

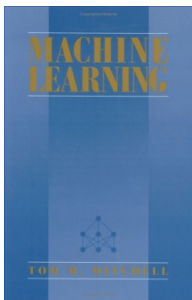
Outline

- What is Computational Learning Theory?
- Study case: same easy classifiers...
- Vapnik Chervonenkis dimension (VCdim)
- Application to NNs
- Case: generalization



Comparison Inductive to Analytic Learning

	<u>Inductive learning</u>	<u>Analytical learning</u>
Goal:	Hypothesis fits data	Hypothesis fits domain theory
Justification:	Statistical inference	Deductive inference
Advantages:	Requires little prior knowledge	Learns from scarce data
Pitfalls:	Scarce data, incorrect bias	Imperfect domain theory

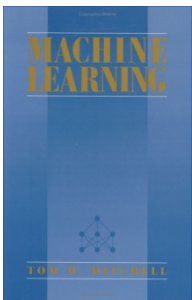


Computational Learning Theory

What general laws constrain inductive learning?

We seek theory to relate:

- **Complexity** of hypothesis space low complexity and training samples
- **Number** of training examples
- **Generalizations** to which degree untrained data is hit?
- **Probability** of successful learning
- **Accuracy** to which target concept is approximated



VC-dimension for characterizing classifiers

Note to other teachers and users of these slides. Andrew would be delighted if you found this source material useful in giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. PowerPoint originals are available. If you make use of a significant portion of these slides in your own lecture, please include this message, or the following link to the source repository of Andrew's tutorials: <http://www.cs.cmu.edu/~awm/tutorials>. Comments and corrections gratefully received.

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University

www.cs.cmu.edu/~awm

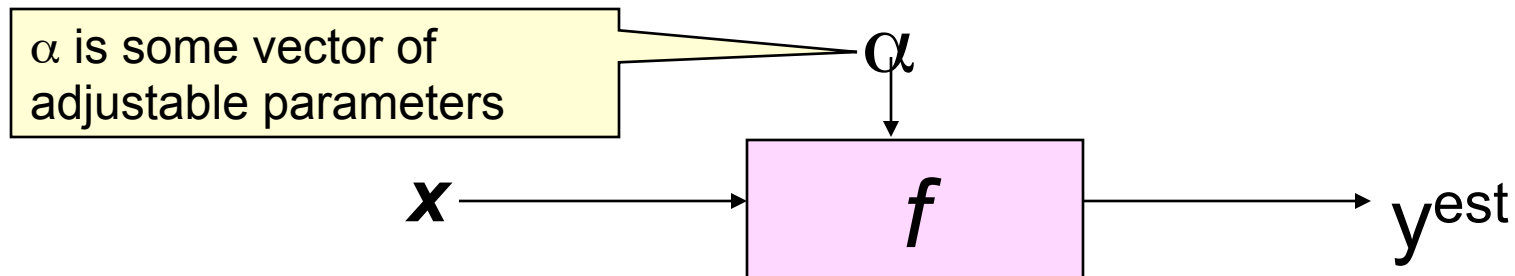
awm@cs.cmu.edu

412-268-7599

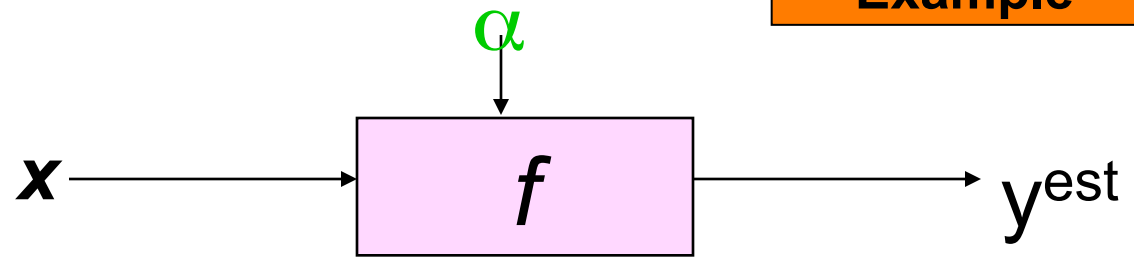


A learning machine

- A learning machine f takes an input \mathbf{x} and transforms it, somehow using weights α , into a predicted output $y^{est} = +/- 1$

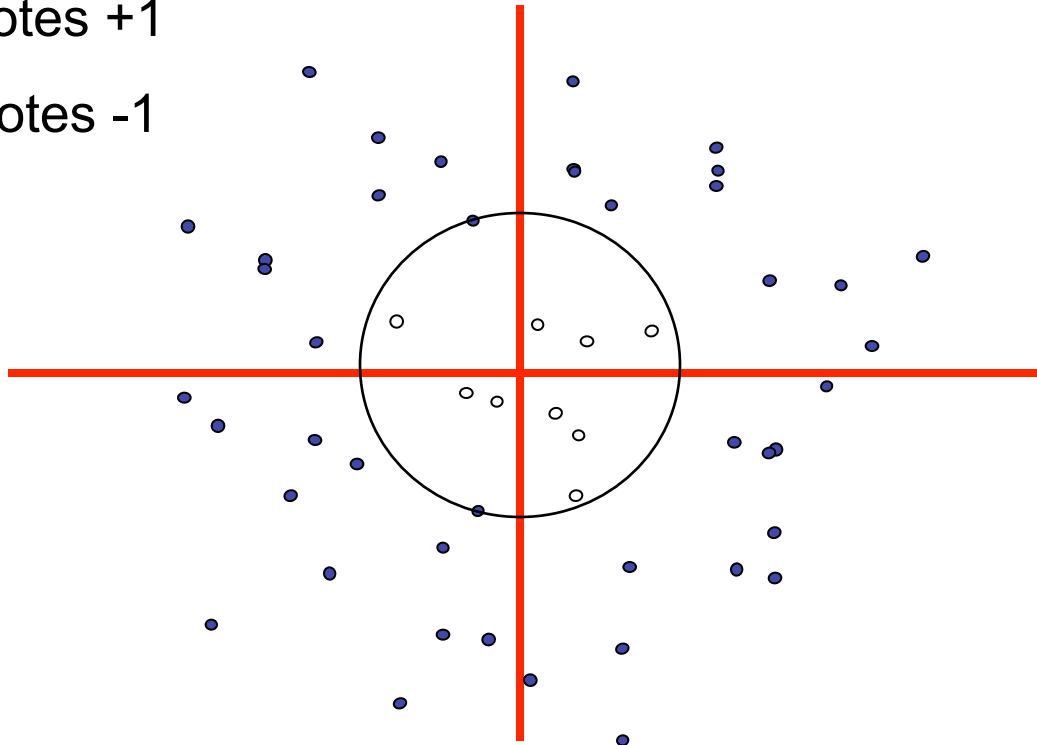


Examples

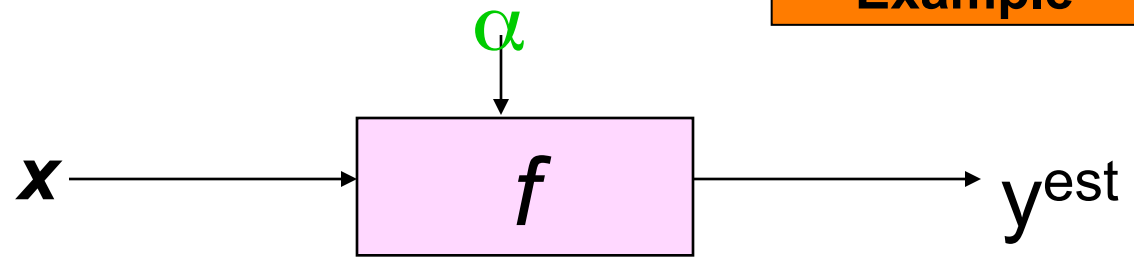


$$f(\mathbf{x}, \mathbf{b}) = \text{sign}(\mathbf{x} \cdot \mathbf{x} - \mathbf{b})$$

- denotes +1
- denotes -1

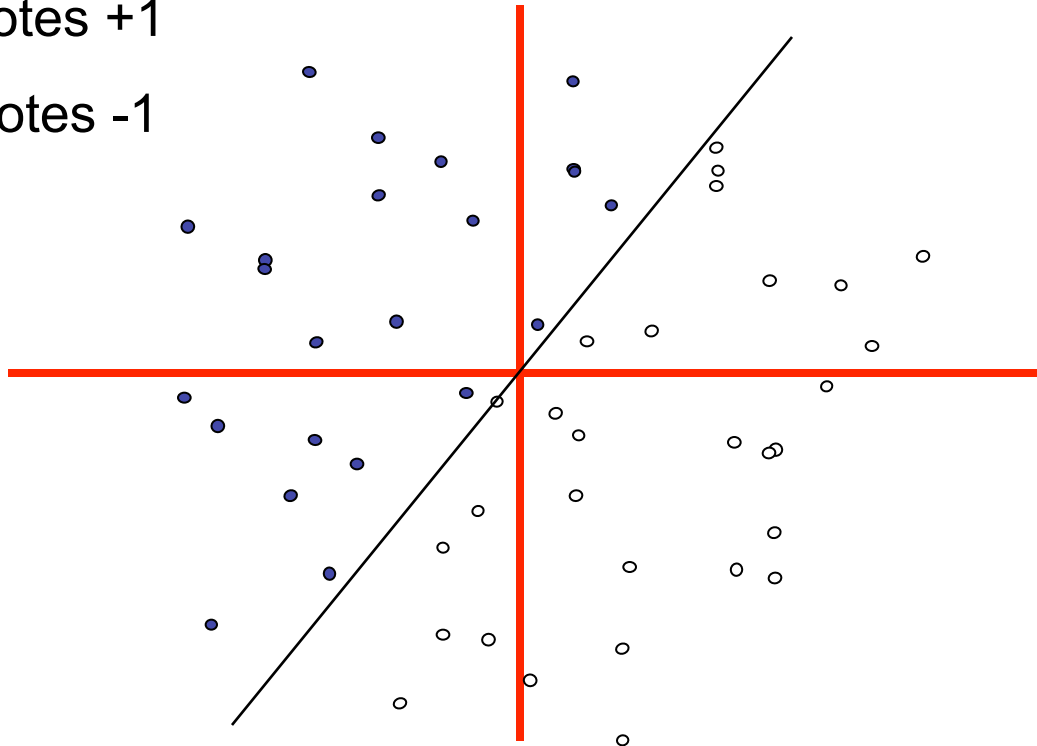


Examples

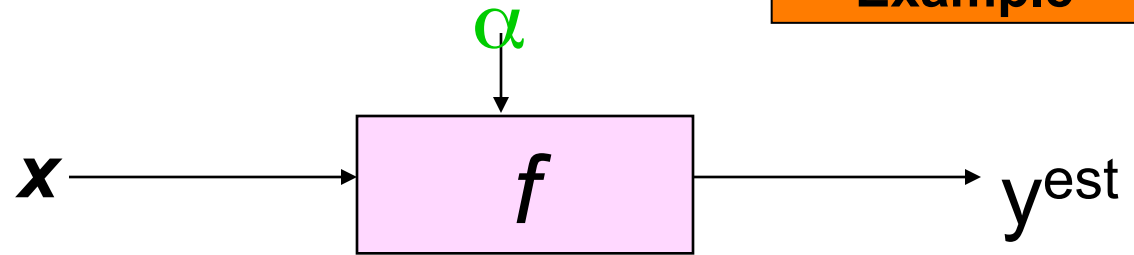


$$f(\mathbf{x}, \mathbf{w}) = \text{sign}(\mathbf{x} \cdot \mathbf{w})$$

- denotes +1
- denotes -1

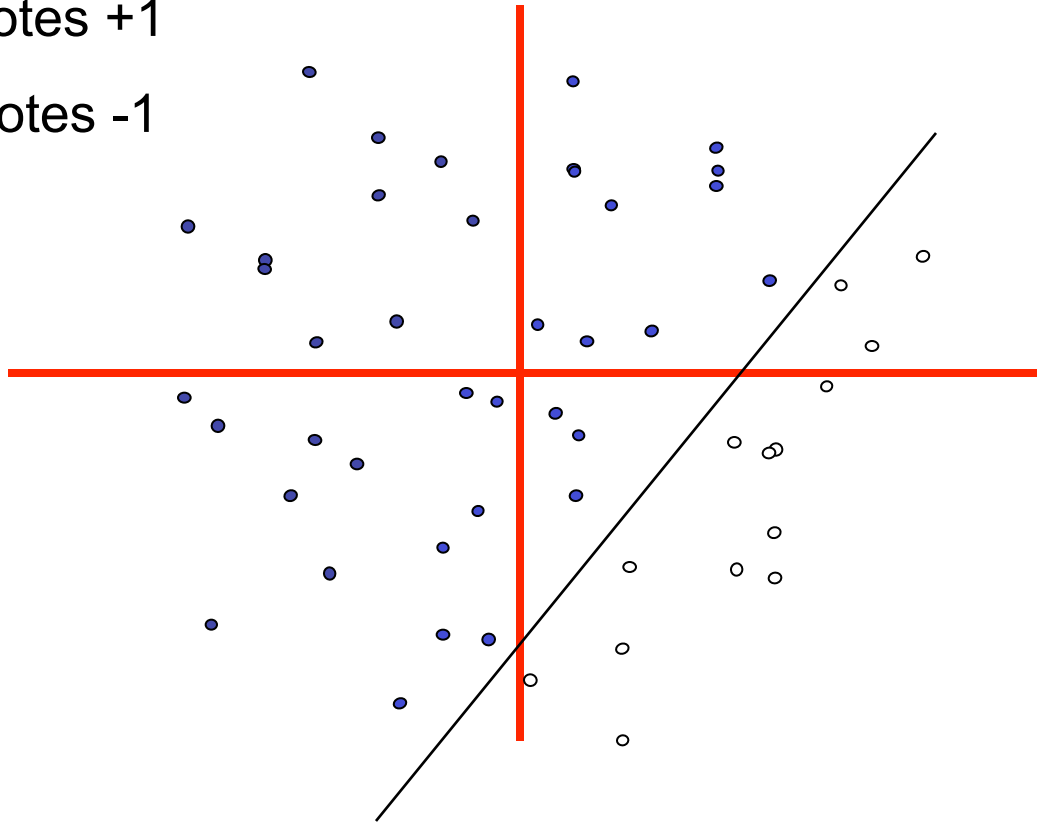


Examples



$$f(\mathbf{x}, \mathbf{w}, \mathbf{b}) = \text{sign}(\mathbf{x} \cdot \mathbf{w} + \mathbf{b})$$

- denotes +1
- denotes -1



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?



Some definitions

- Given some machine \mathbf{f}
- And under the assumption that all training points (x_k, y_k) were drawn i.i.d from some distribution.
- And under the assumption that future test points will be drawn from the same distribution
- Define

$$R(\alpha) = \text{TESTERR}(\alpha) = E\left[\frac{1}{2}|y - f(x, \alpha)|\right] = \begin{array}{l} \text{Probability of} \\ \text{Misclassification} \end{array}$$

Official terminology

Terminology we'll use



Some definitions

- Given some machine \mathbf{f}
- And under the assumption that all training points (x_k, y_k) were drawn i.i.d from some distribution.
- And under the assumption that future test points will be drawn from the same distribution
- Define

$$R(\alpha) = \text{TESTERR}(\alpha) = E \left[\frac{1}{2} |y - f(x, \alpha)| \right] = \begin{matrix} \text{Probability of} \\ \text{Misclassification} \end{matrix}$$

Official terminology

Terminology we'll use

$$R^{emp}(\alpha) = \text{TRAINERR}(\alpha) = \frac{1}{R} \sum_{k=1}^R \frac{1}{2} |y_k - f(x_k, \alpha)| = \begin{matrix} \text{Fraction Training} \\ \text{Set misclassified} \end{matrix}$$

R = #training set
data points



Vapnik-Chervonenkis dimension

$$\text{TESTERR}(\alpha) = E\left[\frac{1}{2}|y - f(x, \alpha)|\right] \quad \text{TRAINERR}(\alpha) = \frac{1}{R} \sum_{k=1}^R \frac{1}{2}|y_k - f(x_k, \alpha)|$$

- Given some machine \mathbf{f} , let h be its VC dimension.
- h is a measure of \mathbf{f} 's power (h does not depend on the choice of training set)
- Vapnik showed that with probability $1-\eta$

$$\text{TESTERR}(\alpha) \leq \text{TRAINERR}(\alpha) + \sqrt{\frac{h(\log(2R/h) + 1) - \log(\eta/4)}{R}}$$

This gives us a way to estimate the error on future data based only on the training error and the VC-dimension of \mathbf{f}



What VC-dimension is used for

$$\text{TESTERR}(\alpha) = E \left[\frac{1}{2} |y - f(x, \alpha)| \right] \quad \text{TRAINERR}(\alpha) = \frac{1}{R} \sum_{k=1}^R \frac{1}{2} |y_k - f(x_k, \alpha)|$$

- Given some machine \mathbf{f} , let h be its VC dimension
- h is a measure of \mathbf{f} 's power
- Vapnik showed that

But given machine \mathbf{f} ,
how do we define
and compute h ?

$$\text{TESTERR}(\alpha) \leq \frac{\text{TRAINERR}(\alpha) + \frac{h}{R} \log(\eta/4)}{R}$$

... way to estimate the error on
future data based only on the training error
and the VC-dimension of \mathbf{f}



Shattering

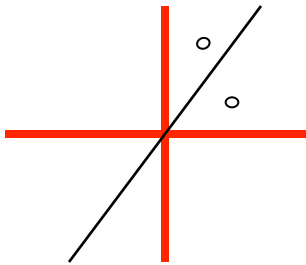
- Machine f can *shatter* a set of points $x_1, x_2 \dots x_r$ if and only if...
For every possible training set of the form $(x_1, y_1), (x_2, y_2), \dots (x_r, y_r)$
...There exists some value of α that gets zero training error.

There are 2^r such training sets to consider, each with a different combination of +1's and -1's for the y 's



Shattering

- Machine f can *shatter* a set of points $x_1, x_2 \dots x_r$ if and only if...
For every possible training set of the form $(x_1, y_1), (x_2, y_2), \dots (x_r, y_r)$
...There exists some value of α that gets zero training error.
- Question: Can the following f shatter the following points?

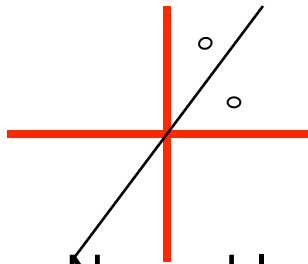


$$f(x, \mathbf{w}) = \text{sign}(x \cdot \mathbf{w})$$



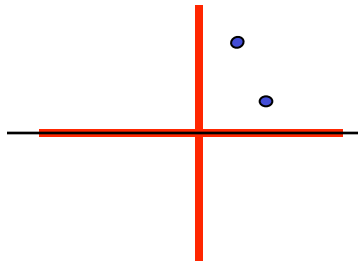
Shattering

- Machine f can *shatter* a set of points $x_1, x_2 \dots x_r$ if and only if...
For every possible training set of the form $(x_1, y_1), (x_2, y_2), \dots (x_r, y_r)$
...There exists some value of α that gets zero training error.
- Question: Can the following f shatter the following points?

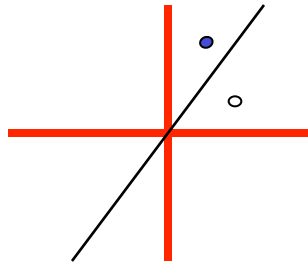


$$f(x, w) = \text{sign}(x \cdot w)$$

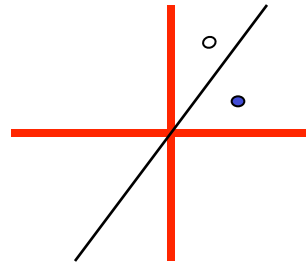
- Answer: No problem. There are four training sets to consider



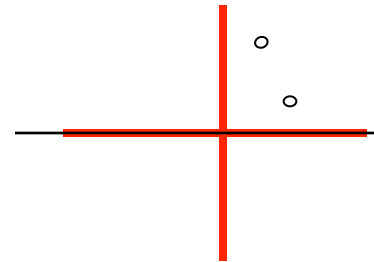
$$w = (0, 1)$$



$$w = (-2, 3)$$



$$w = (2, -3)$$

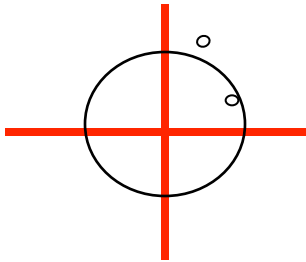


$$w = (0, -1)$$



Shattering

- Machine f can *shatter* a set of points $x_1, x_2 \dots x_r$ if and only if...
For every possible training set of the form $(x_1, y_1), (x_2, y_2), \dots (x_r, y_r)$
...There exists some value of α that gets zero training error.
- Question: Can the following f shatter the following points?

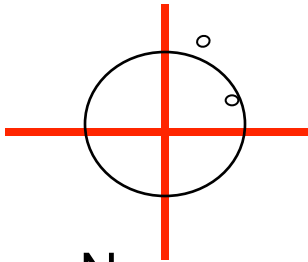


$$f(x, b) = \text{sign}(x \cdot x - b)$$



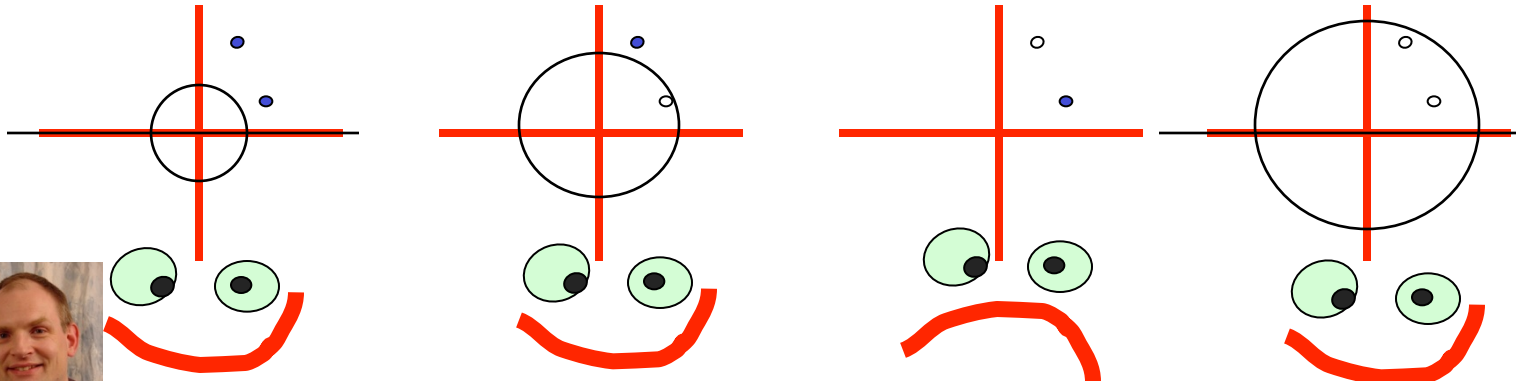
Shattering

- Machine f can *shatter* a set of points $x_1, x_2 \dots x_r$ if and only if...
For every possible training set of the form $(x_1, y_1), (x_2, y_2), \dots (x_r, y_r)$
...There exists some value of α that gets zero training error.
- Question: Can the following f shatter the following points?



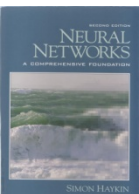
$$f(x, b) = \text{sign}(x \cdot x - b)$$

- Answer: No way my friend.



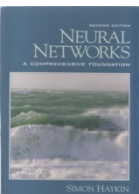
Definition Shatters == separates

- Given some (fixed) **dataset $A \subseteq X$** with **N points** in X , where X is called the **instance space**.
- Then these N points, when labeled “+” or “-”, **build 2^N binary maps**
 \Rightarrow i.e. we have 2^N different learning problems
- If for any of these problems, we can find a hypothesis **$h \in \mathcal{H}$ (i.e. an α for f)** that separates the positive examples from the negative, then we say **\mathcal{H} shatters A** .
- That is, **any learning** problem **definable by N data points from X** can be learned with **no error** by a hypothesis **h from \mathcal{H}** .



Definition VCdim

- The **size of the largest finite subset $A \subseteq X$** that can be shattered by \mathcal{H} is called the **Vapnik-Chervonenkis (VC) dimension** of \mathcal{H} (denoted as **$VC(\mathcal{H})$**)
- It measures the *capacity* of the hypothesis class \mathcal{H} over the instance space X .
- If there is no such number for a finite A then **$VC(\mathcal{H}) = \infty$** .
- Note that if the VC dimension is m then there **exists at least one set of some m points ($\subseteq X$)** that can be shattered
- But in general it will **not be true that every set of m points can be shattered**.



Definition of VC dimension

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: What's VC dimension of $f(x, b) = \text{sign}(x \cdot x - b)$



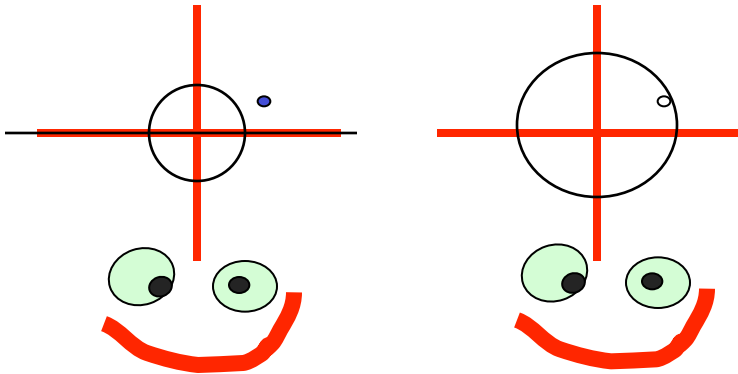
VC dim of trivial circle

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: What's VC dimension of $f(x, b) = \text{sign}(x \cdot x - b)$

Answer = 1: we can't even shatter two points! (but it's clear we can shatter 1)



Reformulated circle

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

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



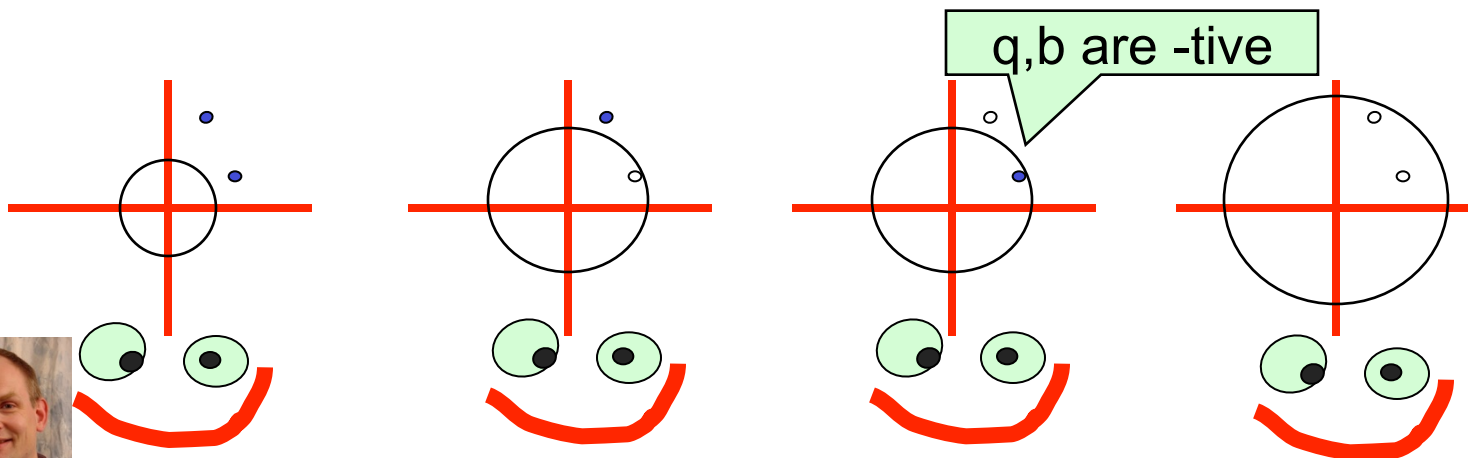
Reformulated circle

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

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

- Answer = 2



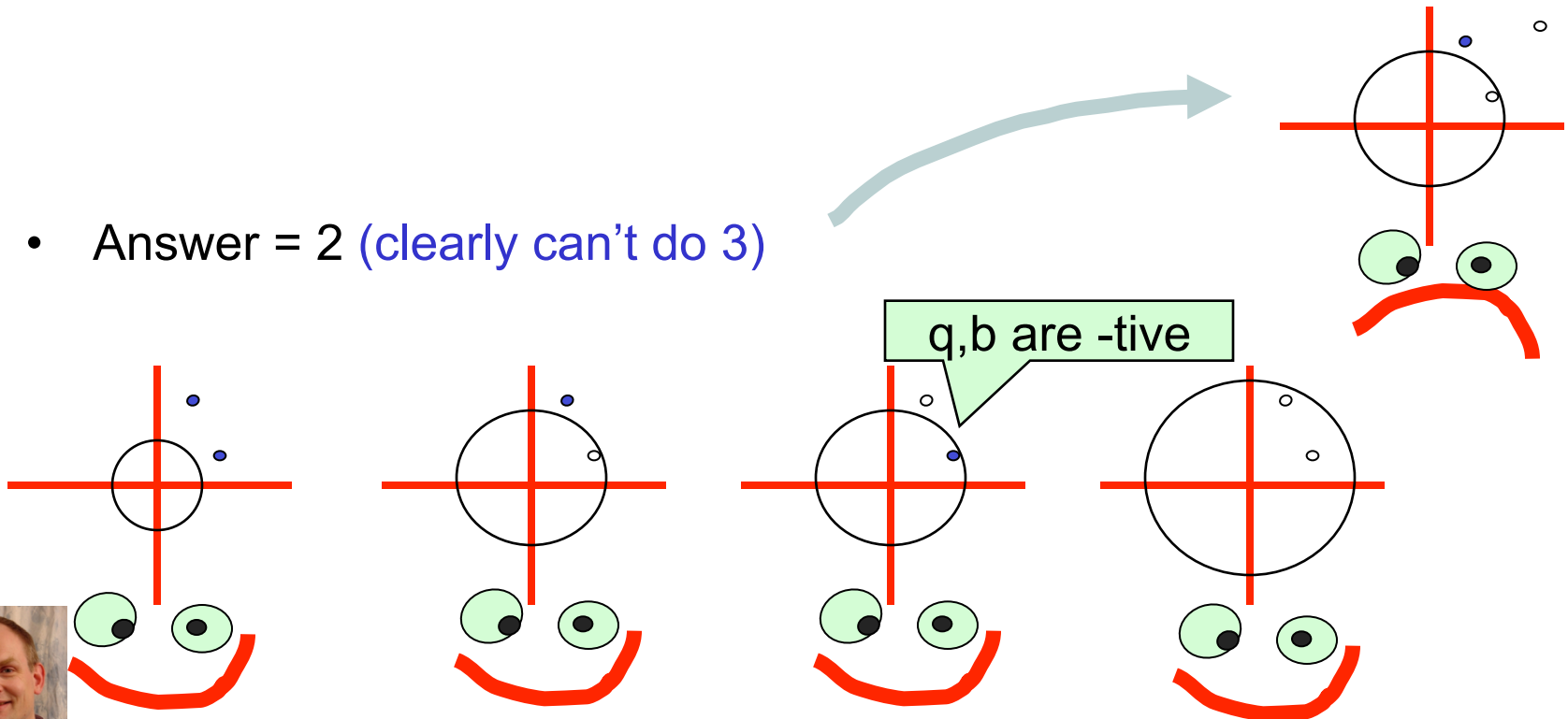
Reformulated circle

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: What's VC dimension of $f(x, q, b) = \text{sign}(qx \cdot x - b)$

- Answer = 2 (clearly can't do 3)



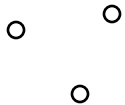
VC dim of separating line

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: For 2-d inputs, what's VC-dim of $f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot x + b)$?

Well, can f shatter these three points?



VC dim of line machine

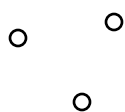
Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: For 2-d inputs, what's VC-dim of $f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot x + b)$?

Well, can f shatter these three points?

Yes, of course.



All -tive or all +sitive is trivial

One +sitive can be picked off by a line

One -tive can be picked off too.



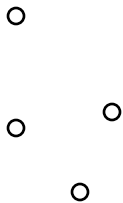
VC dim of line machine

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: For 2-d inputs, what's VC-dim of $f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot x + b)$?

Well, can we find four points that f can shatter?



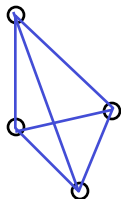
VC dim of line machine

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: For 2-d inputs, what's VC-dim of $f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot x + b)$?

Well, can we find four points that f can shatter?



Can always draw six lines between pairs of four points.



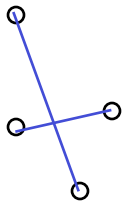
VC dim of line machine

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: For 2-d inputs, what's VC-dim of $f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot x + b)$?

Well, can we find four points that f can shatter?



Can always draw six lines between pairs of four points.

Two of those lines will cross.



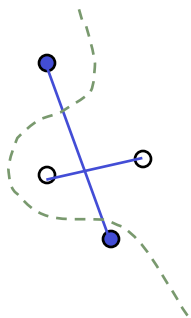
VC dim of line machine

Given machine f , the VC-dimension h is

The maximum number of points that can be arranged so that f shatters them.

Example: For 2-d inputs, what's VC-dim of $f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot x + b)$?

Well, can we find four points that f can shatter?



Can always draw six lines between pairs of four points.

Two of those lines will cross.

If we put points linked by the crossing lines in the same class they can't be linearly separated

So a line can shatter 3 points but not 4

So VC-dim of Line Machine is 3



VC dim of linear classifiers in m-dimensions

If input space is m-dimensional and if \mathbf{f} is $\text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$, what is the VC-dimension h ?

linear - hyperplane

Proof that $h \geq m$: Show that m points can be shattered

Can you guess how?



VC dim of linear classifiers in m-dimensions

If input space is m-dimensional and if \mathbf{f} is $\text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$, what is the VC-dimension h ?

Proof that $h \geq m$: Show that m points can be shattered

Define m input points thus:

$$\mathbf{x}_1 = (1, 0, 0, \dots, 0) \quad \text{basis vectors}$$

$$\mathbf{x}_2 = (0, 1, 0, \dots, 0)$$

\vdots

$$\mathbf{x}_m = (0, 0, 0, \dots, 1) \quad \text{So } x_k[j] = 1 \text{ if } k=j \text{ and } 0 \text{ otherwise}$$

Let y_1, y_2, \dots, y_m , be any one of the 2^m combinations of class labels.

Guess how we can define $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m$ and b to ensure $\text{sign}(\mathbf{w} \cdot \mathbf{x}_k + b) = y_k$ for all k ? **Note:**

$$\text{sign}(\mathbf{w} \cdot \mathbf{x}_k + b) = \text{sign}\left(b + \sum_{j=1}^m w_j \cdot x_k[j]\right)$$



VC dim of linear classifiers in m-dimensions

If input space is m-dimensional and if \mathbf{f} is $\text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$, what is the VC-dimension h ?

Proof that $h \geq m$: Show that m points can be shattered

Define m input points thus:

$$\mathbf{x}_1 = (1, 0, 0, \dots, 0)$$

$$\mathbf{x}_2 = (0, 1, 0, \dots, 0)$$

\vdots

$$\mathbf{x}_m = (0, 0, 0, \dots, 1) \quad \text{So } x_k[j] = 1 \text{ if } k=j \text{ and } 0 \text{ otherwise}$$

Let y_1, y_2, \dots, y_m , be any one of the 2^m combinations of class labels.

Guess how we can define w_1, w_2, \dots, w_m and b to ensure $\text{sign}(\mathbf{w} \cdot \mathbf{x}_k + b) = y_k$ for all k ? **Note:**

Answer: $b=0$ and $w_k = y_k$ for all k .

$$\text{sign}(\mathbf{w} \cdot \mathbf{x}_k + b) = \text{sign}\left(b + \sum_{j=1}^m w_j \cdot x_k[j]\right)$$

VC dim of linear classifiers in m-dimensions

If input space is m-dimensional and if \mathbf{f} is $\text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$, what is the VC-dimension h ?

- Now we know that $h \geq m$
- In fact, $h = m + 1$
- Proof that $h \geq m + 1$ is easy
- Proof that $h < m + 2$ is moderate



What does VC-dim measure?

- Is it the number of parameters?

Related but not really the same.

- I can create a machine with one numeric parameter that really encodes 7 parameters (How?)
- And I can create a machine with 7 parameters which has a VC-dim of 1 (How?)
- *Andrew's private opinion: it often is the number of parameters that counts.*

























Structural Risk Minimization

- Let $\varphi(f)$ denote the set of functions representable by f .
- Suppose $\varphi(f_1) \subseteq \varphi(f_2) \subseteq \dots \varphi(f_n)$
- Then $h(f_1) \leq h(f_2) \leq \dots h(f_n)$ (Hey, can you formally prove this?)
- We're trying to decide which machine to use.
- We train each machine and make a table...

higher dimensionality of the functions leads to overfitting as the function will now be able to pass through all the data points perfectly.

$$\text{TESTERR}(\alpha) \leq \text{TRAINERR}(\alpha) + \sqrt{\frac{h(\log(2R/h) + 1) - \log(\eta/4)}{R}}$$

i	f_i	TRAINER R	VC-Conf	Probable upper bound on TESTERR	Choice
1	f_1				
2	f_2				
3	f_3				
4	f_4			 	
5	f_5			 	
6	f_6			 	



Using VC-dimensionality

That's what VC-dimensionality is about

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 for to choose the best learning machine



Application to aNNs

- aNNs realize learning machines as functions f .
- three theorems give bounds on VC dim:

- network \mathcal{N} with a single linear neuron
 $\Rightarrow VCdim(\mathcal{N})=m+1$ (see above)

- \mathcal{N} as an arbitrary feedforward with McCulloch Pitts neurons (Cover (1968), Baum and Haussler (1989))
 $\Rightarrow VCdim(\mathcal{N})=O(W \log_2 W)$ (W ==free parameters)

- \mathcal{N} feedforward multilayer sigmoids (Koiran and Sontag (1996))
 $\Rightarrow VCdim(\mathcal{N})=O(W^*W)$ (W ==free parameters)



VC-dimension of directed acyclic layered networks

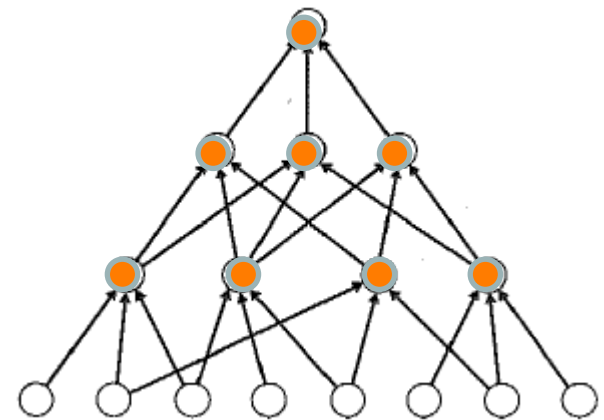
- Let G be a layered directed acyclic graph with n input nodes and $s \geq 2$ internal nodes, each having at most r inputs.

Let C be a concept class over \mathbb{R}^r of VC dimension d , corresponding to the set of functions that can be described by each of the s internal nodes.

Let C_g be the G -composition of C , corresponding to the set of functions that can be represented by G .

\Rightarrow Then $VCdim(C_g) \leq 2ds \log_2(es)$,
where e is the base of the natural logarithm.

(See Kearns and Vazirani 1994.)



Number of examples needed

- We can in general bound the number m of training examples sufficient to learn (with probability at least $(1 - \delta)$) any target concept from C to within error ε . Using the VCdim of the hypothesis space H it was found [Blumer et.al 1989]:

$$m \geq \frac{1}{\varepsilon} (4 \log_2(\frac{2}{\delta}) + 8VCdim(H) \log_2(\frac{13}{\varepsilon}))$$

- For a $H =$ single perceptron with r -inputs we have: $VCdim(H) = r+1$.

$$m \geq \frac{1}{\varepsilon} (4 \log_2(\frac{2}{\delta}) + 8(r+1) \log_2(\frac{13}{\varepsilon}))$$

- For $H = C^{Perceptrons}_G$ we find

$$m \geq \frac{1}{\varepsilon} (4 \log_2(\frac{2}{\delta}) + 16(r+1)s \log_2(\exp(1)s) \log_2(\frac{13}{\varepsilon}))$$























has to go through a huge number of training data examples to train just a small number of parameters.



Alternatives to VC-dim-based model selection

- What could we do instead of the scheme below?














$$\text{TESTERR}(\alpha) \leq \text{TRAINERR}(\alpha) + \sqrt{\frac{h(\log(2R/h) + 1) - \log(\eta/4)}{R}}$$

i	f_i	TRAINER R	VC-Conf	Probable upper bound on TESTERR	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?
 - 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			



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

$$\text{AICSCORE} = \underbrace{LL(\text{Data} \mid \text{MLE params})}_{\text{log-likelihood}} - (\# \text{ parameters})$$

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



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.

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

*Another million caveats

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

'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 (which we’ll be studying soon):

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>



Example on Generalization

- **Binary context**, aNN with single **binary output**
- given $x \in X$, $P(x)$ random distribution on X , target function $h(x)$ (noiseless), aNN learned a function $y(x, w)$
- $g(y)$ is the **likelihood** that for a $\underline{x} \in X$ drawn from X according to $P(x)$ it holds:
 $h(\underline{x}) = y(\underline{x}, w)$
- $P(x)$ unknown \Rightarrow use $g_N(y)$ instead (i.e. count performance of y on finite set N)
Expect: $g_N(y) \rightarrow g(y) \quad (N \rightarrow \infty)$
- But: $y(x, w)$ it tuned to data set \Rightarrow expect $g_N(y) > g(y)$
- „it appears more general then it is“
- E.g. we might find $g_N(y) = 1$, yet $g(y) \ll 1$ on new examples
- $g_N(y)$ is a **biased estimate** of $g(y)$
- Let $\{y\}$ denote all functions that the aNN can realize.
- We are interested in worst case generalization performance:

$$\max_{\{y\}} | g_N(y) - g(y) |$$

...cont...

- Vapnik Chervonenkis 1971 give upper bound for exceeding some small error ε :

$$(*) \Pr\left(\max_{\{y\}} |g_N(y) - g(y)| > \varepsilon\right) \leq 4\Delta(2N)\exp(-\varepsilon^2 N / 8)$$

where $\Delta(N)$ is called growth function

- E.g. assume $g_N(y)=1$ (i.e. perfect training), ε given, let r.h.s.= $\delta=0.05$, then determine an N
- Using this N training patterns we will be 95% certain that $g(y) > 1-\varepsilon$.
- Now: how does Δ grow in comparison to $\exp(-\varepsilon^2 N)$?

...cont...

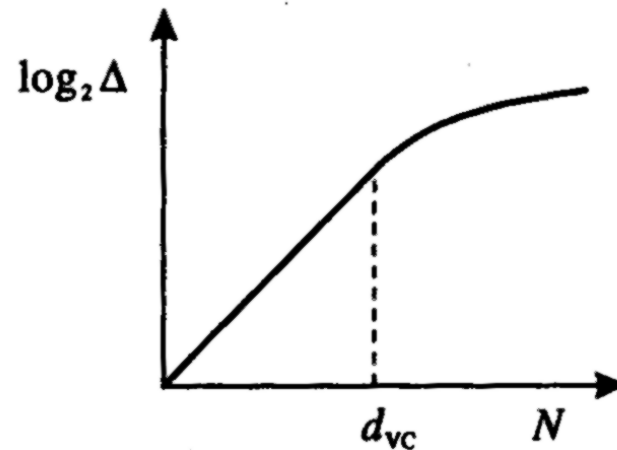


Figure 9.19. General form of the growth function $\Delta(N)$ shown as a plot of $\log_2 \Delta$ versus N . The function initially grows like 2^N up to some critical number of patterns, given by $N = d_{VC}$, at which point the growth slows to become a power law. The value d_{VC} is called the Vapnik–Chervonenkis dimension.

- $\Delta(N)$ is the number of distinct binary functions (dichotomies) which can be implemented in the aNN on N input vectors x_n , $n=1 \dots N$
- Potentially 2^N , if the aNN can do all $\Rightarrow \Delta(N)=2^N$
- For small $N \sim 2^N$, after that \Rightarrow slow down, critical number of patterns is called d_{VC} or just VC dimension
- Theorem: $\Delta(N)$ is bounded above by either 2^N or $N^{d_{VC}} + 1 \Rightarrow$ at most polynomial growth

Thus we can make (*) arbitrary small by making N sufficiently large
 d_{VC} depends on the particular network

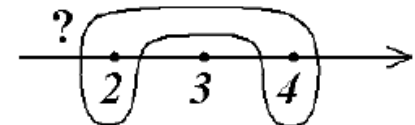
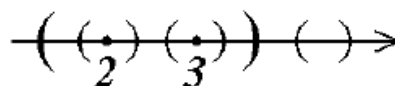
What you should know

- The definition of a learning machine: $f(\mathbf{x}, \alpha)$
- 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
- Complexity bounds for certain classes of aNNs



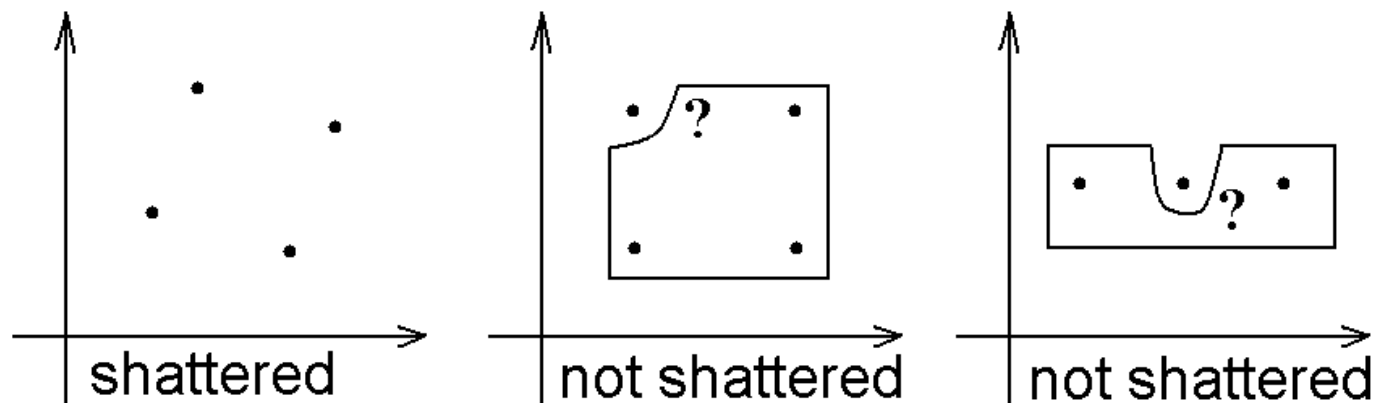
Example 1 shatter

- Let $X = \mathbb{R}$ and $\mathcal{H} = \{(a, b) : a < b\}$. Consider two sets: $A = \{2, 3\}$ and $A' = \{2, 3, 4\}$.
- It is easy to see that the class \mathcal{H} of all intervals shatters A and does not shatter A' .
- Indeed, we can obtain sets \emptyset , $\{2\}$, $\{3\}$, and $\{2, 3\}$ by intersecting A with intervals.
- However, for set A' , there is no interval that contains 2 and 4 and does not contain 3; therefore, subset $\{2, 4\}$ cannot be obtained, and so A' is not shattered by intervals.



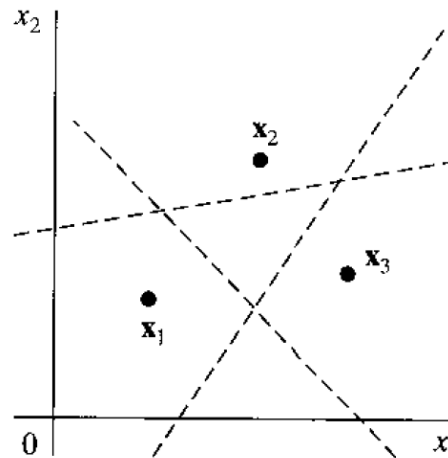
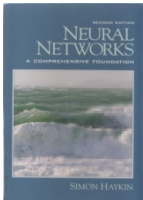
Example 2 shatter

- Let $X = \mathbb{R}^2$ and $\mathcal{H} = \text{Rect}(a_1, a_2, b_1, b_2) = \{ \langle x_1, x_2 \rangle : a_1 \leq x_1 \leq b_1, a_2 \leq x_2 \leq b_2 \}$,
- \mathcal{H} is the collection of axis-aligned rectangles in \mathbb{R}^2 .

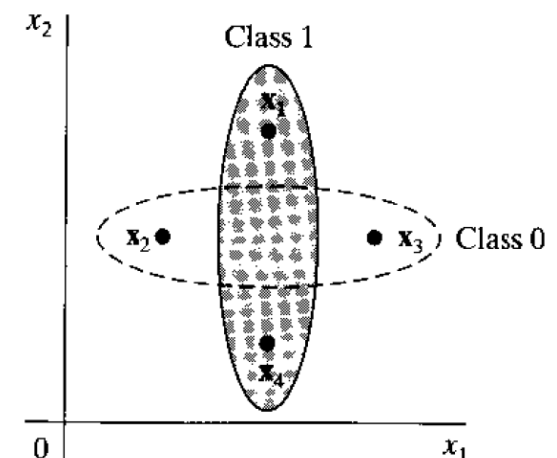


Example 3 shatter

- M-dim. input vectors \mathcal{H} , b bias, \mathbf{w} weight
- \mathcal{F} : $y = \phi(\mathbf{w}^T \mathbf{x} + b)$
- ϕ threshold function
- $VCdim(\mathcal{F}) = m + 1$



(a)



(b)

Example 4 VCdim

- Let $X = \mathbb{R}$ and $\mathcal{H} = \{(a, b) : a < b\}$ then $VCdim(\mathcal{H}) = 2$.
- Indeed, as we saw in Example 1, there is a set of size 2 that is shattered. Now, take any set A of 3 or more points;
- Then A contains three points $x_1 < x_2 < x_3$.
- The set $B = A \cap ((-\infty, x_1] \cup [x_3, \infty))$ cannot be obtained by intersecting A with an interval, because any interval that contains x_1 and x_3 must also contain x_2 , but $x_2 \notin B$.



Example 5 VCdim

- Let $X = \mathbb{R}^2$ and let \mathcal{H} be the collection of axis-aligned rectangles in \mathbb{R}^2 ; then $\text{VCdim}(\mathcal{H}) = 4$.
- We saw in Example 2 that $\text{VCdim}(\mathcal{H}) \geq 4$.
- Consider any set $A \subset \mathbb{R}^2$ of size 5 or more, and take a five-point subset $C \subseteq A$.
- In C , take a leftmost point (whose first coordinate is the smallest in C), a rightmost point (first coordinate is the largest), a lowest point (second coordinate is the smallest), and a highest point (second coordinate is the largest); let $x \in C$ be the (fifth) point that was not selected.
- Now, define $B = A \setminus \{x\}$. It is impossible to make B by intersecting A with an axis-aligned rectangle.
- Indeed, such a rectangle must contain all four selected points in C ; but in this case the rectangle contains x as well, because its coordinates are within the intervals spanned by selected points.
- So, A is not shattered by \mathcal{H} , and therefore $\text{VCdim}(\mathcal{H}) = 4$.



Example 6 VCdim

- Let $X = \mathbb{R}$ and \mathcal{H} be the collection of all finite subsets of \mathbb{R} .
- Then $\text{VCdim}(\mathcal{H}) = \infty$, because any finite set can be shattered by \mathcal{H} .

