VC Dimension

Berrin Yanikoglu

Slides are expanded from the
Gutierrez-Osuna and Andrew Moore Slides

- Previous slides (PAC learning) put a bound on the true error for finite hypothesis spaces.

- What if the hypothesis space H is infinite dimensional?
  - In that case the bound is trivially true (even bigger than 1).
- Can we still find a bound for the true error?

True Error of A Hypothesis

- Two Notions of Error
  - Training error of hypothesis h with respect to target concept c :

    How often $h(x) \neq c(x)$ **over training instances**

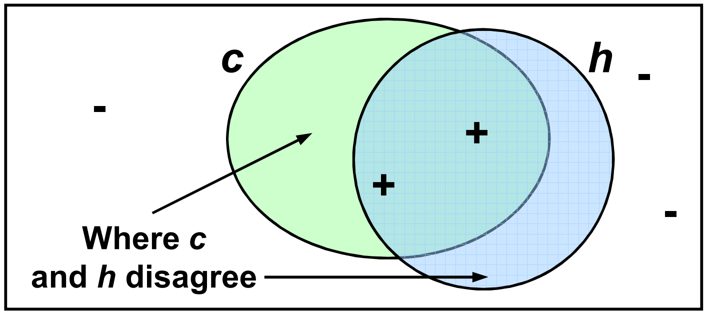
  - True error of hypothesis h with respect to target concept c :

    How often $h(x) \neq c(x)$ **over random instances drawn from distribution D**

- Definition
  - The true error (denoted $error_D(h)$) of hypothesis h with respect to target concept c and distribution D is the probability that h will misclassify an instance drawn at random according to D.

$$error_D(h) \equiv \Pr_{x \in D}[c(x) \neq h(x)]$$

Instance Space *X*



Two Notions of Error Mitchell Book notation

Training error of hypothesis h with respect to target concept c

- How often $h(x) \neq c(x)$ over training instances D

$$error_{\mathrm{D}}(h) \equiv \Pr_{x \in \mathrm{D}}[c(x) \neq h(x)]$$

Set of training examples

True error of hypothesis h with respect to c

- How often $h(x) \neq c(x)$ over future instances drawn at random from $\mathcal{D}$

$$error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}}[c(x) \neq h(x)]$$

Probability distribution P(x)

- **Empirical Risk Minimization (ERM)**
  - A formal term for a simple concept: find the function f(x) that minimizes the average risk on the training set
  - Minimizing the empirical risk is not a bad thing to do, provided that sufficient training data is available, since the law of large numbers ensures that the empirical risk will asymptotically converge to the expected risk for $n \to \infty$
  - However, for small samples, one cannot guarantee that ERM will also minimize the expected risk. This is the all too familiar issue of generalization.

- **How do we avoid overfitting?**
  - By controlling model complexity.
  - Intuitively, we should prefer the simplest model that explains the data (Occam's razor)

Triple Trade-Off

- There is a trade-off between three factors (Dietterich, 2003):
  1. Complexity of $\mathcal{H}$, $c(\mathcal{H})$,
  2. Training set size, N,
  3. Generalization error, E, on new data

- As $N\uparrow$, $E\downarrow$
- As $c(\mathcal{H})\uparrow$, first $E\downarrow$ and then $E\uparrow$

Complexity

- “Complexity” is a measure of a set of classifiers, not any specific (fixed) classifier

- • Many possible measures
  - degrees of freedom
  - description length
  - Vapnik-Chervonenkis (VC) dimension
  - etc.

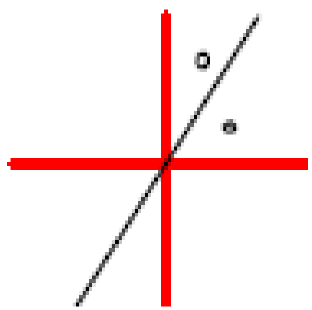
SHATTERING

Shattering a Set of Instances

Definition: a **dichotomy** of a set S is a partition of S into two disjoint subsets.

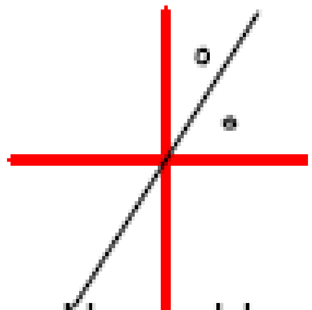
Definition: a set of instances S **is shattered** by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy.

- Question: Can the following f shatter the following points?



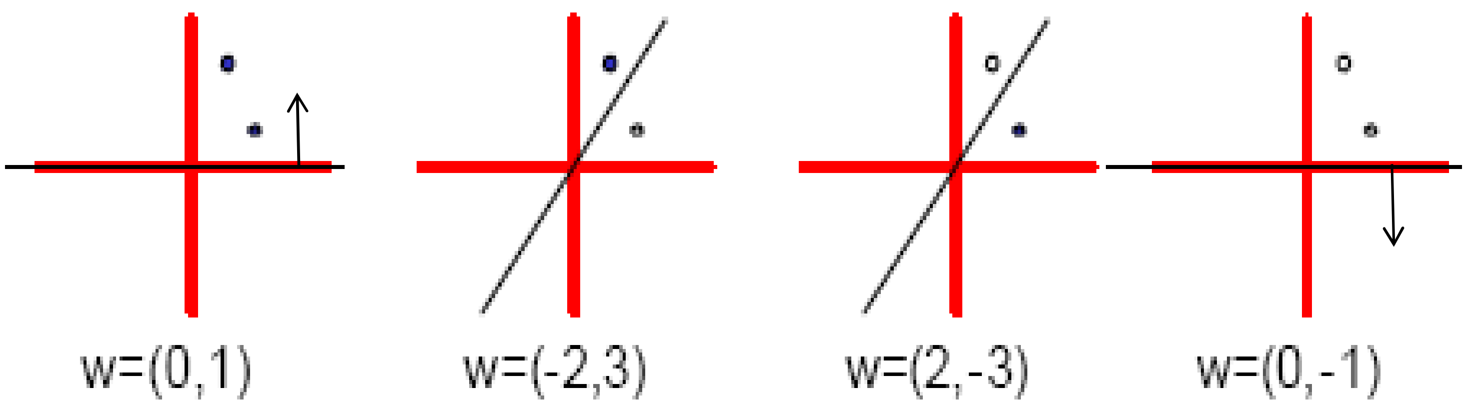
$f(x, \mathbf{w}) = \text{sign}(x.\mathbf{w})$

- Question: Can the following f shatter the following points?



$f(x,w) = \text{sign}(x.w)$

- Answer: No problem. There are four training sets to consider

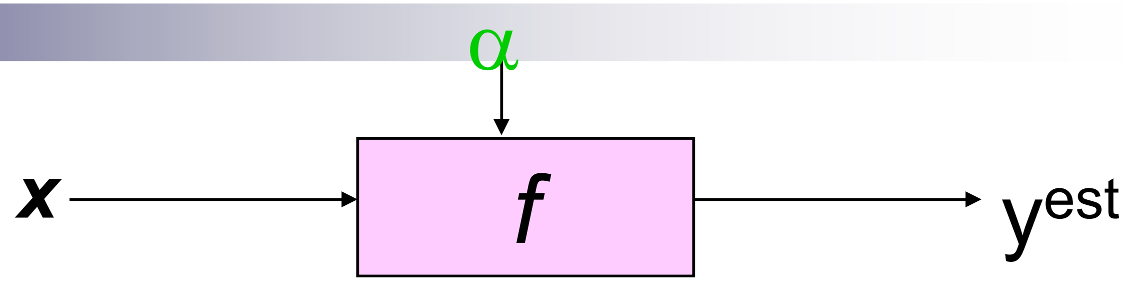


The Vapnik-Chervonenkis Dimension

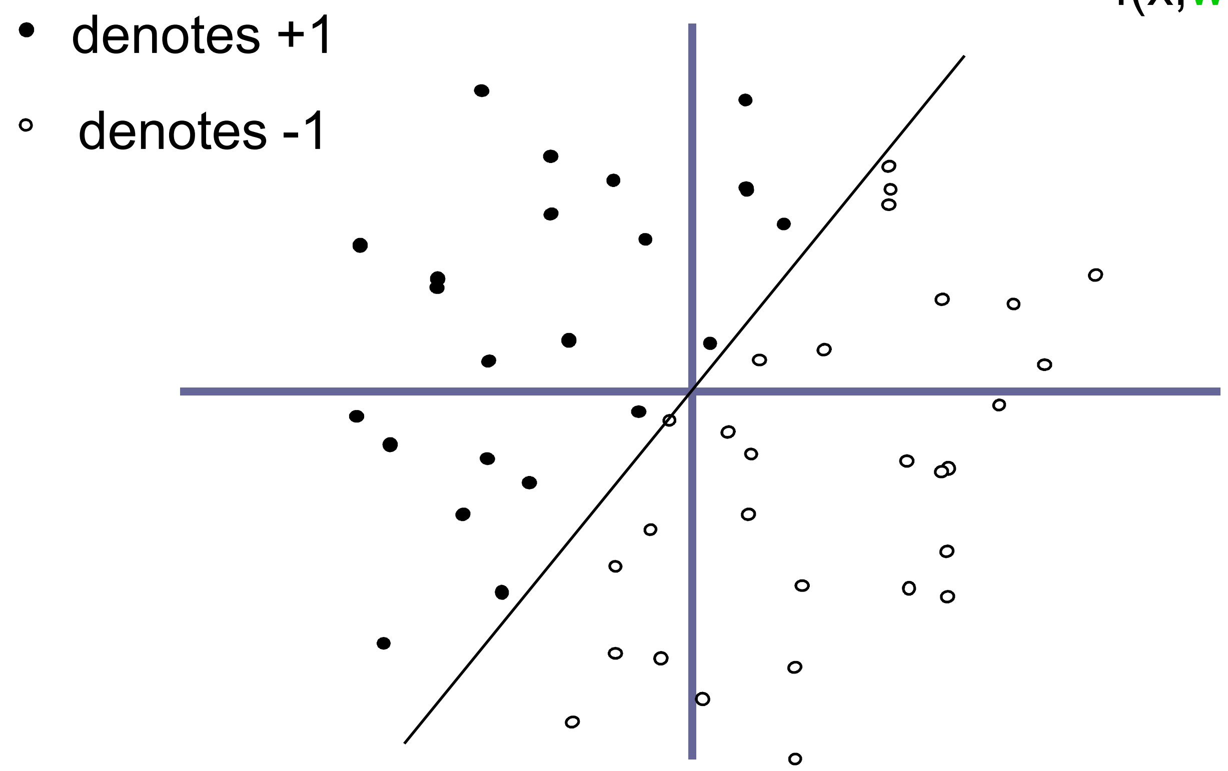
Definition: The **Vapnik-Chervonenkis dimension**, $VC(H)$, of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H. If arbitrarily large finite sets of X can be shattered by H, then $VC(H) \equiv \infty$.

VC DIMENSION EXAMPLES

Examples

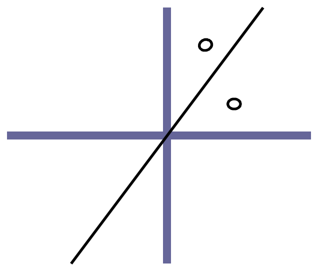


$f(x, \mathbf{w}) = \mathrm{sign}(x.\mathbf{w})$



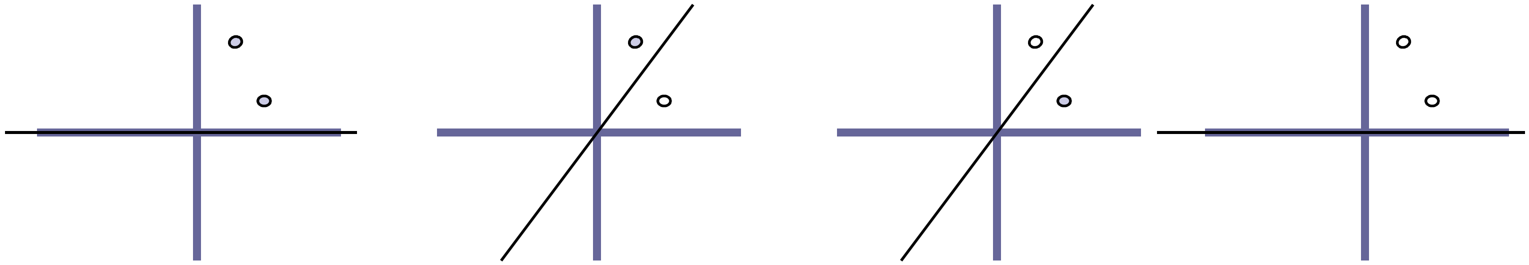
Shattering

- Question: Can the following f shatter the following points?



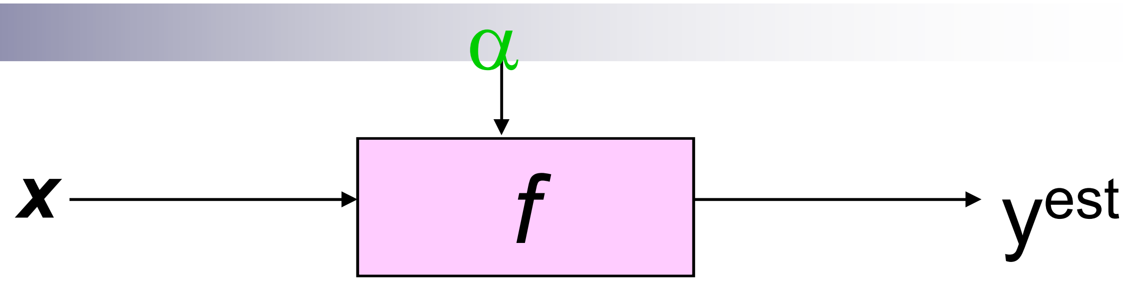
$f(x,w) = sign(x.w)$

- Answer: Yes. There are four possible training set types to consider:



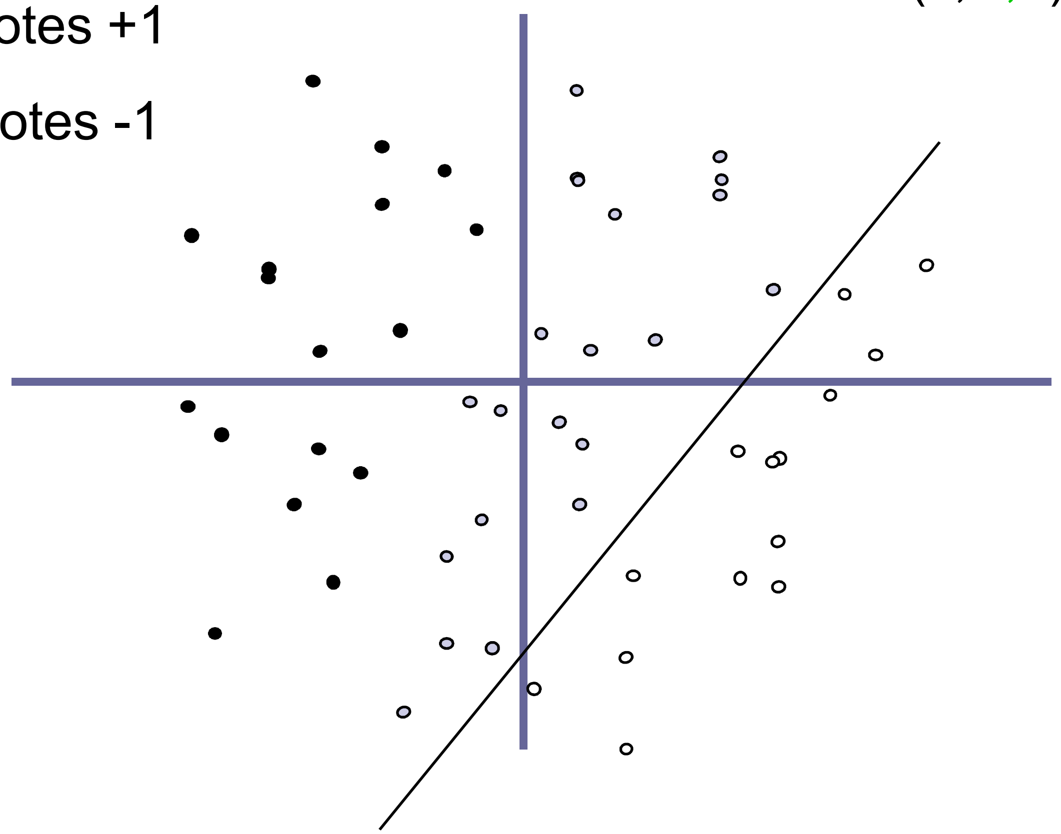
w=(0,1) w=(-2,3) w=(2,-3) w=(0,-1)

Examples



f(x,w,b) = sign(x.w+b)

- • denotes +1
- ° denotes -1



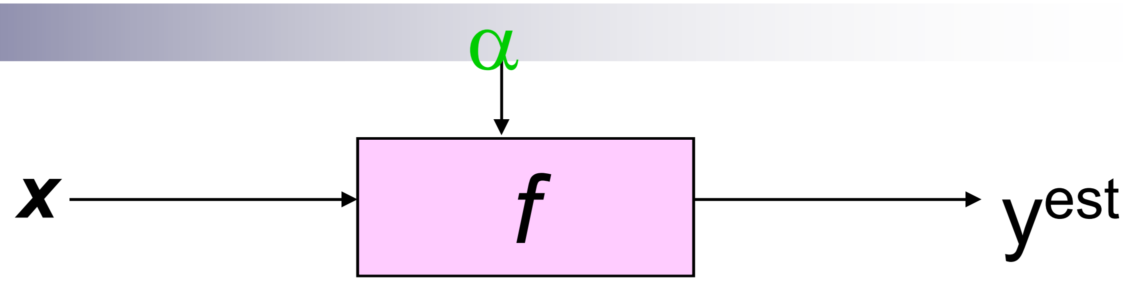
VC dim of linear classifiers in d-dimensions

If input space is d-dimensional and if **f** is *sign(**w**.**x**-b)*, what is the VC-dimension?

- h=d+1

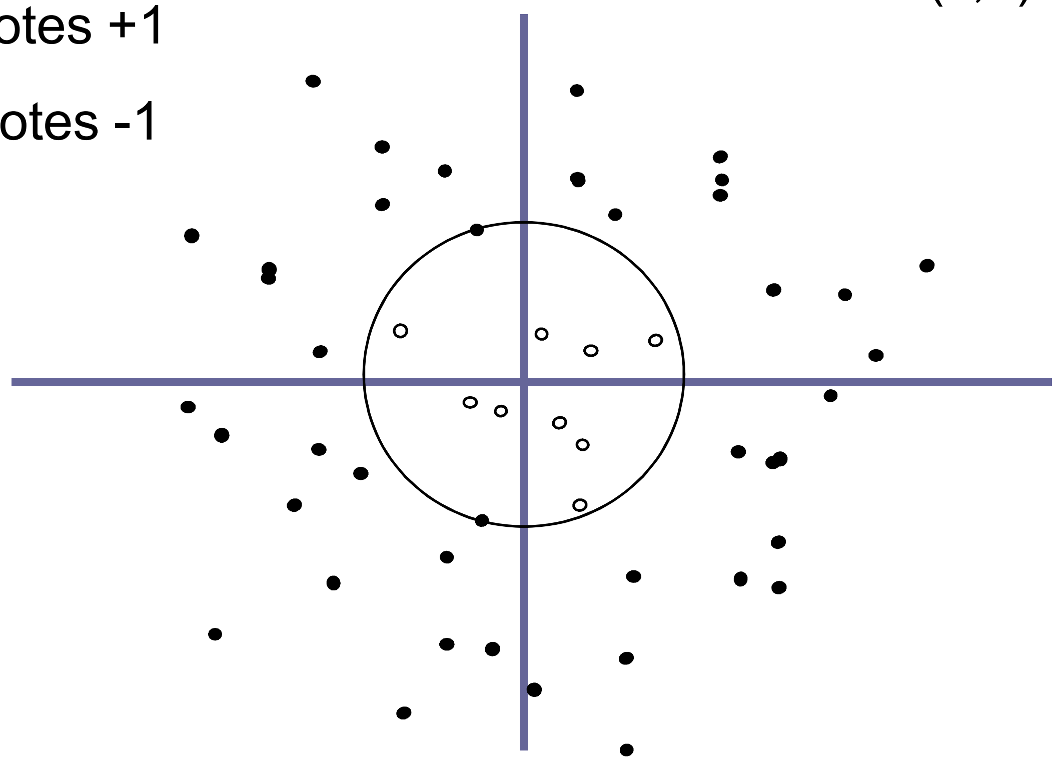
- Lines in 2D can shatter 3 points
- Planes in 3D space can shatter 4 points
- Hyperplanes in D-dimensional can shatter d+1 points

Examples



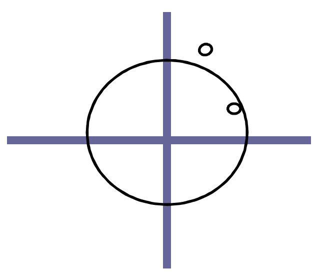
f(x,b) = sign(x.x – b)

- • denotes +1
- ∘ denotes -1



Shattering

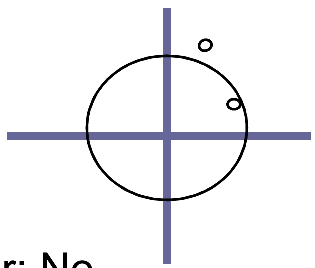
- Question: Can the following f shatter the following points?



$f(x,b) = \mathrm{sign}(x.x-b)$

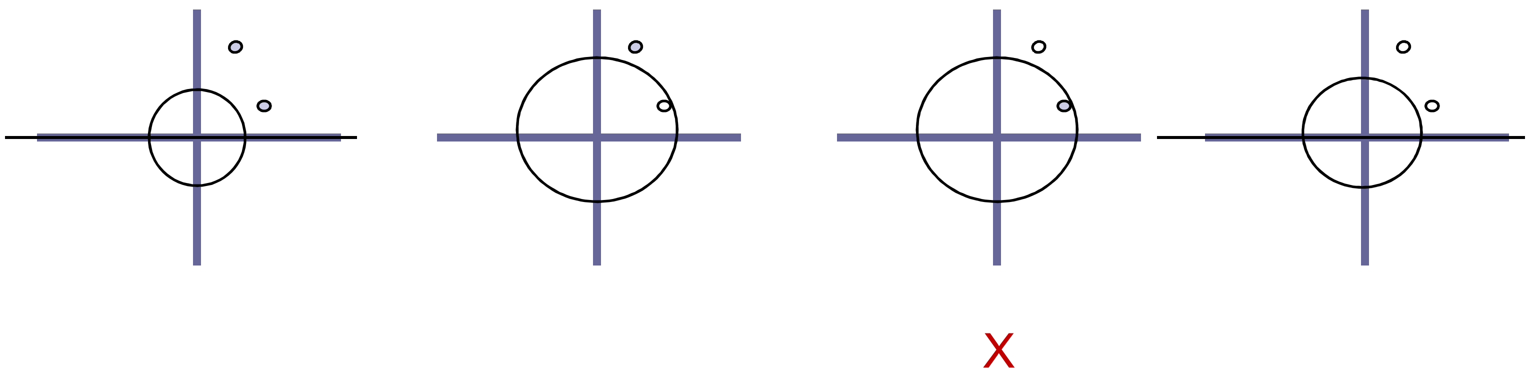
Shattering

- Question: Can the following f shatter the following points?



$f(x,b) = sign(x.x-b)$

- Answer: No.



Reformulated circle

Given machine ***f***, the VC-dimension *h* is

> The maximum number of points that can be arranged so that ***f*** shatter them.

Example: For 2-d inputs, what's VC dimension of f(x,q,b) = sign(qx.x-b)

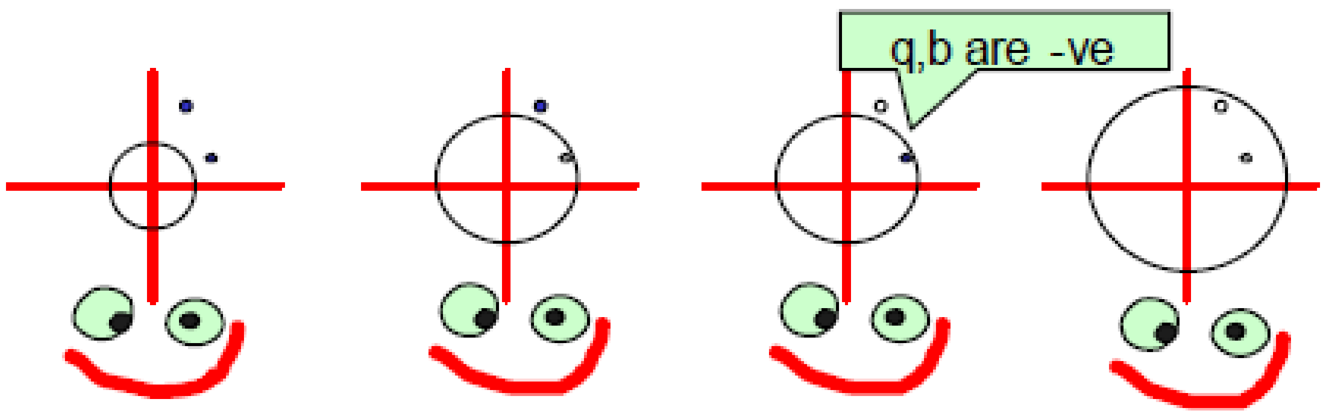
Reformulated circle

Given machine ***f***, the VC-dimension *h* is

> The maximum number of points that can be arranged so that ***f*** shatter them.

Example: What's VC dimension of f(x,q,b) = sign(qx.x-b)

- Answer = 2



- Note that if we pick 2points at the same distance to the origin, they cannot be shattered. But we are interested to know

  "if all possible labellings of some n-points can be shattered".

- Can you find 3 points such that all possible labellings can be shattered?

VC dimension: examples

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

Using a more specific terminology

What is VC dimension of

- H1 = { (w·x+b)>0 → y=1) | w $\in \Re^2$, b $\in \Re$}
  - VC(H1)=3
  - For linear separating hyperplanes in n dimensions, VC(H)=n+1

VC dimension: examples

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

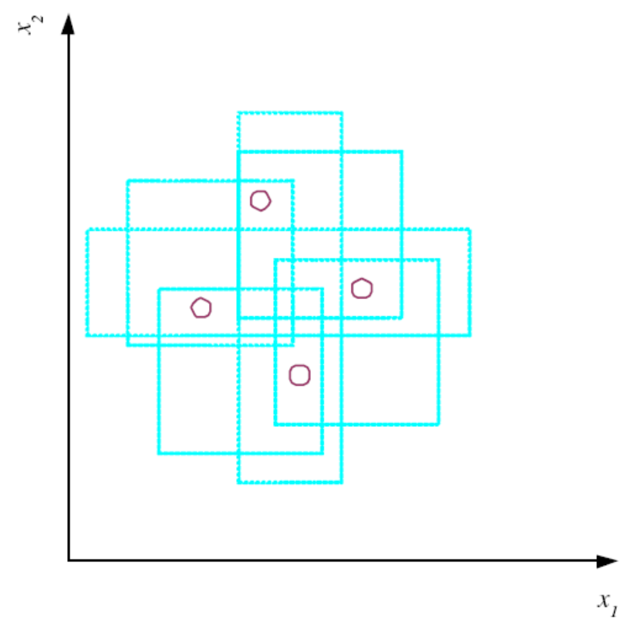
What is VC dimension of

- H1 = $\{ (x > a \rightarrow y=1) \mid a \in \Re\}$
  - VC(H1)=1
- H2 = $\{ (x > a \rightarrow y=1) \mid a \in \Re\} + \{ (x < a \rightarrow y=1) \mid a \in \Re\}$
  - VC(H2)=2

What is the VC dimension of axis-aligned rectangles?

- $\mathcal{H}$ shatters *N* if there exists N points and $h \in \mathcal{H}$ such that h is consistent for any labelings of those N points.

- VC(*axis aligned rectangles*) = *4*



- What does this say about using rectangles as our hypothesis class?

VC (*Vapnik-Chervonenkis)* Dimension

- VC dimension is pessimistic: in general we do not need to worry about *all* possible labelings

- It is important to remember that one can choose the arrangement of points in the space, but then the hypothesis must be consistent with all possible labelings of those fixed points.

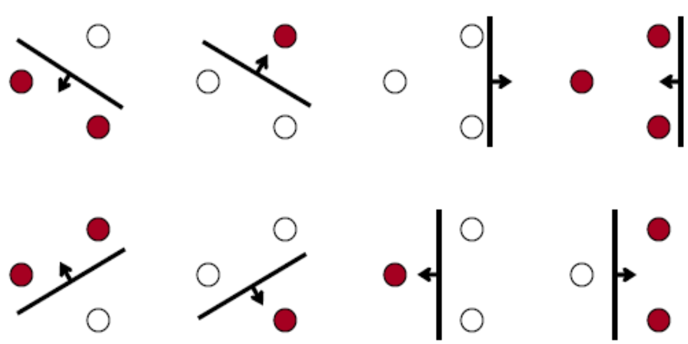
VC (*Vapnik-Chervonenkis)* Dimension

- **The Vapnik-Chervonenkis dimension is a measure of the complexity (or capacity) of a class of functions f(α)**
  - The VC dimension measures the **largest number of examples that can be explained by the family f(α).**

- **The basic argument is that high capacity and generalization properties are at odds**
  - If the family f(α) has enough capacity to explain every possible dataset, we should not expect these functions to generalize very well.
  - On the other hand, if functions f(α) have small capacity but they are able to explain our particular dataset, we have stronger reasons to believe that they will also work well on unseen data.

VC Dimension (3)

- **Consider a binary classification problem in R^2, and let $f(\alpha)$ be the family of oriented hyperplanes (e.g., perceptrons)**
  - For N=3, one can perform a linear separation of all points for every possible class assignment (see examples below)
  - For N=4, a hyperplane cannot separate all possible class assignments (e.g., consider the XOR problem)
    - Regardless of how you select the 4 points…
- **Therefore, the VC dimension of the set of oriented lines in R^2 is 3**
  - It can be shown that the VC dimension of the family of oriented separating hyperplanes in R^D is at least D+1



Structural Risk Minimization

Learning and VC-dimension

- Let d_{VC} be the VC-dimension of our set of classifiers F.

  Theorem: With probability at least $1 - \delta$ over the choice of the training set, for all $h \in F$

$$\mathcal{E}(h) \leq \hat{\mathcal{E}}_n(h) + \epsilon(n, d_{VC}, \delta)$$

  where

$$\epsilon(n, d_{VC}, \delta) = \sqrt{\frac{d_{VC}(\log(2n/d_{VC}) + 1) + \log(1/(4\delta))}{n}}$$

n is the size of the training set; d_{VC} is the VC dimension

Structural risk minimization

- In structural risk minimization we define the models in terms of VC-dimension (or refinements)

  Model 1 $\quad d_{VC} = d_1$
  Model 2 $\quad d_{VC} = d_2$
  Model 3 $\quad d_{VC} = d_3$

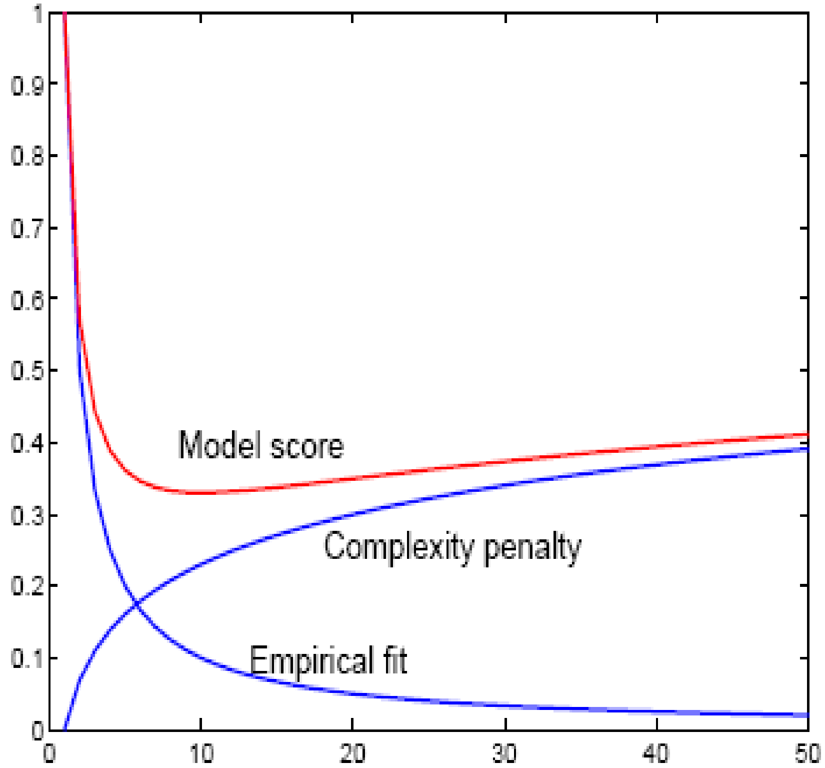
  where $d_1 \leq d_2 \leq d_3 \leq \dots$

- The selection criterion: lowest upper *bound* on the expected loss

$$\text{Expected loss} \leq \text{Empirical loss} + \text{Complexity penalty}$$

Structural risk minimization cont'd

- Competition of terms...
  1. Empirical loss decreases with increasing d_{VC}
  2. Complexity penalty increases with increasing d_{VC}



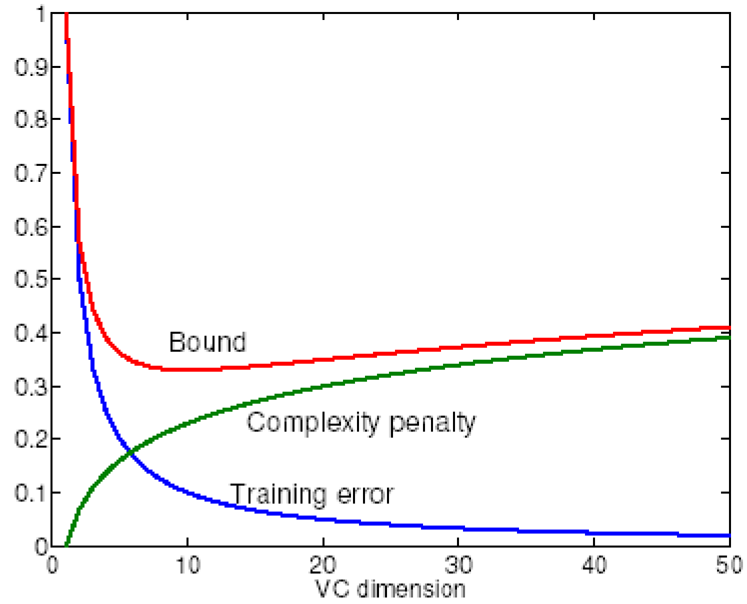
- We find the minimum of the model score (bound).

Structural risk minimization cont'd

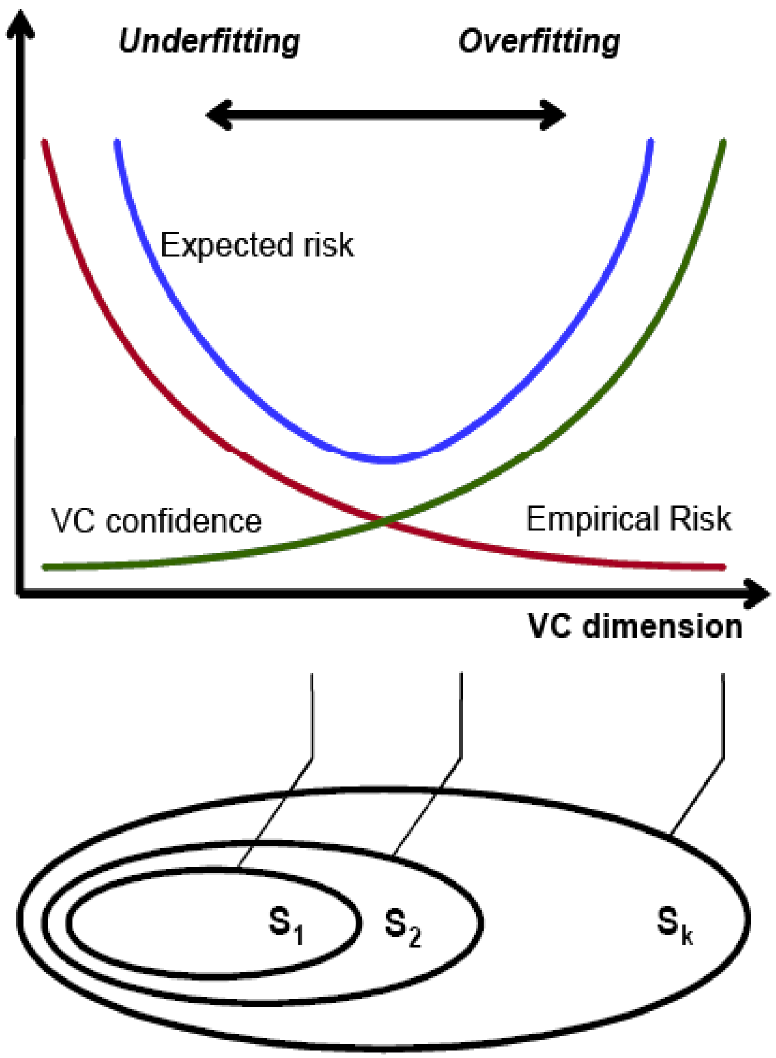
- We choose the model class F_i that minimizes the upper bound on the expected error:

$$\mathcal{E}(\hat{h}_i) \leq \hat{\mathcal{E}}_n(\hat{h}_i) + \sqrt{\frac{d_i(\log(2n/d_i)+1)+\log(1/(4\delta))}{n}}$$

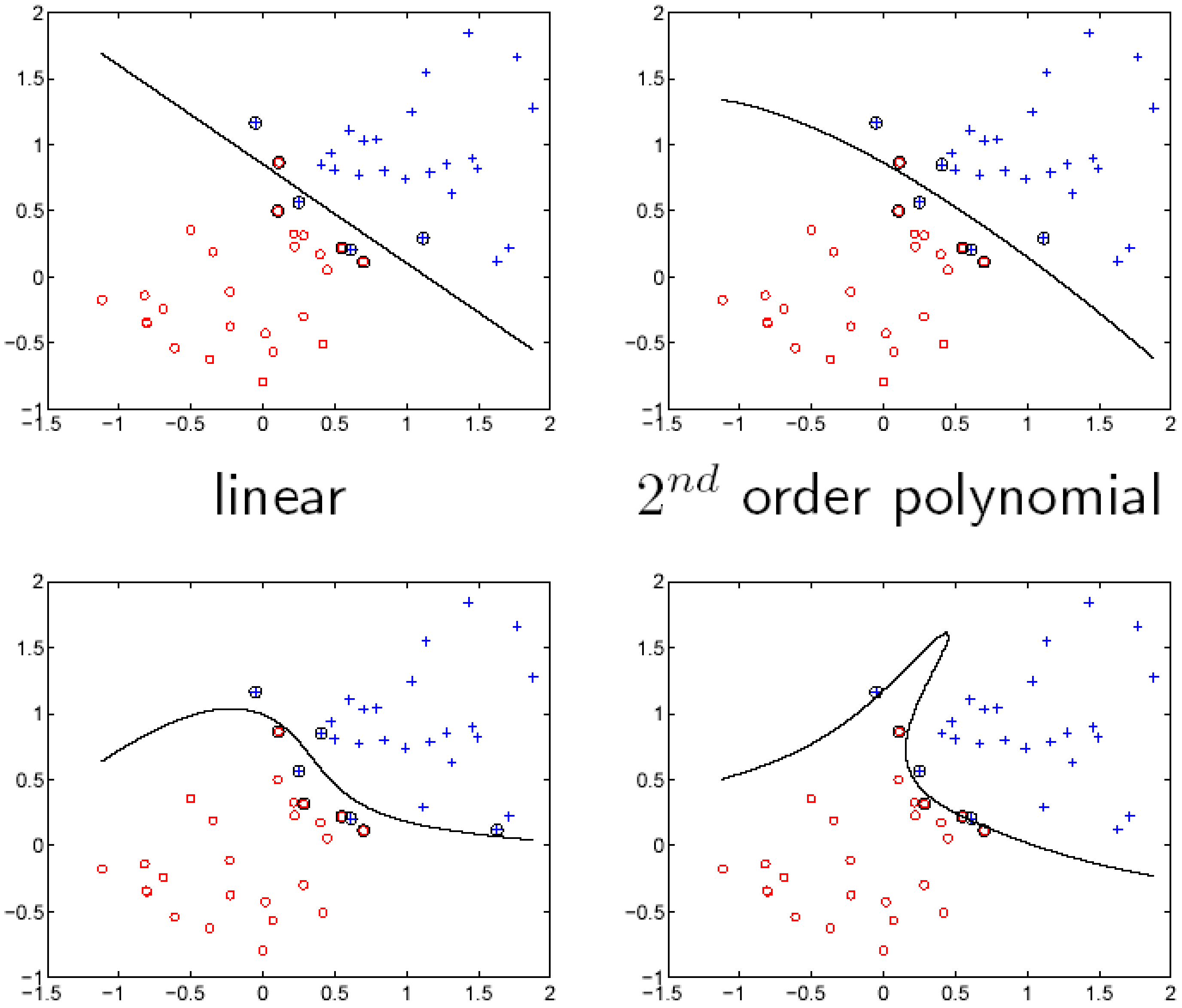
where $\hat{h}_i$ is the best classifier from F_i selected on the basis of the training set.



Structural Risk Minimization (3)



Structural risk minimization: example



linear

2^{nd} order polynomial

4^{th} order polynomial

8^{th} order polynomial

Structural risk minimization: example cont'd

- Number of training examples $n = 50$, confidence parameter $\delta = 0.05$.

Model	d_{VC}	Empirical fit	Complexity penalty $\epsilon(n, \delta, d_{VC})$
1^{st} order	3	0.06	0.5501
2^{nd} order	6	0.06	0.6999
4^{th} order	15	0.04	0.9494
8^{th} order	45	0.02	1.2849

- Structural risk minimization would select the simplest (linear) model in this case.

Structural Risk Minimization (1)

- **Why is the VC dimension relevant?**
  - Because the VC dimension provides bounds on the expected risk as a function of the empirical risk and the number of available examples
  - It can be shown that, with probability 1-η, the following bound holds

$$R(f) \leq R_{emp}(f) + \underbrace{\sqrt{\frac{h(\ln(2N/h)+1)-\ln(\eta/4)}{N}}}_{\text{VC confidence}} \qquad \textbf{Eq. (1)}$$

    - where h is the VC dimension of f(α), N is the number of training examples, and N>h
  - As the ratio N/h gets larger, the VC confidence becomes smaller and the actual risk becomes closer to the empirical risk
    - Therefore, this expression is consistent with the intuition that ERM is only suitable when sufficient data is available
  - This and other results are part of the field known as **Statistical Learning Theory** or Vapnik-Chervonenkis Theory, from which Support Vector Machines originated

Cross Validation

- To estimate generalization error, we need data unseen during training. We can use
  - Separate validation data when data is aboundant
    - Training set (50%)
    - Validation set (25%)
    - Test (publication) set (25%)
  - k-fold cross validation or leave-one-out cross validation when data is small
- Resampling methods when there is few data

Alternatives to VC-dim-based model selection

- What could we do instead of the scheme below?
    1. Cross-validation

i	f_i	TRAINER R	10-FOLD-CV-ERR	Choice
1	f_1			
2	f_2			
3	f_3			
4	f_4			
5	f_5			
6	f_6			

Using VC-dimensionality

People have worked hard to find VC-dimension for..

- Decision Trees
- Perceptrons
- Neural Nets
- Decision Lists
- Support Vector Machines
- And many many more

All with the goals of:

1. Understanding which learning machines are more or less powerful under which circumstances
2. Using Structural Risk Minimization to choose the best learning machine

The VC dimension in practice

- **Unfortunately, computing an upper bound on the expected risk is not practical in various situations**
  - The VC dimension cannot be accurately estimated for non-linear models such as neural networks
  - Implementation of Structural Risk Minimization may lead to a non-linear optimization problem
  - The VC dimension may be infinite (e.g., k=1 nearest neighbor), requiring infinite amount of data
  - The upper bound may sometimes be trivial (e.g., larger than one)

- **Fortunately, Statistical Learning Theory can be rigorously applied in the realm of linear models**

What you should know

- The definition of a learning machine: $\boldsymbol{f(x,\alpha)}$
- The definition of Shattering
- Be able to work through simple examples of shattering
- The definition of VC-dimension
- Be able to work through simple examples of VC-dimension
- Structural Risk Minimization for model selection
- Awareness of other model selection methods

ALTERNATIVES

SKIP AFTER CROSS-VALIDATION

Alternatives to VC-dim-based model selection

- What could we do instead of the scheme below?
  1. Cross-validation

i	f_i	TRAINERR	10-FOLD-CV-ERR	Choice
1	f_1			
2	f_2			
3	f_3			
4	f_4			
5	f_5			
6	f_6			

Alternatives to VC-dim-based model selection

- What could we do instead of the scheme below?
  1. Cross-validation
  2. AIC (Akaike Information Criterion)

As the amount of data goes to infinity, AIC promises* to select the model that'll have the best likelihood for future data

*Subject to about a million caveats

$$\text{AICSCORE} = LL(Data \mid \text{MLE params}) - (\#\,\text{parameters})$$

i	f_i	LOGLIKE(TRAINERR)	#parameters	AIC	Choice
1	f_1				
2	f_2				
3	f_3				
4	f_4				
5	f_5				
6	f_6				

Alternatives to VC-dim-based model selection

- What could we do instead of the scheme below?
    1. Cross-validation
    2. AIC (Akaike Information Criterion)
    3. BIC (Bayesian Information Criterion)

As the amount of data goes to infinity, BIC promises* to select the model that the data was generated from. More conservative than AIC.

*Another million caveats

$$\text{BICSCORE} = LL(Data \mid \text{MLE params}) - \frac{\#\,\text{params}}{2}\log R$$

i	f_i	LOGLIKE(TRAINERR)	#parameters	BIC	Choice
1	f_1				
2	f_2				
3	f_3				
4	f_4				
5	f_5				
6	f_6				

Which model selection method is best?

1. (CV) Cross-validation
2. AIC (Akaike Information Criterion)
3. BIC (Bayesian Information Criterion)
4. (SRMVC) Structural Risk Minimize with VC-dimension

- AIC, BIC and SRMVC have the advantage that you only need the training error.
- CV error might have more variance
- SRMVC is wildly conservative
- Asymptotically AIC and Leave-one-out CV should be the same
- Asymptotically BIC and a carefully chosen k-fold should be the same
- BIC is what you want if you want the best structure instead of the best predictor (e.g. for clustering or Bayes Net structure finding)
- Many alternatives to the above including proper Bayesian approaches.
- It's an emotional issue.

Extra Comments

- Beware: that second "VC-confidence" term is usually very very conservative (at least hundreds of times larger than the empirical overfitting effect).
- An excellent tutorial on VC-dimension and Support Vector Machines

    C.J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998. http://citeseer.nj.nec.com/burges98tutorial.html