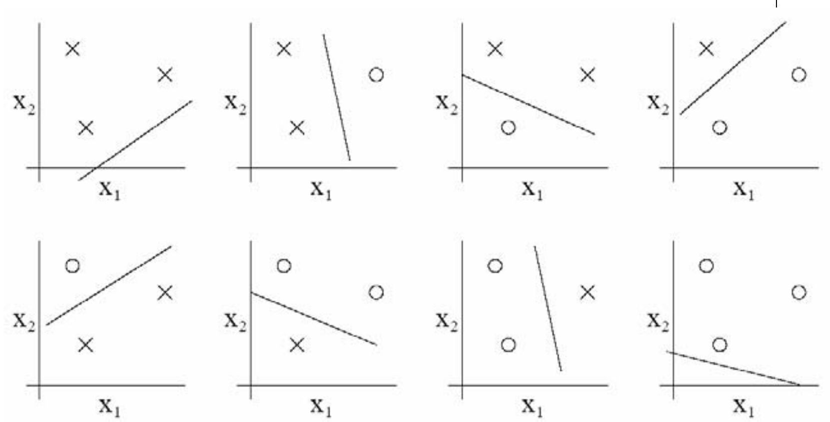
Consider X = $\mathbb{R}^2$, want to learn c: X→{0,1}

- What is VC dimension of lines in a plane?

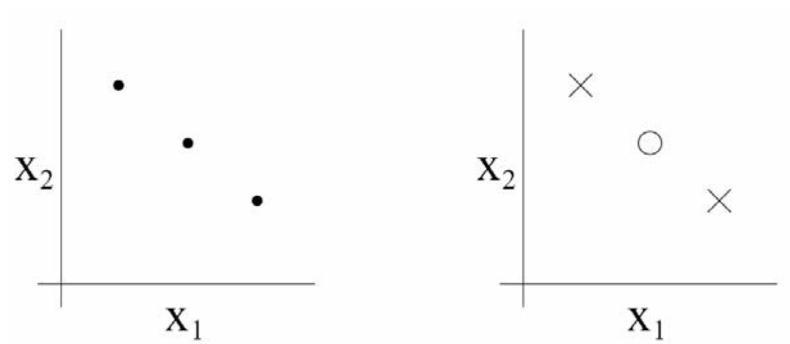
  H= { ((wx+b)>0 → y=1) }

  VC(H)=3

(a) (b)

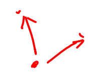


- For any of the eight possible labelings of these points, we can find a linear classier that obtains "zero training error" on them.
- Moreover, it is possible to show that there is no set of 4 points that this hypothesis class can shatter.



- The VC dimension of H here is 3 even though there may be sets of size 3 that it cannot shatter.
- under the definition of the VC dimension, in order to prove that VC(H) is at least *d*, we need to show only that there's at least one set of size d that H can shatter.

- **Theorem** Consider some set of *m* points in $\mathbb{R}^n$. Choose any one of the points as origin. Then the *m* points can be shattered by oriented hyperplanes if and only if the position vectors of the remaining points are linearly independent.



- **Corollary**: The VC dimension of the set of oriented hyperplanes in $\mathbb{R}^n$ is *n*+1.

  Proof: we can always choose $n + 1$ points, and then choose one of the points as origin, such that the position vectors of the remaining *n* points are linearly independent, but can never choose $n + 2$ such points (since no $n + 1$ vectors in $\mathbb{R}^n$ can be linearly independent).

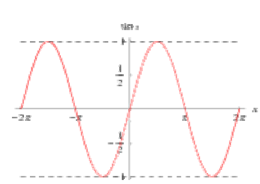
The VC Dimension and the Number of Parameters

- The VC dimension thus gives concreteness to the notion of the capacity of a given set of *h*.
- Is it true that learning machines with many parameters would have high VC dimension, while learning machines with few parameters would have low VC dimension?

An infinite-VC function with just one parameter!

$$f(x, \alpha) \equiv \theta(\sin(\alpha x)), \quad x, \alpha \in R$$

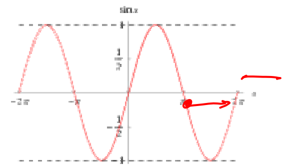
where θ is an indicator function



An infinite-VC function with just one parameter

- You choose some number *l*, and present me with the task of finding *l* points that can be shattered. I choose them to be

$$x_i = 10^{-i} \quad i = 1, \ldots, l.$$



- You specify any labels you like:

$$y_1, y_2, \ldots, y_l, \quad y_i \in \{-1, 1\}$$

- Then $f(\alpha)$ gives this labeling if I choose α to be

$$\alpha = \pi\left(1 + \sum_{i=1}^{l} \frac{(1 - y_i)10^i}{2}\right)$$

- Thus the VC dimension of this machine is infinite.

Sample Complexity from VC Dimension

- How many randomly drawn examples suffice to ε-exhaust $VS_{H,S}$ with probability at least (1 - δ)?

ie., to guarantee that any hypothesis that perfectly fits the training data is probably (1-δ) approximately (ε) correct on testing data from the same distribution

$$m \geq \frac{1}{\varepsilon}\left(4\log_2(2/\delta) + 8VC(H)\log_2(13/\varepsilon)\right)$$

Compare to our earlier results based on $|H|$:

$$m \geq \frac{1}{2\varepsilon^2}\left(\ln|H| + \ln(1/\delta)\right)$$

Mistake Bounds

So far: how many examples needed to learn?

What about: how many mistakes before convergence?

Let's consider similar setting to PAC learning:

- Instances drawn at random from *X* according to distribution *D*
- Learner must classify each instance before receiving correct classification from teacher
- Can we bound the number of mistakes learner makes before converging?

Statistical Learning Problem

- A model computes a function: $h(X, w)$
- Problem : minimize in *w* Risk Expectation

$$R(w) = \int Q(z, w) dP(z)$$

  - w : a parameter that specifies the chosen model
  - $z = (X, y)$ are possible values for attributes (variables)
  - Q measures (quantifies) model error cost
  - $P(z)$ is the underlying probability law (unknown) for data z

Statistical Learning Problem (2)

- We get L data from learning sample $(z_1, .. , z_L)$, and we suppose them iid sampled from law $P(z)$.
- To minimize $R(w)$, we start by minimizing Empirical Risk over this sample :

$$E(W) = \frac{1}{L}\sum_{i=1}^{L} Q(Z_i, W)$$

- We shall use such an approach for :
  - classification (eg. Q can be a cost function based on cost for misclassified points)
  - regression (eg. Q can be a cost of least squares type)

Statistical Learning Problem (3)

- Central problem for Statistical Learning Theory:

  What is the relation
  between Risk Expectation $R(W)$
  and Empirical Risk $E(W)$?

- How to define and measure a generalization capacity ("robustness") for a model ?

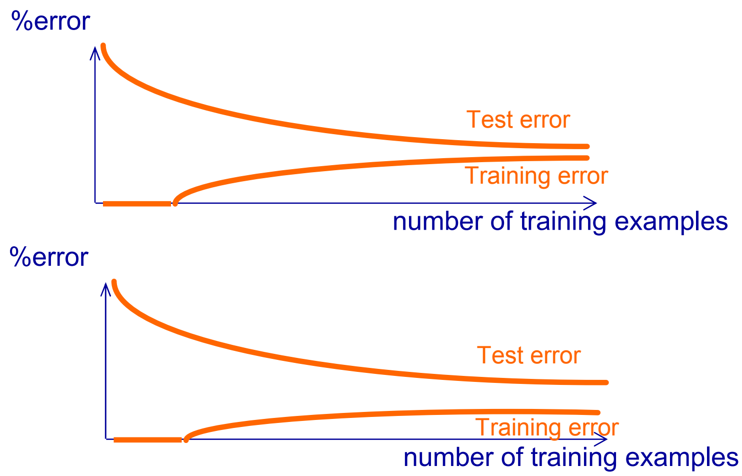
Four Pillars for SLT

- Consistency (guarantees generalization)
  - Under what conditions will a model be consistent ?
- Model convergence speed (a measure for generalization)
  - How does generalization capacity improve when sample size L grows?
- Generalization capacity control
  - How to control in an efficient way model generalization starting with the only given information we have: our sample data?
- A strategy for good learning algorithms
  - Is there a strategy that guarantees, measures and controls our learning model generalization capacity ?

Consistency

A learning process (model) is said to be consistent if model error, measured on new data sampled from the same underlying probability laws of our original sample, converges, when original sample size increases, towards model error, measured on original sample.

Consistent training?



Vapnik main theorem

- Q : Under which conditions will a learning model be consistent?
- A : A model will be consistent if and only if the function *h* that defines the model comes from a family of functions *H* with finite VC dimension *d*
- A finite VC dimension d not only guarantees a generalization capacity (consistency), but to pick *h* in a family *H* with finite VC dimension *d* is the only way to build a model that generalizes.

Model convergence speed (generalization capacity)

- Q : What is the nature of model error difference between learning data (sample) and test data, for a sample of finite size *m*?
- A : This difference is no greater than a limit that only depends on the ratio between VC dimension *d* of model functions family *H*, and sample size *m*, ie *d/m*

This statement is a new theorem that belongs to Kolmogorov-Smirnov way for results, ie theorems that do not depend on data's underlying probability law.

Agnostic Learning: VC Bounds

- ***Theorem***: Let H be given, and let d = VC(H). Then with probability at least 1-δ, we have that for all h $\in$ H,

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} - \frac{1}{m}\log\delta}\right)$$

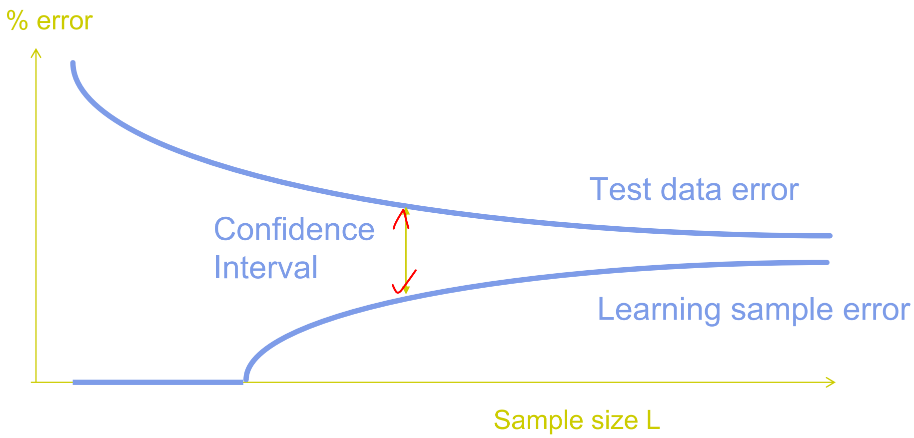
or

$$\epsilon(h) \leq \underbrace{\hat{\epsilon}(h)}_{\text{bias}} + \underbrace{O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} - \frac{1}{m}\log\delta}\right)}_{\text{variance}}$$

recall that in finite H case, we have:

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq \sqrt{\frac{1}{m}\log 2k - \frac{1}{m}\log\delta}$$

Model convergence speed



How to control model generalization capacity

Risk Expectation = Empirical Risk + Confidence Interval

- To minimize Empirical Risk alone will not always give a good generalization capacity: one will want to minimize the sum of Empirical Risk and Confidence Interval
- What is important is not the numerical value of the Vapnik limit, most often too large to be of any practical use, it is the fact that this limit is a non decreasing function of model family function “richness”

Empirical Risk Minimization

- With probability $1-\delta$, the following inequality is true:

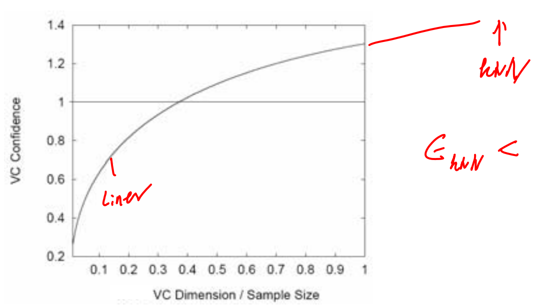
$$\int \left(y - f(x, w^0)\right)^2 dP(x,y) < \frac{1}{m}\sum_{i=1}^{m}\left(y_i - f(x_i, w^0)\right)^2 + \sqrt{\frac{d\left(\ln(2m/d)+1\right) - \ln\delta}{m}}$$

- where w^0 is the parameter w value that minimizes Empirical Risk:

$$E(W) = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - f(x_i, w)\right)^2$$

Minimizing The Bound by Minimizing *d*

- Given some selection of learning machines whose empirical risk is zero, one wants to choose that learning machine whose associated set of functions has minimal VC dimension.

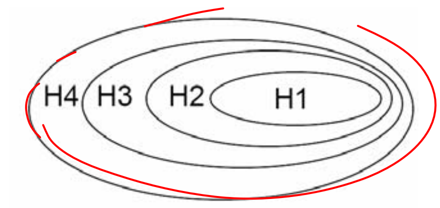


$E_{kNN} < E_{linear}$

  - By doing this we can attain an upper bound on the actual risk. This does not prevent a particular machine with the same value for empirical risk, and whose function set has higher VC dimension, from having better performance.
  - What is the VC of a kNN? $VC_{kNN} = \infty$

Structural Risk Minimization

- Which hypothesis space should we choose?
- Bias / variance tradeoff



- SRM: choose H to minimize bound on true error!

$$\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} - \frac{1}{m}\log\delta}\right)$$

unfortunately a somewhat loose bound...

SRM strategy (1)

- With probability $1-\delta$,

$$\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} - \frac{1}{m}\log\delta}\right)$$

- When m/d is small (d too large), second term of equation becomes large
- SRM basic idea for strategy is to minimize simultaneously both terms standing on the right of above majoring equation for $\varepsilon(h)$
- To do this, one has to make *d* a controlled parameter

SRM strategy (2)

- Let us consider a sequence $H_1 < H_2 < .. < H_n$ of model family functions, with respective growing VC dimensions

  $d_1 < d_2 < .. < d_n$

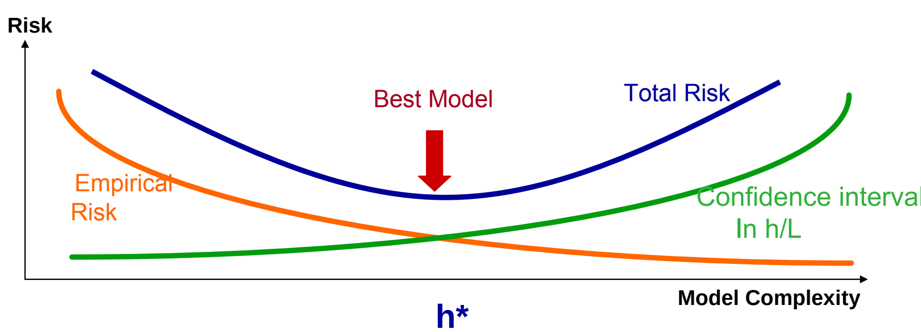
- For each family H_i of our sequence, the inequality

$$\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} - \frac{1}{m}\log\delta}\right)$$

  is valid
  - That is, for each subset, we must be able either to compute *d*, or to get a bound on *d* itself.
- SRM then consists of finding that subset of functions which minimizes the bound on the actual risk.

SRM strategy (3)

SRM : find i such that expected risk $\varepsilon(h)$ becomes minimum, for a specific $d^*=d_i$, relating to a specific family H_i of our sequence; build model using h from H_i



Putting SRM into action: linear models case (1)

- There are many SRM-based strategies to build models:
- In the case of linear models

  $y = \langle w|x\rangle + b,$

  one wants to make $||w||$ a controlled parameter: let us call H_C the linear model function family satisfying the constraint:

  $||w|| < C$

> Vapnik Major theorem:
>
> When C decreases, $d(H_C)$ decreases
>
> $||x|| < R$

Putting SRM into action: linear models case (2)

- To control $||w||$, one can envision two routes to model:
  - *Regularization/Ridge Regression*, ***ie min. over w and b***

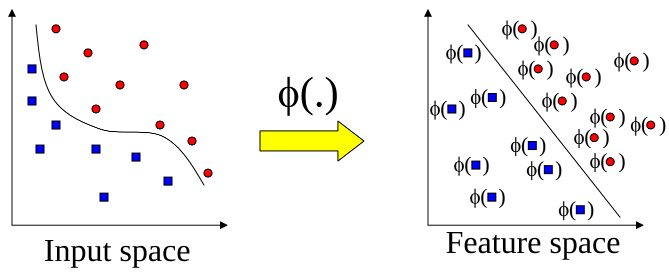
    $RG(w,b) = S\{(y_i - \langle w|x_i\rangle - b)^2 \,|\, i=1,..,L\} + \lambda \,||w||^2$

  - *Support Vector Machines (SVM)*, ***ie solve directly an optimization problem (hereunder: classif. SVM, separable data)***

    Minimize $||w||^2$,
    with ($y_i = +/-1$)
    and $y_i(\langle w|x_i\rangle + b) >= 1$ *for all* $i=1,..,L$

The VC Dimension of SVMs

- An SVM finds a linear separator in a Hilbert space, where the original date x can be mapped to via a transformation $\phi(x)$.



- Recall that the kernel trick used by SVM alleviates the need to find explicit expression of $\phi(.)$ to compute the transformation

The Kernel Trick

- Recall the SVM optimization problem

$$\max_{\alpha} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, k$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

- The data points only appear as inner product
- As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly
- Define the kernel function K by $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

Mercer's Condition

- For which kernels does there exist a pair $\{\mathcal{H}, \phi(.)\}$ with the valid geometric properties (e.g., nonnegative dot-product) for a transformation satisfied, and for which does there not?

- *Mercer's Condition for Kernels*
  - There exists a mapping $\phi(.)$ and an expansion

$$K(x,y) = \sum_i \phi_i(x)\phi_i(y)$$

iff for any g(x) such that

$$\int g(x)^2 dx \quad \text{is finite}$$

then

$$\int K(x,y)g(x)g(y)dxdy \geq 0$$

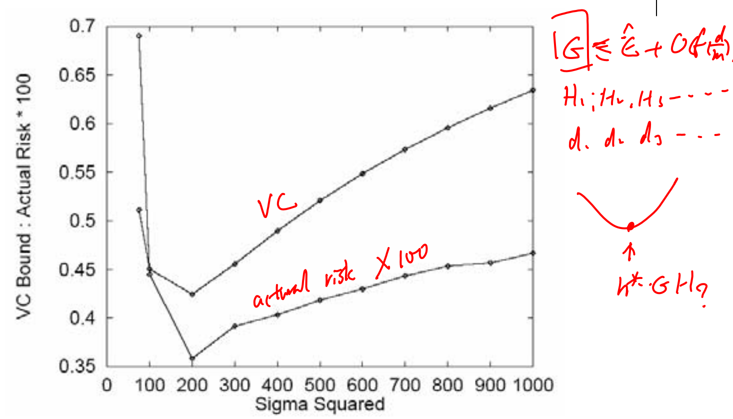
The VC Dimension of SVMs

- We will call any kernel that satisfies Mercer's condition a positive kernel, and the corresponding space *H* the embedding space.

- We will also call any embedding space with minimal dimension for a given kernel a "minimal embedding space".

- **Theorem**: *Let K be a positive kernel which corresponds to a minimal embedding space H. Then the VC dimension of the corresponding support vector machine (where the error penalty C is allowed to take all values) is dim*(*H*) + 1

VC and the Actual Risk



- It is striking that the two curves have minima in the same place: thus in this case, the VC bound, although loose, seems to be nevertheless predictive.

What You Should Know

- Sample complexity varies with the learning setting
  - Learner actively queries trainer
  - Examples provided at random
- Within the PAC learning setting, we can bound the probability that learner will output hypothesis with given error
  - For ANY consistent learner (case where c in H)
  - For ANY "best fit" hypothesis (agnostic learning, where perhaps c not in H)
- VC dimension as measure of complexity of H
- Quantitative bounds characterizing bias/variance in choice of H
  - but the bounds are quite loose...
- Mistake bounds in learning
- Conference on Learning Theory: http://www.learningtheory.org