Computational Learning Theory

Inductive Learning

- So far, we have been talking about inductive learning.
- Can be defined as learning the class separating function by learning from labeled training data.
- We have looked at training error, test error (true error) and various learning techniques.
- Now it's time to see the relationship between training sample size, number of hypotheses, and the true error rate.
- This field is called Computational Learning Theory (CLT).
- Let's look at binary classification using Boolean attributes.

Binary Classification

- **Fundamental Question: Predict Error Rates**
  - Given:
    - The instance space X
      - e.g., Possible days, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
    - A target function (or concept) f: X → {0,1}
      - E.g., f: *EnjoySport* → {0,1}
    - The space H of hypotheses
      - E.g., conjunctions of literals: <?,*Cold*,*High*,?,?,?>
    - A set of training examples S (containing positive and negative examples of the target function)

      $\langle \mathbf{x}_1, f(\mathbf{x}_1) \rangle, \ldots, \langle \mathbf{x}_m, f(\mathbf{x}_m) \rangle$
  - Find:
    - A hypothesis $h \in H$ such that $h(\mathbf{x}) = f(\mathbf{x}) \; \forall \, \mathbf{x} \in S$

Some key assumptions in CLT

- Training examples are noise free
- All attributes are discrete or Boolean (not continuous)
- Output hypothesis is in the form of a logical function of attributes i.e. $f = x1 \wedge x2 \wedge \neg x3$
- There is one hypothesis that correctly classifies the training examples

Some key results

- If there are n Boolean attributes, there exist:

$|X| = 2^n$ possible instances and

Each of instances can be labeled in 2 ways (0 or 1), so total number of ways of labeling them are:

$|H| = 2^{2^n}$ possible hypotheses

Sample Complexity

How many training examples are sufficient to learn the target concept?

1. If learner proposes instances as queries to teacher *(active learning)*
   - Learner proposes **x**, teacher provides $f(\mathbf{x})$
2. If teacher (who knows f) provides training examples
   - Teacher provides example sequence $\langle \mathbf{x}, f(\mathbf{x}) \rangle$
3. If some random process (e.g., nature) proposes instances *(standard case in supervised learning)*
   - **x** generated randomly, teacher provides $f(\mathbf{x})$
4. If examples are given by an opponent (who knows f) *(on-line learning, mistake-bound model)*
   - (we won't cover this here)

Sample Complexity 1

- Student can ask for a particular data instance and teacher has to label it
- Learner proposes instance **x** and teacher provides f(**x**)
- What is the optimal strategy?
- Well, you have to choose instances that lead to a faster elimination of incorrect hypotheses
- Consider a 2 attribute case: $\mathbf{X} = (x1\ x2)^T$ and Boolean Y.

Sample Complexity 1

4 distinct instances

X1	X2
0	0
0	1
1	0
1	1

One out of the 16 possible hypotheses is correct. Let's say:

$$\begin{pmatrix}0\\1\\1\\0\end{pmatrix}$$

16 possible ways of labeling Y i.e. |H|

$$\begin{pmatrix}0\\0\\0\\0\end{pmatrix}\ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\ \begin{pmatrix}0\\0\\1\\1\end{pmatrix}$$

$$\begin{pmatrix}0\\1\\0\\0\end{pmatrix}\ \begin{pmatrix}0\\1\\0\\1\end{pmatrix}\ \begin{pmatrix}0\\1\\1\\0\end{pmatrix}\ \begin{pmatrix}0\\1\\1\\1\end{pmatrix}$$

$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\ \begin{pmatrix}1\\0\\0\\1\end{pmatrix}\ \begin{pmatrix}1\\0\\1\\0\end{pmatrix}\ \begin{pmatrix}1\\0\\1\\1\end{pmatrix}$$

$$\begin{pmatrix}1\\1\\0\\0\end{pmatrix}\ \begin{pmatrix}1\\1\\0\\1\end{pmatrix}\ \begin{pmatrix}1\\1\\1\\0\end{pmatrix}\ \begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Sample Complexity 1

- If you had the freedom to choose instances and get their labels from the teacher.
- What is the minimum number of instances that you need to see?

Sample Complexity 1

- If you had the freedom to choose instances and get their labels from the teacher.
- What is the minimum number of instances that you need to see?
- You can ask what is the label of (0 0). Teacher says 0
  How many hypotheses are eliminated? 8
- At each step, half of the hypotheses are eliminated:
  So number of questions needed: $\log_2|H|$

Sample Complexity 2

- If a teacher who knows f provides you the training examples
  Assuming teacher is friendly and wants to help you ☺
- In this case, you can't make a general statement about sample complexity. It depends on type of hypothesis learned.

Consider the case H = conjunctions of up to n Boolean literals and their negations

e.g., (*AirTemp* = *Warm*) ^ (*Wind* = *Strong*), where *AirTemp*,*Wind*, ... each have 2 possible values.

- if n possible Boolean attributes in H, n+1 examples suffice. Why?

Sample Complexity 2

- Consider the case where f = n Boolean literals
  For example f = x1 ^ ¬x2 ^ x3
- A teacher needs to show only 1 true instance AND shows an example, with each Boolean literal missing, where the example is false
- Example

x1	x2	x3	y
1	0	1	1
?	1	1	0
1	?	0	0
0	0	?	0

Hint: For a conjunction each term can be a positive or negative of an attribute.
Each such term or group of terms should be positive.
Take rows 1 and 2: We care only about x2 and x3. Value of x2 is changed to 1 and it causes output to be 0, so it should be in negation i.e. ¬x2
Take rows 1 and 3: We care only about x1 and x3. When value of x3 goes from 1 to 0, it causes output to be 0, so it should be a positive term i.e. x3

Sample Complexity 3

Given:

- set of instances X
- set of hypotheses H
- set of possible target concepts F
- training instances generated by a fixed, unknown probability distribution D over X

Sample Complexity 3

Learner observes a sequence D of training examples of form $\langle \mathbf{x}, f(\mathbf{x}) \rangle$, for some target concept $f \in F$

- instances $\mathbf{x}$ are drawn from distribution D
- teacher provides target values $f(\mathbf{x})$

Learner must output a hypothesis h estimating f

- h is evaluated by its performance on subsequent instances drawn from D

Note: randomly drawn instances, noise-free classifications

Errors

Training error of hypothesis h with respect to target concept f:

- How often $h(\mathbf{x}) \neq f(\mathbf{x})$ over the training instances

True error of hypothesis h with respect to target concept f:

- How often $h(\mathbf{x}) \neq f(\mathbf{x})$ over future, unseen instances (but drawn according to D)

Questions:

- Can we *bound* the true error of a hypothesis given only its training error?
- How many examples are needed for a good approximation?

Approximate Learning

The Need for Approximating the True Error of a Hypothesis

Suppose that we would like to get a hypothesis h with true error 0:

1. **the learner should choose among hypotheses having the training error 0, but since there may be several such candidates, it cannot be sure which one to choose**
2. **as training examples are drawn randomly, there is a non-0 probability that they will mislead the learner**

Consequence: demands on the learner should be weakened

1. **let $error_{\mathcal{D}}(h) < \epsilon$ with ϵ arbitrarily small**
2. **not every sequence of training examples should succeed, but only with probability $1 - \delta$, with δ arbitrarily small**

Approximate Learning

PAC Learnability: Remarks (I)

- If C is PAC-learnable, and each training example is processed in polynomial time,
  then each $c \in C$ can be learned from a polynomial number of training examples.
- Usually, to show that a class C is PAC-learnable, we show that each $c \in C$ can be learned from a polynomial number of examples, and the processing time for each example is polynomially bounded.

What is PAC Learning

Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using the hypothesis space H.

C is PAC-learnable by L using H if

for all $c \in C$, distributions $\mathcal{D}$ over X, ϵ such that $0 < \epsilon < 1/2$, and δ such that $0 < \delta < 1/2$,

the learner L will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \epsilon$,
in time that is polynomial in $1/\epsilon$, $1/\delta$, n and $size(c)$,

where $size(c)$ is the encoding length of c, assuming some representation for C.

PAC Learning

- How does PAC make learning and generalization feasible?

- I don't need to aim for 0 true error with 100% probability.
  It's OK to say true error will be less than 5% with 90% probability
  $\in = 0.05$, $1-\delta = 0.90$ or $\delta = 0.10$

- The learning algorithm is only acceptable if it happens in polynomial time of the parameters

Sample Complexity is important

In practical applications of machine learning, evaluating the sample complexity (i.e. the number of needed training examples) is of greatest interest because in most practical settings limited success is due to limited available training data.

We will present results that relate (for different setups)

- **the size of the instance space** (m)

to

- **the accuracy to which the target concept is approximated** (ϵ)
- **the probability of successfully learning such an hypothesis** $(1-\delta)$
- **the size of the hypothesis space** $(|H|)$

Sample Complexity for Finite Hypothesis Spaces

First, we will present a general bound on the sample complexity for **consistent learners**, i.e. which perfectly fit the training data.

Recall the **version space** notion:

$$VS_{H,D} = \{h \in H | \forall \langle x, c(x) \rangle \in D, h(x) = c(x)\}$$

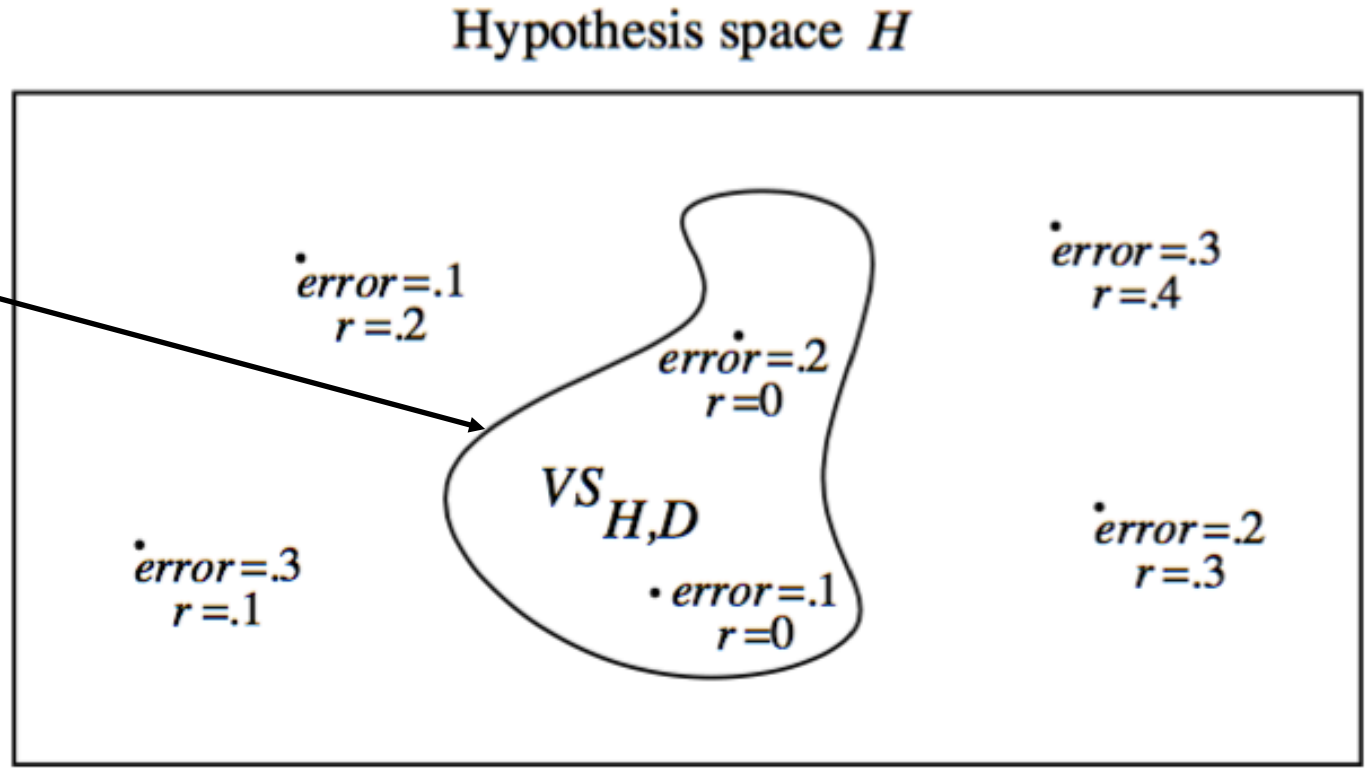
Later, we will consider **agnostic learning**, which accepts the fact that a zero training error hypothesis cannot always be found.

Exhaustion of the Version Space

Definition:

$VS_{H,D}$ is ϵ-exhausted with respect to the target concept c and the training set D if $error_{\mathcal{D}}(h) < \epsilon$, $\forall h \in VS_{H,D}$.

Version space (VS) is the set of hypotheses that get 0 training error. ∈-exhausted means a VS whose true error has an upper bound of ∈.



r = training error, $error$ = true error

Sample Complexity Evaluation

- If Version Space (VS) is ϵ-exhausted, what does it mean?
  - a hypothesis in this VS has training error = 0 **AND**
  - a hypothesis in this VS has true error < ϵ with probability (1- δ)

  for PAC learning.

  => By PAC learning, we want the error to be less that error with a probability of at least (1-δ)

  => I am not insisting that the error be less than ϵ on ALL instances.
- Is it possible that a hypothesis that has true error of more than ϵ be part of the VS?
- Well, for this to happen it should classify ALL the training data correctly.

Finite Hypothesis Set

- Assume H is finite
- Consider $h_1 \in H$ such that $error(h,f) > \varepsilon$. What is the probability that it will correctly classify m training examples?
- If we draw <u>one</u> training example, $(\mathbf{x}_1, y_1)$, what is the probability that h_1 classifies it correctly?

$$P\left[h_1(\mathbf{x}_1) = y_1\right] \leq (1 - \varepsilon)$$

By definition of true error

- What is the probability that h_1 will be right m times?

$$P_D^m\left[h_1(\mathbf{x}_i) = y_i\right] \leq (1 - \varepsilon)^m$$

Finite Hypothesis Set

- Now consider a second hypothesis h_2 that is also ε-bad. What is the probability that either h_1 or h_2 will survive the m training examples?

$$
\begin{aligned}
P_D^m\left[h_1 \vee h_2 \text{ survives}\right] &= P_D^m\left[h_1 \text{ survives}\right] + \\
&\quad P_D^m\left[h_2 \text{ survives}\right] - \\
&\quad P_D^m\left[(h_1 \wedge h_2) \text{ survives}\right] \\
&\leq P_D^m\left[h_1 \text{ survives}\right] + \\
&\quad P_D^m\left[h_2 \text{ survives}\right] \\
&\leq 2(1-\varepsilon)^m
\end{aligned}
$$

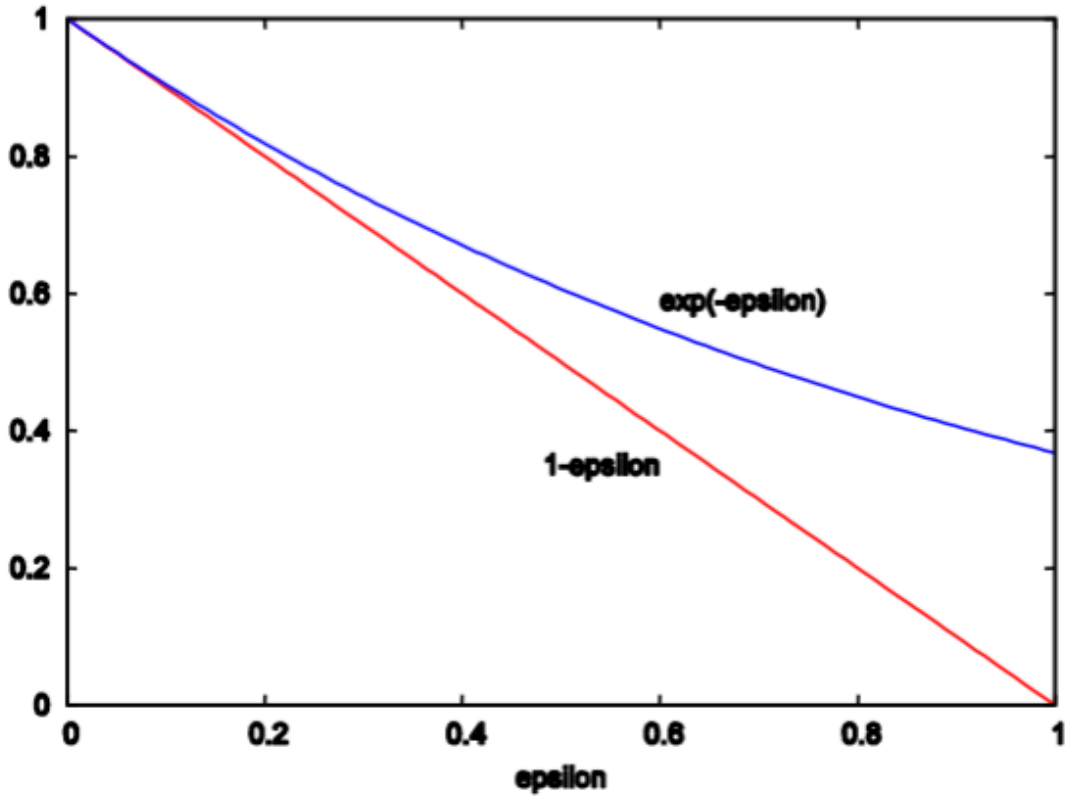
- So if there are k ε-bad hypotheses, the probability that any one of them will survive is $\leq k(1-\varepsilon)^m$
- Since $k < |H|$, this is $\leq |H|(1-\varepsilon)^m$

Finite Hypothesis Set

- Fact: When $0 \leq \varepsilon \leq 1$, $(1 - \varepsilon) \leq e^{-\varepsilon}$

  therefore

  $$|H|(1 - \varepsilon)^m \leq |H| \, e^{-\varepsilon m}$$



Blumer Bound

- Lemma. For a finite hypothesis space H, given a set of m training examples drawn independently according to D, the probability that there exists a hypothesis $h \in H$ with true error greater than ε consistent with the training examples is less than $|H|e^{-\varepsilon m}$.
- We want to ensure that this probability is less than δ.

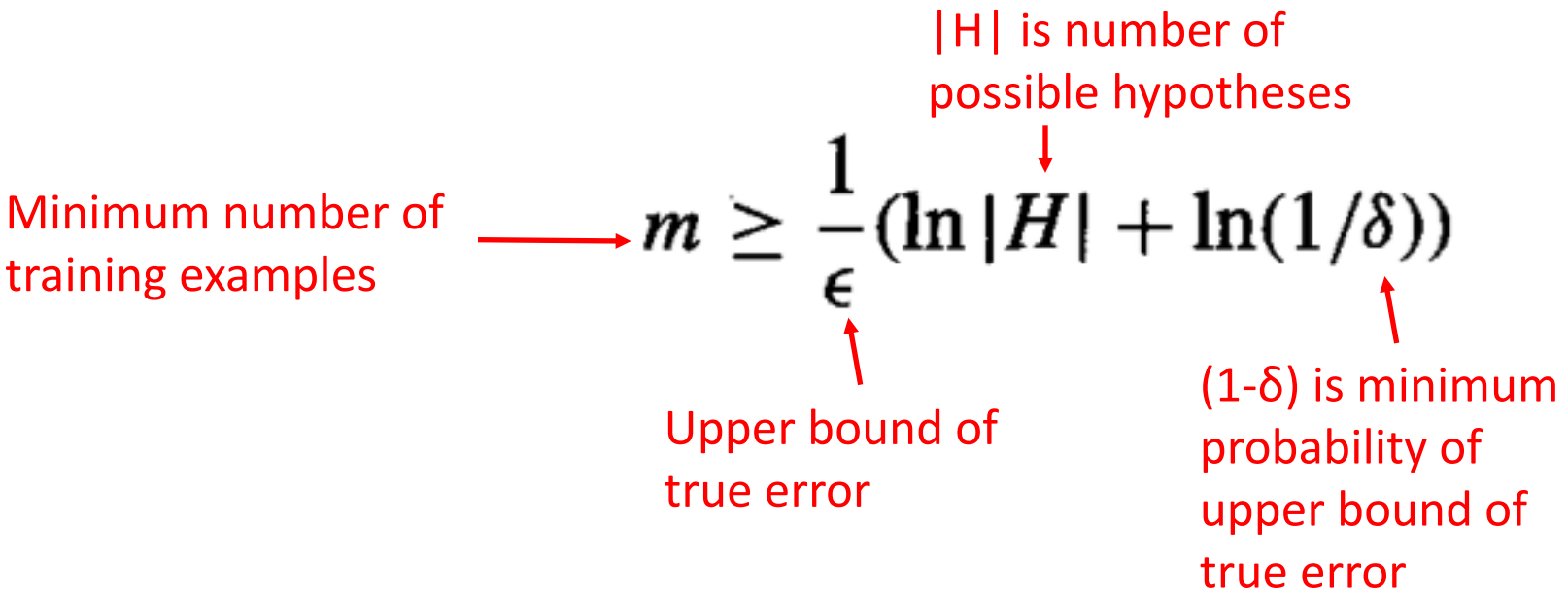
$$|H|e^{-\varepsilon m} \leq \delta$$

- This will be true when

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

Sample Complexity

- Sample complexity: How many training instances should I train my classifier on for PAC learning



- This number ***m*** of training examples is sufficient to assure that any consistent hypothesis will be probably (with probability (1 - δ)) approximately (within error ∈) correct.

Blumer Bound

- Corollary: If $h \in H$ is consistent with all m examples drawn according to D, then the error rate ε on new data points can be estimated as

$$\varepsilon = \frac{1}{m}\left(\ln|H| + \ln\frac{1}{\delta}\right)$$

Evaluating number of hypotheses

- Most frequently, you would like to evaluate m (number of training instances), given parameters like $\in$, and (1-δ). You have to evaluate the number of hypotheses possible for your dataset.
- There are two cases:
  1. Conjunctions learning
     h(x) = x1 ^ x2 ^… ^xn
  or h(x) = x1 ^ ¬x2 ^ … ^ ¬xn

In first case, I only take positive value of some literals and in second case, I consider positive values and negations

The number of hypotheses |H| is simple to evaluate. For first case $|H| = 2^n$ and second case $|H| = 3^n$. Note that we can learn positive instances i.e. h(x) = 1 or negative instances i.e. h(x) = 0.

See more details at: https://www.cs.utexas.edu/~mooney/cs391L/slides/colt.pdf

Evaluating number of hypotheses

- 2. Learning a complex Boolean function, such as a decision tree (DT) or DNF.
  DT: h(x) = (1st branch of tree) $\vee$ (2nd branch of tree) $\vee$ (nth branch of tree)

  Note: now you have to take permutations of all branches i.e. you can label first branch as 0 and second as 1 and so on
  so $|H| = 2^{2^n}$.

  Thought questions:
  1. Suppose you grow the tree fully to a depth of k, how many hypotheses are possible.
  2. Suppose you have attributes that can take k values each, how many hypotheses are possible

See more details at: https://www.cs.utexas.edu/~mooney/cs391L/slides/colt.pdf

Evaluating number of hypotheses

- 2. Learning a complex Boolean function, such as a decision tree (DT) or DNF.
  k-DNF: $h(x) = (T_1) \vee (T_2) \vee (T_n)$
  where each term T can have up to k conjunctions of literals and their negations i.e. x1 or ¬x1.

  So, you can have up to $(2n)^k$ terms. Since it is a complex function, you can label each term as 0 or 1 and make a grand hypothesis.
  So, $|H| = 2^{2n^k}$

See more details at: https://www.cs.utexas.edu/~mooney/cs391L/slides/colt.pdf

Examples

- Boolean conjunctions over n features.

  $|H| = 3^n$, since each feature can appear as x_j, $\neg x_j$, or be missing.

$$\varepsilon = \frac{1}{m}\left(\ln 3^n + \ln\frac{1}{\delta}\right) = \frac{1}{m}\left(n\ln 3 + \ln\frac{1}{\delta}\right)$$

Suppose hypotheses are in the form:
h(x) = 1
where h(x) is a Boolean conjunction

- k-DNF formulas:

www.math.tau.ac.il/~mansour/ml-course-02/ml5.ppt
https://en.wikipedia.org/wiki/Disjunctive_normal_form

$$(x_1 \wedge x_3) \vee (x_2 \wedge \neg x_4) \vee (x_1 \wedge x_4)$$

There are at most $(2n)^k$ disjunctions, so $|H| \leq 2^{(2n)^k}$

This is assuming that for each k-DNF, you have 2 choices – label it 0 or 1.

- for <u>fixed</u> k, this gives

$$\log_2 |H| = (2n)^k$$

- which is polynomial in n: $\varepsilon = \frac{1}{m} O\left(n^k + \ln\frac{1}{\delta}\right)$

This is different from the k-term DNF that the textbook talks about in section 7.3.3.2

Examples

If you are given a fixed number of hypotheses, what's the value of m

$$\varepsilon = \frac{1}{m}\left(\ln|H| + \ln\frac{1}{\delta}\right)$$

If H is as given in *EnjoySport* and $|H| = 729$, and

$$m \geq \frac{1}{\varepsilon}\left(\ln 729 + \ln\frac{1}{\delta}\right)$$

... if want to assure that with probability 95%, VS contains only hypotheses with $error_D(h) \leq .1$, then it is sufficient to have m examples, where

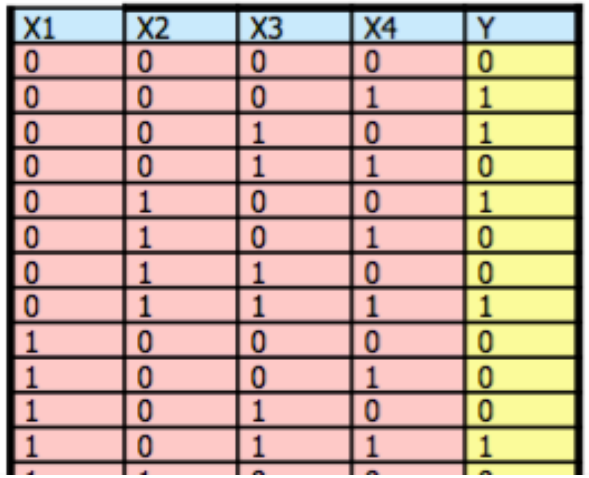
$$m \geq \frac{1}{.1}\left(\ln 729 + \ln\frac{1}{0.05}\right)$$

$$= 10 \cdot (\ln 729 + \ln 20) = 10 \cdot (6.59 + 3.00)$$

$$= 95.9$$

PAC in action

For first 2 cases, hypotheses are simple and are of the form h(x) = 1

Machine	Example Hypothesis	|H| (**n** features)	**m** required to PAC-learn
And-positive-literals	X3 ^ X7 ^ X8	2^n	$\frac{1}{\varepsilon}\left(n(\ln 2)+\ln\frac{1}{\delta}\right)$
And-literals	X3 ^ ¬X7	3^n	$\frac{1}{\varepsilon}\left(n(\ln 3)+\ln\frac{1}{\delta}\right)$
Lookup Table		2^{2^n}	$\frac{1}{\varepsilon}\left(2^n(\ln 2)+\ln\frac{1}{\delta}\right)$

X1	X2	X3	X4	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1

In this case, you find all possible permutations of attributes and then label them as 0 or 1.

Example

- Your company decides to take on a project where there are 10 Boolean attributes and a Boolean output. The client wants you to use 100 training instances to generate a hypothesis in the form of AND-POSITIVE literals. The requirement is this: your hypothesis should have an accuracy of at least 95% on more than 99% of ALL the instances in the universe. Would you take on this project.

Example

- Required values $\in$ = 0.05, 1-δ = 0.99
  $|H| = 2^{10} = 1024$
  Using equation:

  With 100 training data, how much error can I expect:

  $$\epsilon = \frac{1}{m}\left(\ln(|H|) + \ln\left(\frac{1}{\delta}\right)\right)$$

  $$\epsilon = \frac{1}{100}\left(\ln(1024) + \ln\left(\frac{1}{0.01}\right)\right)$$

  = 0.115

I can only guarantee max error of 11.5% on 99% of the data. So, this project is not feasible.

Something to think about

- Story so far…

You have a machine that generates the hypotheses H.
You match them against training data, eliminate most of H.
At the end you output the one with minimum (hopefully 0) training error.

We make a big assumption: The concept is in the hypotheses set generated by the machines.

What if it is not??

Sample Complexity for Agnostic Learning

Agnostic learning doesn't assume $c \in H$, therefore c may or may not be perfectly learned in H. In this more general setting, a hypothesis which has a zero training error cannot always be found.

- **What can we search for?**
  A hypothesis h that makes the fewest errors on training data.
- **What is the sample complexity in this case?**

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

Proof idea: use Hoeffding-Chernoff bounds

$$Pr[error_{\mathcal{D}}(h) > error_D(h) + \epsilon] \leq e^{-2m\epsilon^2}$$

Something to think about

- We started off by considering discrete attributes.
- For discrete attributes, H is finite.
- What if the attributes are continuous?
- You would get infinite number of attributes.
- The process of elimination by using training example would not work.
- Instead, we consider the expressive power of a hypothesis.
  We take n points and find all the ways in which they can be labeled and check whether the hypothesis can "shatter" it (separate the data correctly).

Shattering of a set

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.

In other words: The instances can be classified in every possible way.

Dichotomy

- Dichotomy refers to the number of ways of labeling N points as either + or -

- Given N points, they can be labeled in 2^N ways as positive or negative.

Dichotomy

- Dichotomy means finding all possible class distributions. For a sample of 3 instances and 2 classes:

x1	x2	x3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

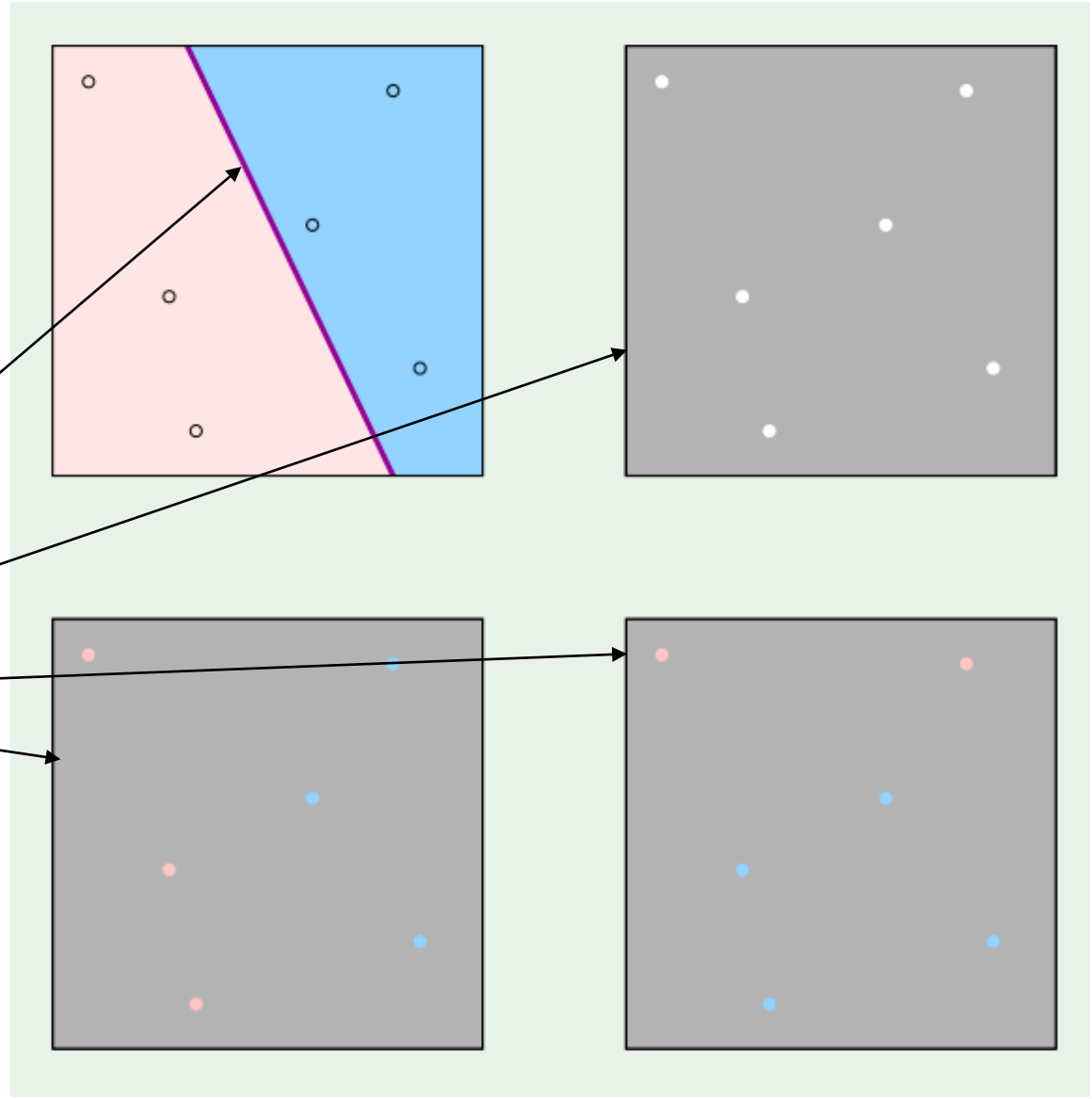
All 3 instance belong to class 0

All 3 instance belong to class 1

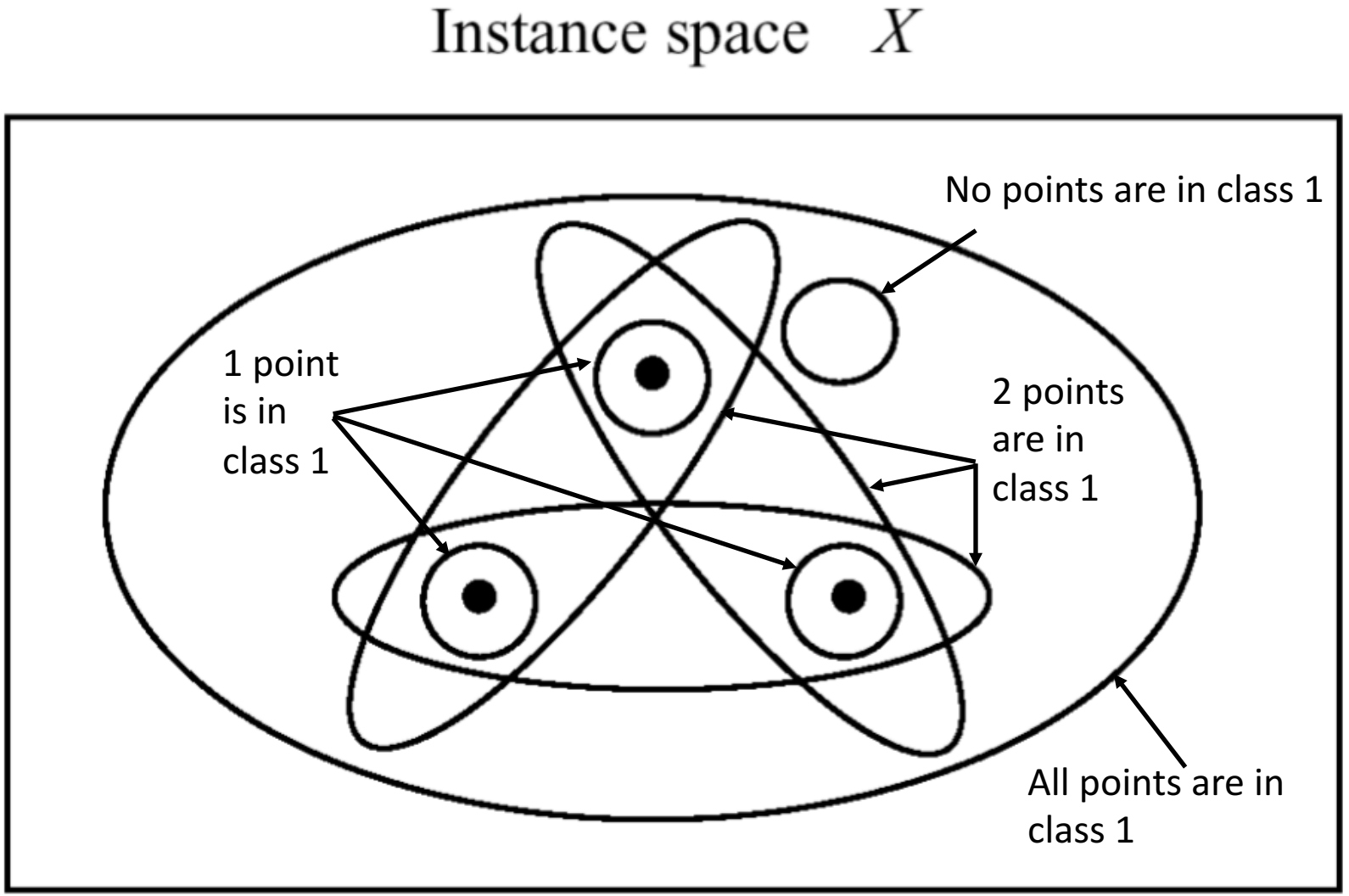
The dataset is said be to shattered by a hypothesis if it can represent all the cases

Why do we study dichotomy

- measure of the representational power of the hypothesis.
- If I label a set of points in a given way, can the hypothesis work on them?
- Can the hypothesis also work with this data or a different dichotomy?



Shattering of 3 points



3 points have 8 possible dichotomies.
We need to look for a hypothesis set H that can shatter all of these cases.

We label everything within the shape to be class 1 and outside the shape to be class 0

How many of these labelings can a particular hypothesis correctly represent ("shatter")?

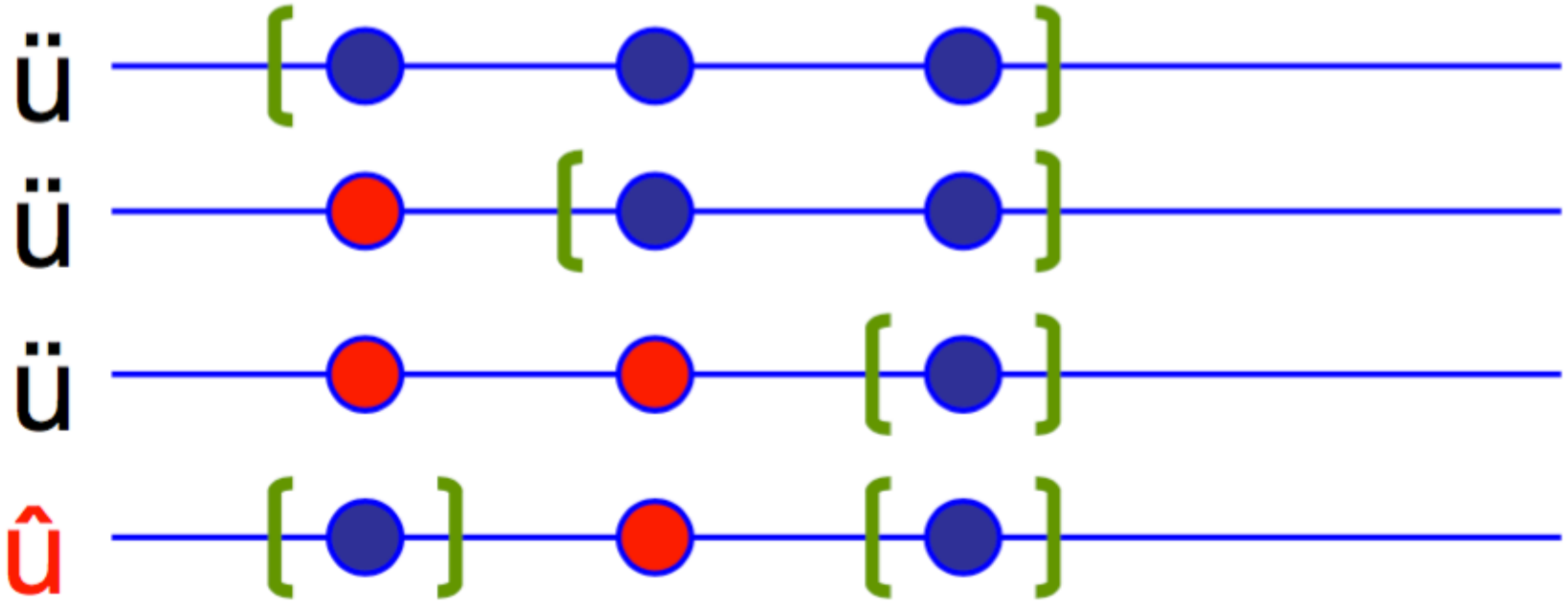
VC Dimension

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 (but finite) sets of X can be shattered by H, then VC(H) $\equiv \infty$.

- For finite H, VC(H) $\leq \log_2 |H|$

VC Dimension Example

- Let H be the set of intervals on the real line such that $h(\mathbf{x})=1$ iff $\mathbf{x}$ is in the interval.
- How many points can be shattered by H?



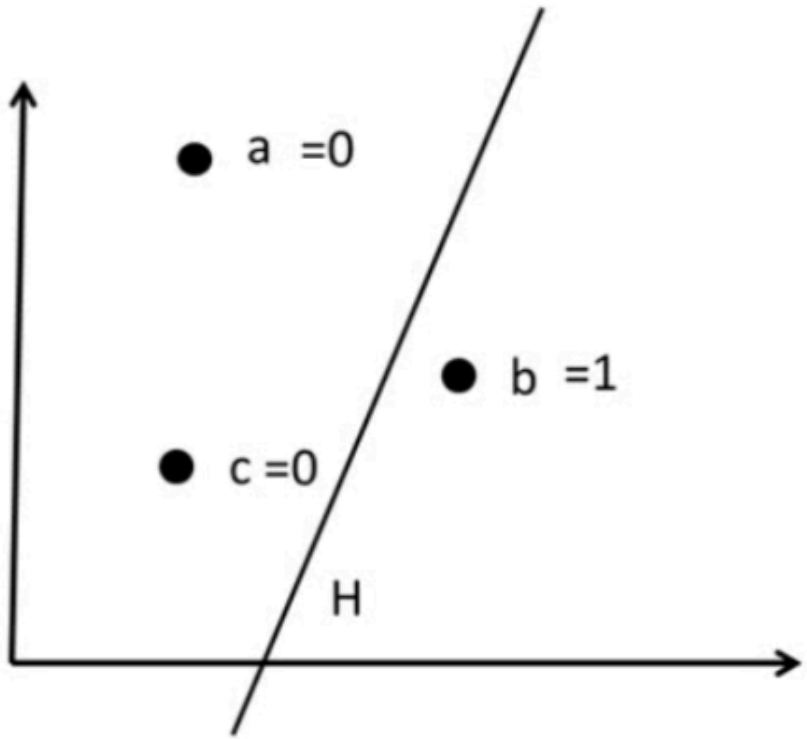
- 2 points. It cannot shatter 3. VC(H)=2

Shattering

Statement: A Hypothesis (H) shatters m points in n-dimensional space if all possible combinations of m points in n-dimensional space are correctly classified by H.

2 dimensions, 3 points - > Can you shatter them using linear classifier

Explanation: H



n=2

m=3

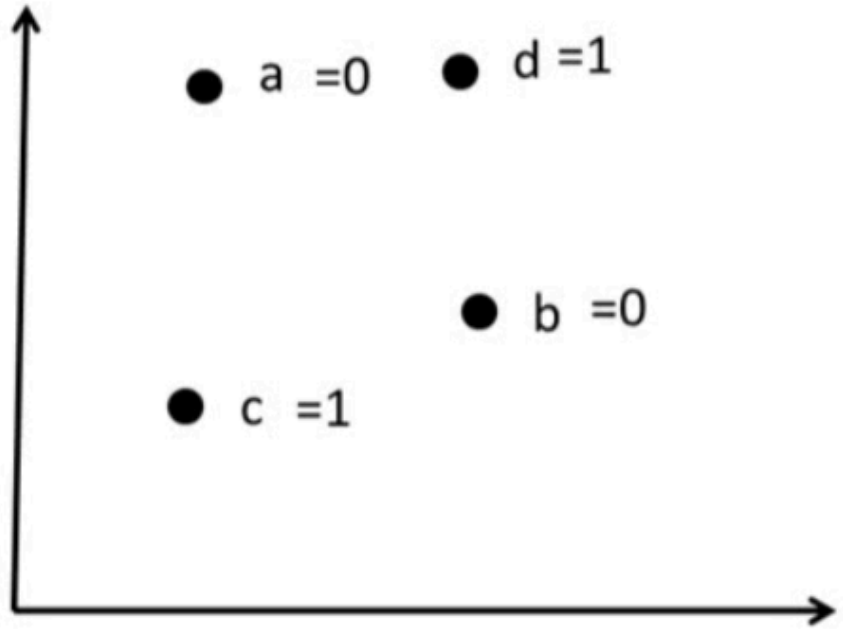
2^m possible arrangements

a	b	c
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Statement: A Hypothesis (H) shatters m points in n-dimensional space if all possible combinations of m points in n-dimensional space are correctly classified by H.

2 dimensions, 4 points - > Can you shatter them using linear classifier

Explanation: H



n=2

m=4

2^m possible arrangements

Notice that I have labeled the diagonally opposite points to be of the same class.

You can vary the straight line as much as you want, but you won't be able to shatter this combination

So, VC dimension of a straight line in 2-D = 3

VC Dimension Linear Separator

- We cannot separate any set of 4 points (XOR).
- In general, the VC(LTU) in n-dimensional space is n+1.
- A good heuristic is that the VC-dimension is equal to the number of tunable parameters in the model (unless the parameters are redundant)

Growth Function

- The growth function counts the maximum number of dichotomies on any N points that a hypothesis can shatter.

$$m_{\mathcal{H}}(N) = \max_{\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathcal{X}} |\mathcal{H}(\mathbf{x}_1, \cdots, \mathbf{x}_N)|$$

- In other words, how many dichotomies can your hypothesis represent (or shatter).
- m will satisfy:

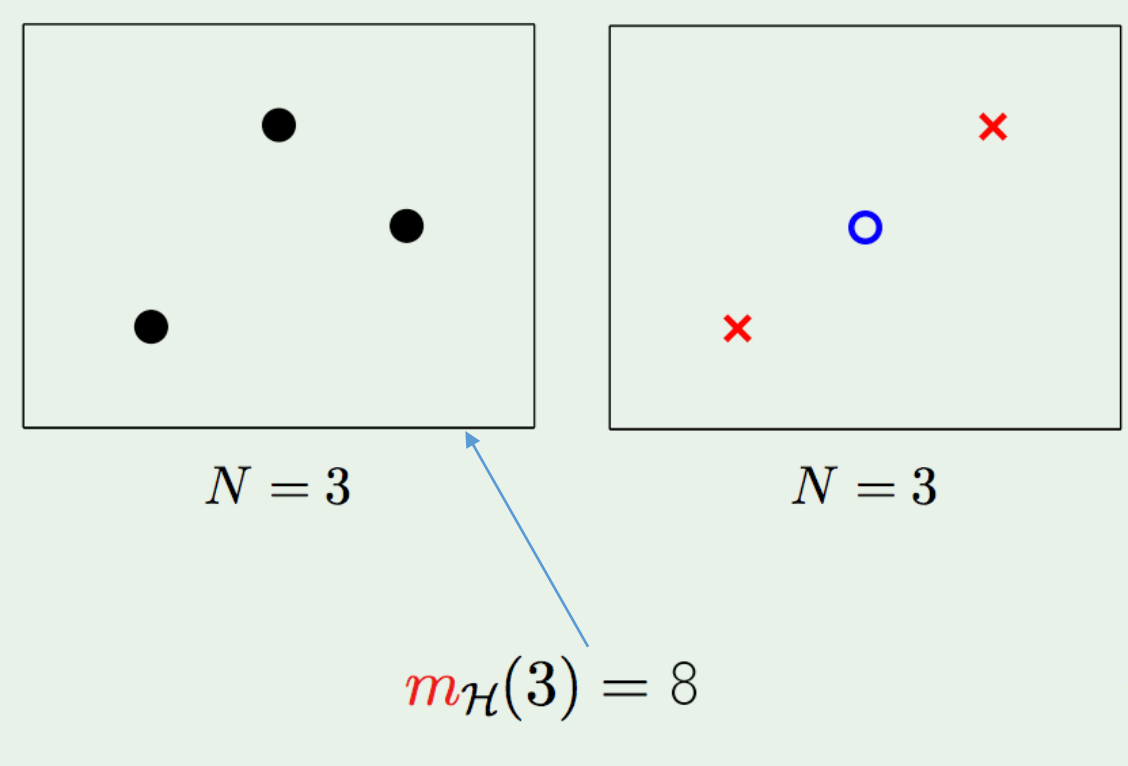
$$m_{\mathcal{H}}(N) \leq 2^N$$

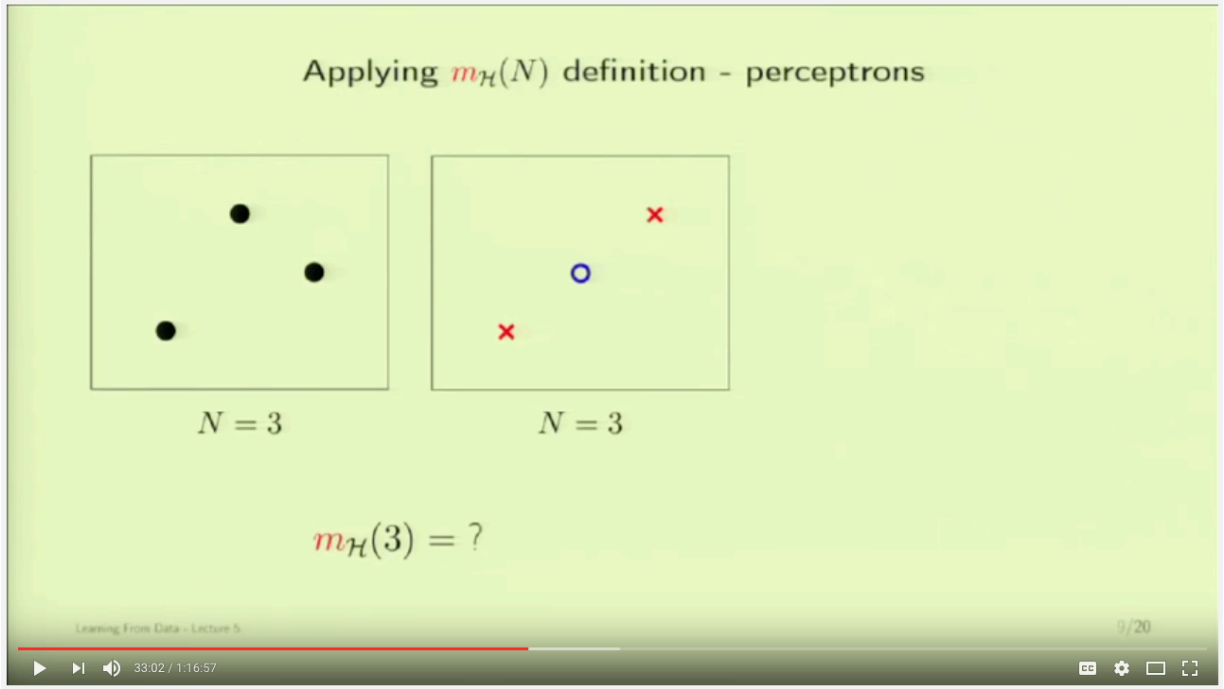
How does it work

- I give you N and you choose where to place the points.
- Choose the placement to the advantage of your hypothesis.
- Watch the video below.

How does it work

- I give you N = 3 and hypothesis is a perceptron in 2-D.
- You will place the points to your advantage

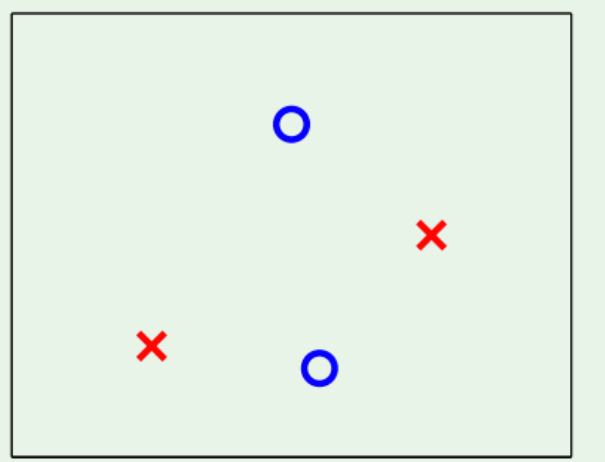




Watch the video above

What about 4 points

- If you have 4 points, a perceptron in 2-D can only shatter 14 out of 16 (2^4) dichotomies at best.
- Try this yourself.



$$N = 4$$

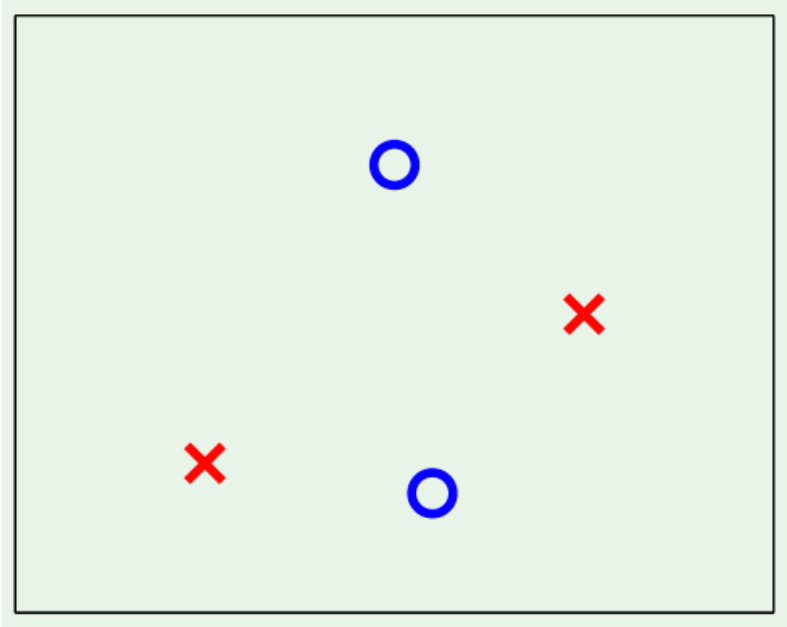
$$m_{\mathcal{H}}(4) = 14$$

Break Point of a Hypothesis

- If no data set of size k can be shattered (completely represented) by H, then k is the ***break point*** for H.

  $m_H(k) < 2^k$

- For 2-D perceptron, k = 4.



Relation to VC dimension

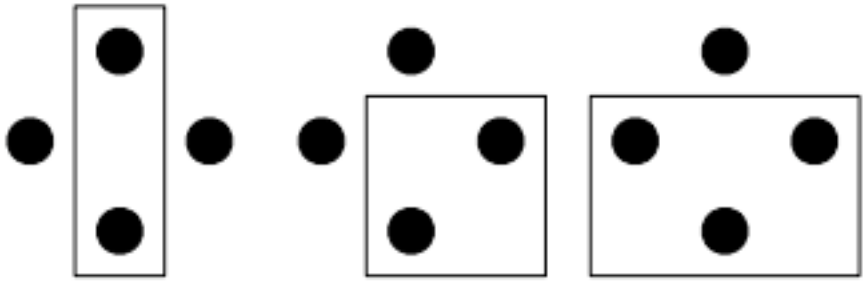
- The VC dimension of a hypothesis set H is the largest value of N for which:
  $m_H(N) = 2^N$
  i.e. H can shatter any dichotomy of the N points.

VC Dimension – Examples

- In order to show that the *VCdim* of a class is *d* we have to show:
  - $VC\dim \geq d$: find **some** shattered set of size d.
  - $VC\dim < d+1$: show that no set of size d+1 is shattered

VC dimension of a rectangle.

- Adversary tells you to shatter 4 points. You choose the arrangement.

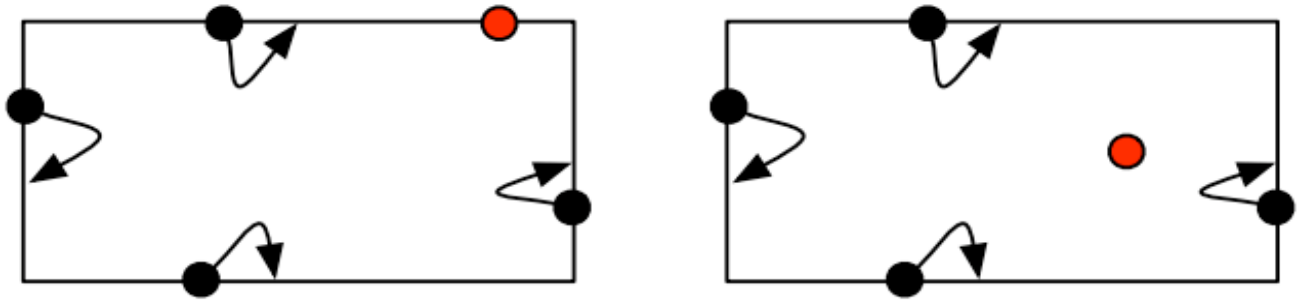


- Trivial to capture just 1 or all 4. Above shows way to capture 2 and 3.
- So 4 can be done. Just don't insist that all be on the same straight line or arranged like a rectangle. ☺

VC dimension of a rectangle.

- Adversary tells you to shatter 5 points now. You choose the arrangement.

Suppose we have 5 points. A shattering must allow us to select all 5 points and allow us to select 4 points without the 5th.



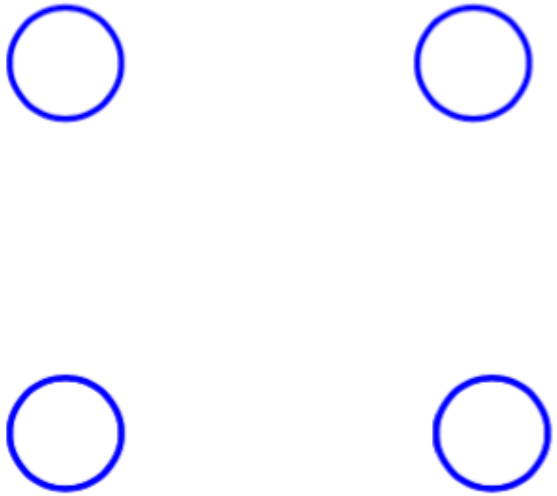
Our minimum enclosing rectangle that allows us to select all five points is defined by only four points – one for each edge. So, it is clear that the fifth point must lie either on an edge or on the inside of the rectangle. This prevents us from selecting four points without the fifth.

VC Dimension for Circle

- Let H be the set of circles in 2-D such that $h(\mathbf{x})=1$ iff $\mathbf{x}$ is inside the circle.
- How many points can be shattered by H?

VC Dimension for Circle

- Let H be the set of circles in 2-D such that $h(\mathbf{x})=1$ iff $\mathbf{x}$ is inside the circle.
- How many points can be shattered by H?



- VC(H)=3

VC Dimension of a triangle

- What would be the VC dimension of a triangle in x-y plane (2-D)? You can assume that the hypothesis would be that points in the triangle are of class +1 and outside are class -1.
  Please do some research on the above question.

Resources for learning VC dimensions

- http://www.cs.nyu.edu/~mohri/mls/lecture_3.pdf
- http://www.ccs.neu.edu/home/rjw/csg220/lectures/VC-dimension.pdf
- http://work.caltech.edu/slides/slides07.pdf

Relationship between Sample Complexity and VC Dimension

- We can use VC dimension of a hypothesis to find sample complexity as follows:

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