CS 4375 – Introduction to Machine Learning

Computational Learning Theory

Erick Parolin

THE UNIVERSITY OF TEXAS AT DALLAS

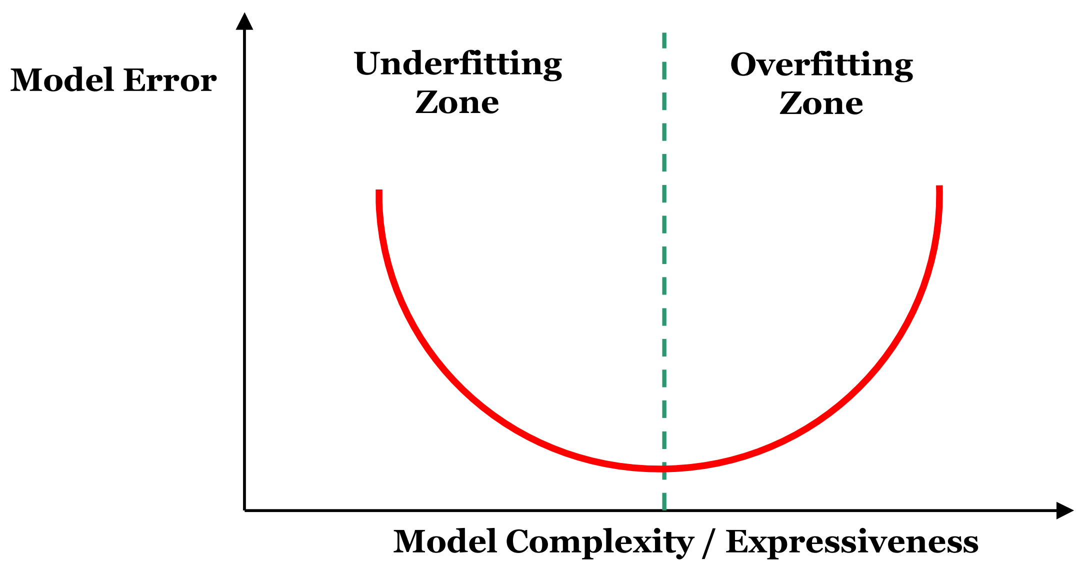
[Based on the slides from Dr. Vibhav Gogate, Dr. Nicholas Ruozzi, and Dr. Raymond J. Mooney]

Computational Learning Theory

- We have been focusing on learning algorithms, each of which explore different hypothesis space to find the best hypothesis.
- **Hypothesis space refers** to the set of all possible models or functions that a learning algorithm can choose from to explain or predict the data.
  - Formally, if X is the input space and Y is the output space, the hypothesis space H consists of all functions $h: X \rightarrow Y$ that the learning algorithm can potentially select as the final model based on the training data.
- **Expressiveness of a hypothesis space** refers to how well the set of hypotheses (models or functions) in the space can capture or represent the underlying patterns in the data. It describes the **range and complexity** of the functions that the hypothesis space can potentially model.
  - A linear classifier (e.g., a perceptron or logistic regression) has limited expressiveness because it can only separate data using straight lines (hyperplanes). It cannot model complex decision boundaries.
  - A deep neural network has much higher expressiveness because it can model more complex, non-linear decision boundaries and relationships between inputs and outputs.

Computational Learning Theory

Spoiler: The expressiveness of the hypothesis space must be carefully balanced to avoid both underfitting and overfitting. Ideally, you want a hypothesis space that is expressive enough to model the true underlying patterns but not so expressive that it captures noise or irrelevant details in the data.



Computational Learning Theory

But…

- How do we know that the learned hypothesis will perform well on the test set (or unseen data)?
- How many samples do we need to make sure that we learn a good hypothesis?
- In what situations is learning possible?

Computational Learning Theory

Complexity of a Learning Problem

- Complexity of a learning problem depends on
  - Size/expressiveness of the hypothesis space
  - Accuracy to which the true function must be approximated
  - Probability with which the learner must produce a successful hypothesis

- Measures of complexity:
  - **Sample complexity:** how much data you need in order to (with high probability) learn a good hypothesis
  - **Computational complexity:** Amount of time and space required to accurately solve (with high probability) the learning problem
  - Higher sample complexity means higher computational complexity

Is Perfect Learning Possible?

- **What is the number of training examples needed to learn a hypothesis *h* for which *error(h)=0* ?**
  - There may be multiple hypotheses that are consistent with the training data and the learner cannot be certain to pick the one that equals the target concept.
  - Since training data is drawn randomly, there is always a chance that the training examples are misleading!

PAC Learning

Probably Approximately Correct (PAC)

- The only reasonable expectation of a learner is that with high probability it learns a close approximation to the target concept.

- In the PAC model, we specify two small parameters, ε and δ, and require that with **probability at least $(1 - \delta)$** a system learns a concept with **error at most ε**.

PAC Learning

Problem Setting

- **X** is the set of **all possible instances**
- **C** is the **set of target concepts** that our learner might be called upon to learn
- Each **target concept** $\boldsymbol{c} \in C$ corresponds to some subset of X, or equivalently to some Boolean-valued function $c\colon X \rightarrow \{0, 1\}$.
- Instances are generated at random from X according to some probability distribution $\boldsymbol{\mathcal{D}}$
- **Training examples** are generated by drawing an instance x at random according to $\mathcal{D}$, then presenting x along with its target value, c(x), to the learner.

PAC Learning

Problem Setting

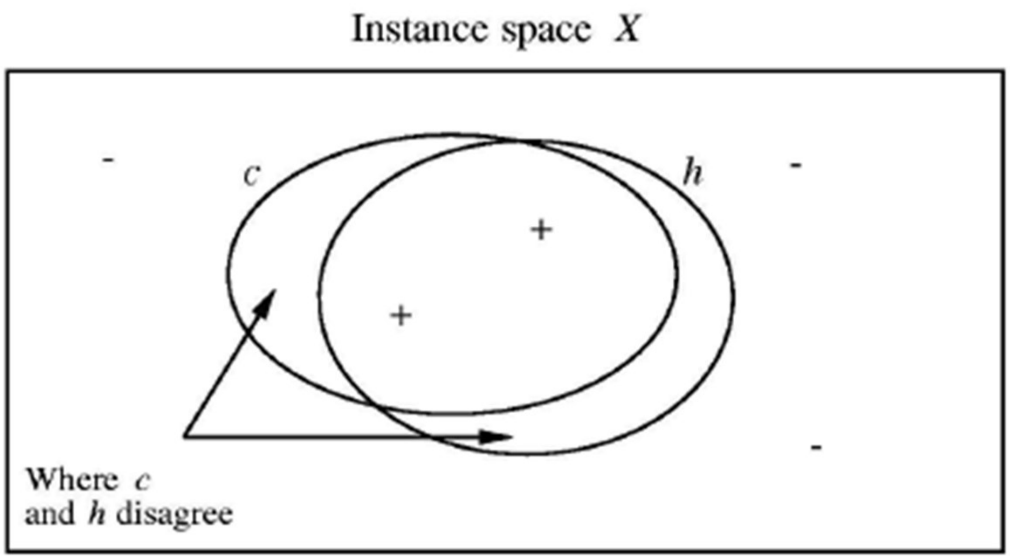
- **The learner L** considers some set H of possible hypotheses when attempting to learn the target concept.
  - **Example:** H might be the set of all hypotheses describable by conjunctions of the attributes age and height
- After observing a sequence of training examples of the target concept c, L must output some hypothesis h from H, which is its estimate of c.
- To be fair, we evaluate the success of L by the performance of h over new instances drawn randomly from X according to $\mathcal{D}$, the same probability distribution used to generate the training data.

We are interested in characterizing the performance of various learners L using various hypothesis spaces H, when learning individual target concepts drawn from various classes C.

Note: Because we demand that L be general enough to learn any target concept from C regardless of the distribution of training examples, we will often be interested in worst-case analyses over all possible target concepts from C and all possible instance distributions $\mathcal{D}$.

PAC Learning

True Error of a Hypothesis



Definition: The **true error** (denoted ***error***$_{\mathcal{D}}$**(*h*)**) of hypothesis ***h*** with respect to target concept c and distribution $\mathcal{D}$ is the probability that ***h*** will misclassify an instance drawn at random according to $\mathcal{D}$**.**

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

PAC Learning

Training vs True 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 c

- How often $h(x) \neq c(x)$ over future random instances

Our concern:

- Can we bound the true error of h given the training error of h?

PAC Learning

PAC Learnability – Definition

Given:

- A concept class C over an instance space X containing instances of **length n.**
- A learner **L**, using a hypothesis space H.
- Two constants: $0 < \varepsilon < 0.5$ and $0 < \delta < 0.5$.

Definition: C is said to be **<u>PAC-learnable</u>** by L using H iff for all $c \in C$, distributions $\mathcal{D}$ over X, $0 < \varepsilon < 0.5$, $0 < \delta < 0.5$; learner L by sampling random examples from distribution $\mathcal{D}$, will with probability at least $1 - \delta$ output a hypothesis $h \in H$ such that **$\text{error}_{\mathcal{D}}(\mathbf{h}) \leq \boldsymbol{\varepsilon}$, in time polynomial in $\mathbf{1/\boldsymbol{\varepsilon}}$, $\mathbf{1/\boldsymbol{\delta}}$, $\boldsymbol{n}$ and $\boldsymbol{size}(\boldsymbol{c})$.**

PAC Learning

PAC Learnability – Definition

- PAC-learnability seems to be concerned about *computational resources* required for learning
  - In practice, **we are only concerned about the number of training examples required**
- Parameters $1/\varepsilon$ and $1/\delta$ directly control how accurate and reliable the learned hypothesis needs to be, which in turn controls the number of training examples.
  - **More examples are needed as ε decreases:** the finer the distinction you want to make (smaller error), the more data you need to distinguish between hypotheses in the hypothesis space.
  - **More examples are needed as δ decreases:** reducing the probability of failure (smaller δ) requires the algorithm to verify that the hypothesis generalizes well, which requires more data.
- The growth in the number of examples with respect to $\boldsymbol{\varepsilon}$ and δ must be polynomial
  - If L requires some minimum processing time per training example, then for C to be PAC-learnable by L, L must learn from a polynomial number of training examples.

Sample Complexity

PAC Learnability

- PAC-learnability is largely determined by the number of training examples required by the learner.

- Proving PAC learnability:
  (1) Prove sample complexity of learning C using H is polynomial.
  (2) Prove that the learner can train on a polynomial-sized data set in polynomial time.

- To be PAC-learnable, there must be a hypothesis in H with arbitrarily small error for every concept in C, generally $C \in H$.

Sample Complexity

Sample Complexity for Consistent Learners

Definition: A learner L using a hypothesis space H and training data **D** is said to be a ***consistent learner*** if it always outputs a hypothesis with zero error on D whenever H contains such a hypothesis.

Definition: The subset of all hypotheses $h \in H$ that correctly classify the training examples **D** is called the **version space** with respect to the hypothesis space H and the training examples D.

- By definition, a **consistent learner** must produce a hypothesis in the **version space** for H given D (let it be $\boldsymbol{VS_{H,D}}$)

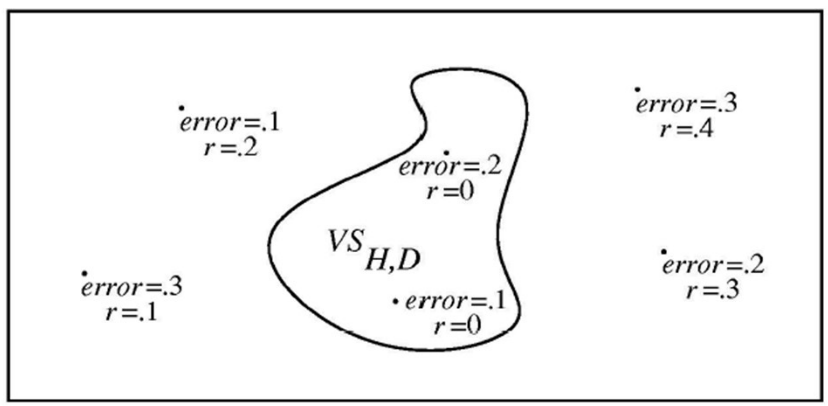
Sample Complexity

Sample Complexity for Consistent Learners

Version Space $VS_{H,D}$: subset of hypothesis in H consistent with training data D.

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

Hypothesis space H



(r = training error, $error$ = true error)

Sample Complexity

Sample Complexity for Consistent Learners

Version Space $VS_{H,D}$: subset of hypothesis in H consistent with training data D.

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

✓ **To bound the number of examples needed by a consistent learner, we just need to bound the number of examples needed to ensure that the version-space contains no hypotheses with unacceptably high error.**

Sample Complexity

ε-Exhausted Version Space

Definition: The version space, $\boldsymbol{VS_{H,D}}$ is said to be **ε-exhausted** iff every hypothesis in it has ***true error*** **less than or equal to ε**.

- In other words, there are enough training examples to guarantee that any consistent hypothesis has true error at most ε.
- One can never be sure that the version-space is ε-exhausted, but one can bound the **probability** that it is not.

Theorem (Haussler, 1988): If the hypothesis space H is finite, and D is a sequence of $m \geq 1$ independent random examples for some target concept c, then for any $0 \leq \varepsilon \leq 1$, the probability that the version space $\mathrm{VS}_{\mathrm{H,D}}$ is ***not*** ε-exhausted is less than or equal to $|\boldsymbol{H}|\boldsymbol{e}^{-\boldsymbol{\varepsilon m}}$

Sample Complexity

Proof:

Let $\boldsymbol{H_{bad}} = \{\boldsymbol{h_1}, \dots, \boldsymbol{h_k}\}$ be the subset of H with true error > ε. The VS is **not** ε-exhausted if any of these are consistent with all **m** examples.

A single $\boldsymbol{h_i} \in \boldsymbol{H_{bad}}$ is consistent with ***one*** example $\boldsymbol{e_j}$ with probability:

$$P(consist(\mathrm{h_i}, e_j)) \leq (1 - \varepsilon)$$

A single $\boldsymbol{h_i} \in \boldsymbol{H_{bad}}$ is consistent with ***all*** m independent random examples with probability:

$$P(consist(\mathrm{h_i}, D)) \leq (1 - \varepsilon)^{\,m}$$

The probability that **any** $\boldsymbol{h_i} \in \boldsymbol{H_{bad}}$ is consistent with ***all*** m examples is:

$$P\big(consist(\mathrm{h}_{bad}, D)\big) = P\big(consist(\mathrm{h}_1, D)\big) \vee \cdots \vee P(consist(\mathrm{h}_k, D))$$

Sample Complexity

Proof (cont.):

Since the probability of a disjunction of events is at most the sum of the probabilities of the individual events:

$$P\big(consist(\mathrm{h}_{bad}, D)\big) \leq |H_{bad}|(1-\varepsilon)^{\,m}$$

Since $|\boldsymbol{H_{bad}}| \leq |\boldsymbol{H}|$ and $(\mathbf{1}-\boldsymbol{\varepsilon})^{\mathbf{m}} \leq \boldsymbol{e^{-\varepsilon m}}$ for $0 \leq \varepsilon \leq 1$ and $m \geq 1$, then we have:

$$P\big(consist(h_{bad}, D)\big) \leq |H|e^{-\varepsilon m}$$

Sample Complexity

Sample Complexity Analysis:

Let δ be an upper bound on the probability of not exhausting the version space. So:

$$P\big(consist(h_{bad}, D)\big) \leq |H|e^{-\varepsilon} \quad \leq \delta$$

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

$$-\varepsilon m \leq ln\left(\frac{\delta}{|H|}\right)$$

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

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

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

Sample Complexity

Sample Complexity Analysis:

Therefore, any consistent learner, given at least

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

This tells us how many training examples suffice to ensure (with probability (1 - δ)) that every hypothesis in H having zero training error will have a true error of at most ε.

examples will produce a result that is PAC.

- Just need to determine the size of a hypothesis space to instantiate this result for learning specific classes of concepts.

- This gives a **sufficient** number of examples for PAC learning, but not a necessary number. Several approximations like that used to bound the probability of a disjunction make this a gross over-estimate in practice.

Sample Complexity – Examples

Examples

Consider conjunctions over b Boolean features. There are 3^b of these since each feature can appear positively, appear negatively, or not appear in a given conjunction. Therefore $|H| = 3^b$, so a sufficient number of examples to learn a PAC concept is:

$$\frac{\left(ln\frac{1}{\delta}+ln\ 3^b\right)}{\varepsilon} = \frac{\left(ln\frac{1}{\delta}+b\ ln3\right)}{\varepsilon}$$

In practice, we would have:

- $\delta = \varepsilon = 0.05, \;\; b = 10 \; gives \; 280 \; examples$
- $\delta = 0.01, \;\; \varepsilon = 0.05, \;\; b = 10 \; gives \; 312 \; examples$
- $\delta = \varepsilon = 0.01, \;\; b = 10 \; gives \; 1{,}560 \; examples$
- $\delta = \varepsilon = 0.01, \;\; b = 50 \; gives \; 5{,}954 \; examples$

Sample Complexity – Examples

Examples

Consider any Boolean function over *b* Boolean features such as the hypothesis space of DNF or decision trees. There are 2^{2^b} of these, so a sufficient number of examples to learn a PAC concept is:

$$\frac{\left(ln\frac{1}{\delta}+ln\ 2^{2^b}\right)}{\varepsilon} = \frac{\left(ln\frac{1}{\delta}+2^b ln\ \right)}{\varepsilon}$$

In practice, we would have:

- $\delta = \varepsilon = 0.05,\quad b = 10\ gives\ 14{,}256\ examples$
- $\delta = \varepsilon = 0.05,\quad b = 20\ gives\ 14{,}536{,}410\ examples$
- $\delta = \varepsilon = 0.05,\quad b = 50\ gives\ 1.561 \times 10^{16}\ examples$

Agnostic Learning

What if the $c \notin H$?

So far, we assumed that $\boldsymbol{c} \in \boldsymbol{H}$

- Haussler Theorem assumes we have a Version Space $\boldsymbol{VS_{H,D}}$.

In other words, we assume that we have a consistent learner L, or a learner algorithm that generates a hypothesis $\boldsymbol{h} \in \boldsymbol{H}$ that correctly classify the training examples **D**.

This is the same as saying that the concept c is in the hypotheses set generated by the learner L.

But what if this assumption does not hold?

Agnostic Learning

What if the $c \notin H$?

In this case, the most we might ask of our learner is to output the hypothesis from H that has the **minimum error** over the training examples.

Agnostic Learner: makes no prior commitment about whether or not $c \in H$

Let h_{best} denote the hypothesis from H having lowest training error over the training examples.

How many training examples suffice to ensure (with high probability) that its true error $error_{\mathcal{D}}(h_{best})$ will be no more than $\varepsilon + error_{D}(h_{best})$?

Note that $error_{\mathcal{D}}(h_{best})$ corresponds to the true error over the over the entire probability distribution $\mathcal{D}$ while $error_{D}(h_{best})$ is the error over the particular sample of training data D.

Agnostic Learning

Agnostic Learning doesn't assume $c \in H$

- **Hoeffding bounds:** if the training error $error_D(h)$ is measured over the set D containing m randomly drawn examples, then

$$P(error_{\mathcal{D}}(h) > \varepsilon + error_D(h)) \leq e^{-2m\varepsilon^2}$$

- Hoeffding's gives us a bound on the probability that an arbitrarily chosen single hypothesis has a very misleading training error.

- To assure that the *best* hypothesis found by L has an error bounded in this way, we must consider the probability that **<u>any</u>** one of the **|H|** hypotheses could have a large error:

$$\boldsymbol{P((\exists\, \mathbf{h} \in H)\ \ (error_{\mathcal{D}}(h) > \varepsilon + error_D(h))) \leq |H| e^{-2m\varepsilon^2}}$$

Agnostic Learning

Agnostic Learning doesn't assume $c \in H$

- **Theorem:** For a finite hypothesis space H finite, m i.i.d. samples, and $0 < \varepsilon < 1$, the probability that true error of any of the best hypothesis (i.e., lowest training error) is larger than its training error plus ε is at most $|H|e^{-2m\varepsilon^2}$

- **For hypothesis space H:**

$$P(error_{\mathcal{D}}(h_{best}) > \varepsilon + error_D(h_{best})) \leq |H|e^{-2m\varepsilon^2}$$

- **Sample Complexity:**

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

PAC bound and Bias-Variance Tradeoff

We can also express it in terms of ε:

$$\boldsymbol{P}(\boldsymbol{error_{\mathcal{D}}(h_{best}) - error_{D}(h_{best}) > \varepsilon}) \leq |H|e^{-2m\varepsilon^2} \leq \delta$$

$$\varepsilon \geq \sqrt{\frac{ln\frac{1}{\delta} + ln|H|}{2m}}$$

For all h, with probability at least $1 - \delta$:

$$\boldsymbol{error_{\mathcal{D}}(h_{best})} \leq \boldsymbol{error_{D}(h_{best})} + \sqrt{\frac{ln\frac{1}{\delta} + ln|H|}{2m}}$$

PAC bound and Bias-Variance Tradeoff

For all h, with probability at least $1 - \delta$:

$$\boldsymbol{error_D(h_{best})} \leq \underbrace{\boldsymbol{error_D(h_{best})}}_{\textbf{Bias}} + \underbrace{\sqrt{\frac{ln\frac{1}{\delta} + ln|H|}{2m}}}_{\textbf{Variance}}$$

- **For large |H|**
  - Low bias (lots of good hypotheses)
  - High variance (because bound is looser)
- **For small |H|**
  - High bias (may not be enough hypotheses to choose from)
  - Low variance

PAC bound and Bias-Variance Tradeoff

For all h, with probability at least $1-\delta$:

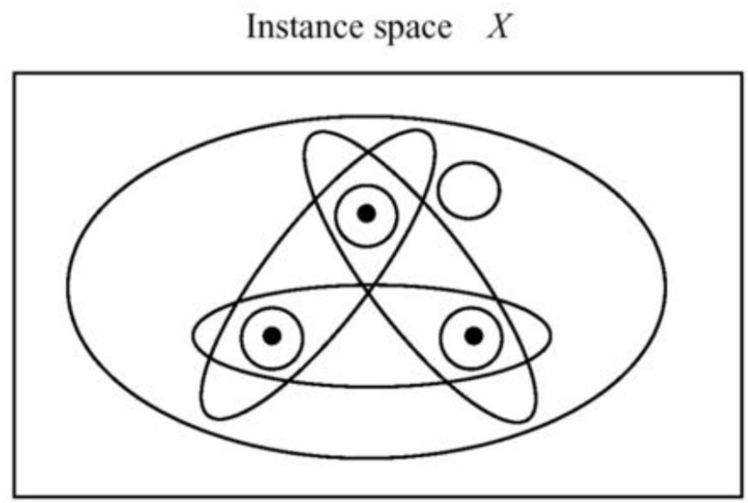
$$\boldsymbol{error_{\mathcal{D}}(h_{best})} \le \underbrace{\boldsymbol{error_{D}(h_{best})}}_{\textbf{Bias}} + \underbrace{\sqrt{\frac{ln\frac{1}{\delta} + ln|H|}{2m}}}_{\textbf{Variance}}$$

- **Bias:** how much the model's predictions deviate from the true underlying pattern or expected value of the target function.
- **High bias** means the model makes strong assumptions about the data (e.g., a linear model for a non-linear relationship), resulting in systematic errors across different datasets.

Infinite Hypothesis Spaces

- The preceding analysis was restricted to **finite hypothesis** spaces (based on $|H|$)
- Some infinite hypothesis spaces (such as those including real valued thresholds or parameters) are more expressive than others.
  - **Example:** Neural Nets can represent an infinite number of functions while Decision Trees with fixed depth are limited by the number of possible feature splits up to a specific depth.
- **We need some measure of the expressiveness of infinite hypothesis spaces.**
- The **Vapnik-Chervonenkis (VC)** dimension provides just such a measure, denoted VC(H).
  - **Measures the complexity of the hypothesis space H by the number of distinct instances from X that can be completely discriminated using H**
  - Analogous to ln|H|, there are bounds for sample complexity using VC(H).

Shattering a Set of Instances



Consider the following:

- We have subset of instances S ⊆ X.
- Each $h \in H$ imposes some dichotomy on S, i.e., partitions S into the two subsets $\{x \in S \mid h(x) = 0\}$ and $\{x \in S \mid h(x) = 1\}$.

- Given some instance set S, there are $2^{|S|}$ possible dichotomies, though H may be unable to represent some of these.

- **We say that H shatters S if every possible dichotomy of S can be represented by some hypothesis from H.**

Shattering a Set of Instances

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.

- If a set of instances is not shattered by a hypothesis space, then there must be some concept (dichotomy) that can be defined over the instances, but that cannot be represented by the hypothesis space.
- The ability of H to shatter a set of instances is thus a measure of its capacity to represent target concepts defined over these instances.
- An **unbiased** hypothesis space shatters the entire instance space.
  - The larger the subset of X that can be shattered, the more expressive the hypothesis space is, i.e., the less biased.

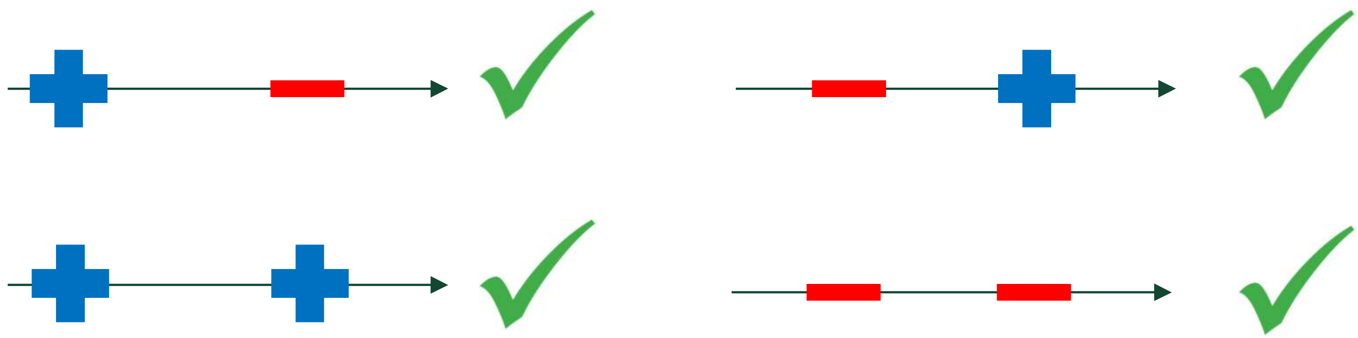
Vapnik-Chervonenkis Dimension

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 $VC(H) \equiv \infty$.

- For any finite H, $VC(H) \le log_2|\mathrm{H}|$.
  - Suppose that VC(H) = d.
  - Then H will require 2^d distinct hypotheses to shatter d instances.
  - Hence, $2^d \le log_2|\mathrm{H}|$, and $\mathrm{d} = \mathrm{VC(H)} \le log_2|\mathrm{H}|$.

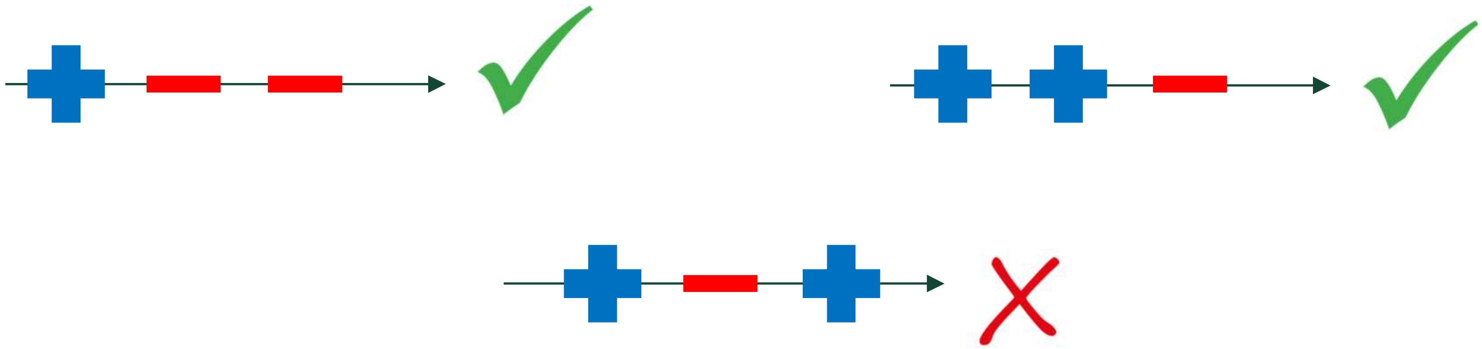
VC Dimension - Examples

- How many points in 1-D can be correctly classified by a linear separator?
    - **Two points:**



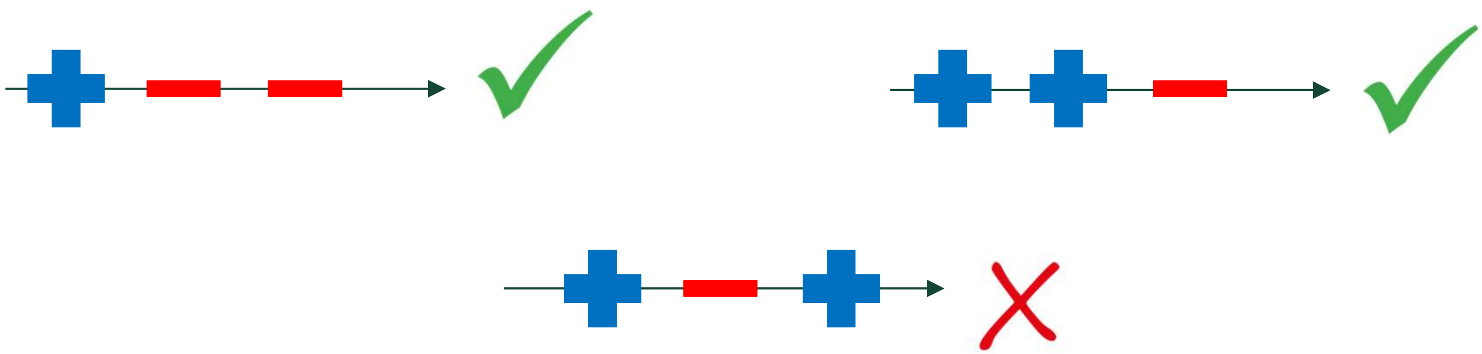
VC Dimension

- How many points in 1-D can be correctly classified by a linear separator?
  - **Three points:**



VC Dimension

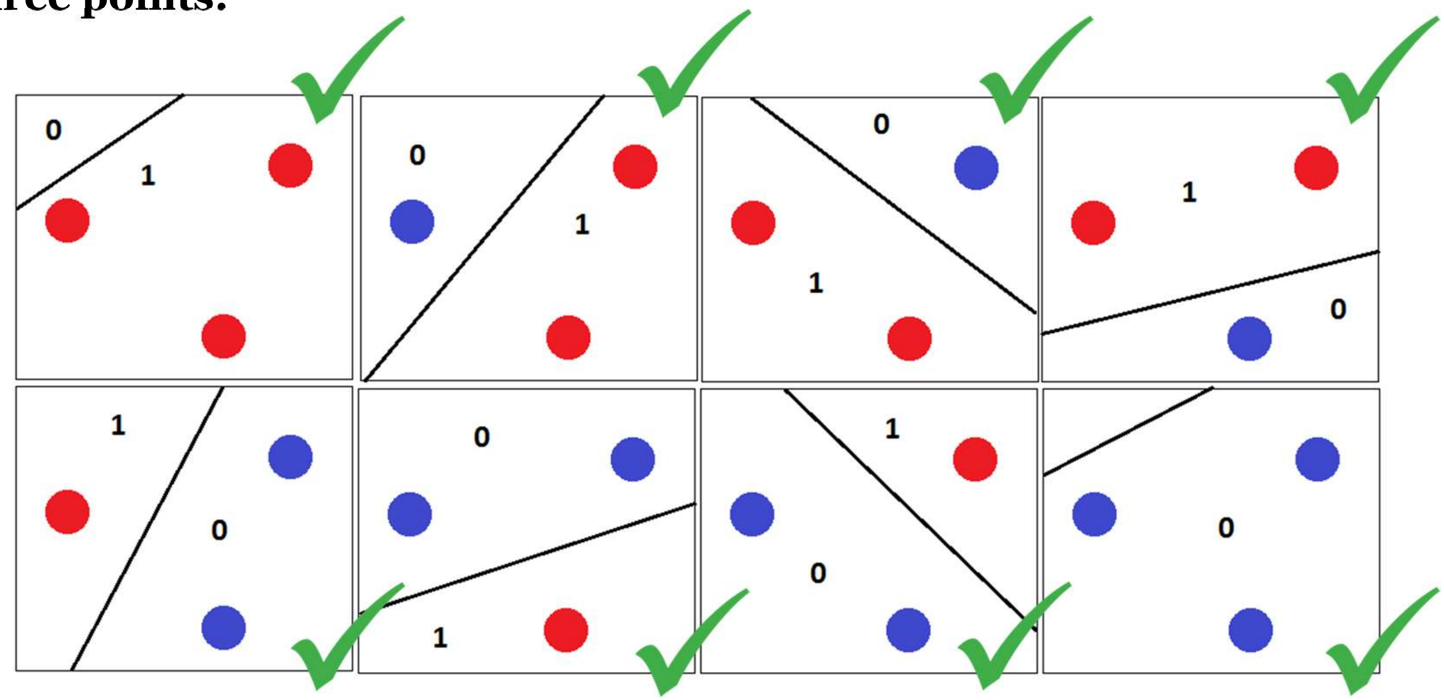
- How many points in 1-D can be correctly classified by a linear separator?
    - **Three points:**



- **3 points and up:** for any collection of three or more there is always some choice of pluses and minuses such that the points cannot be classified with a linear separator (in one dimension).

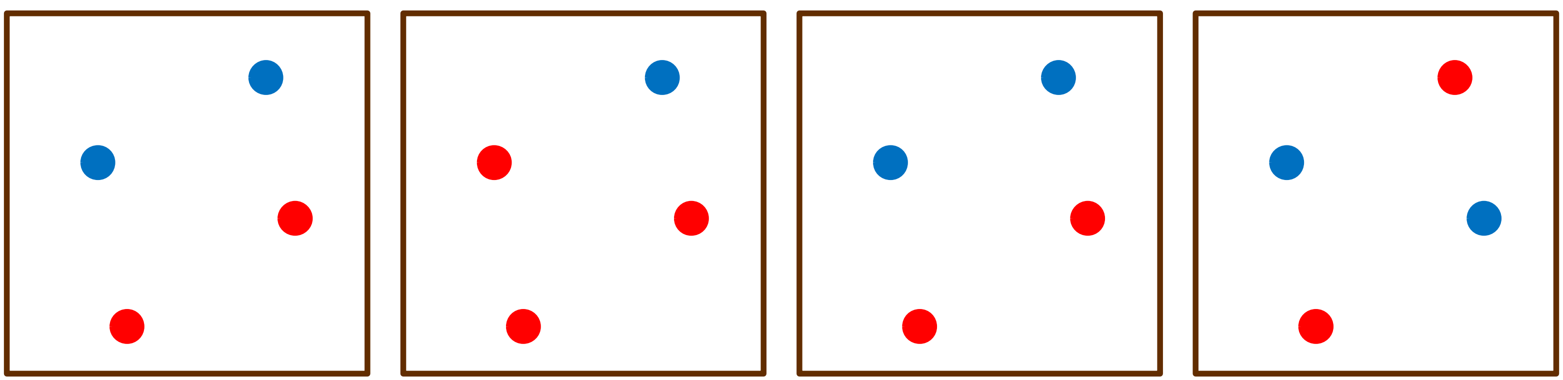
VC Dimension

- What is the VC dimension of 2-D space under linear separators?
  - **Three points:**



VC Dimension

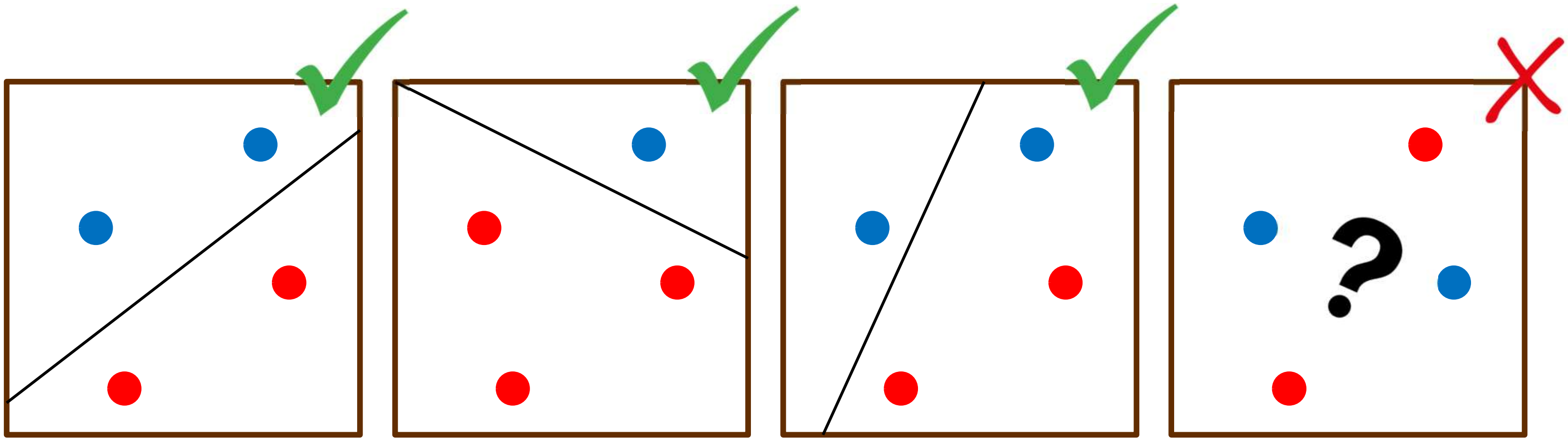
- What is the VC dimension of 2-D space under linear separators?
  - **Can some set of four points be shattered?**



VC Dimension

- What is the VC dimension of 2-D space under linear separators?
  - **Can some set of four points be shattered?**

NO! So, the VC dimension is at most 3

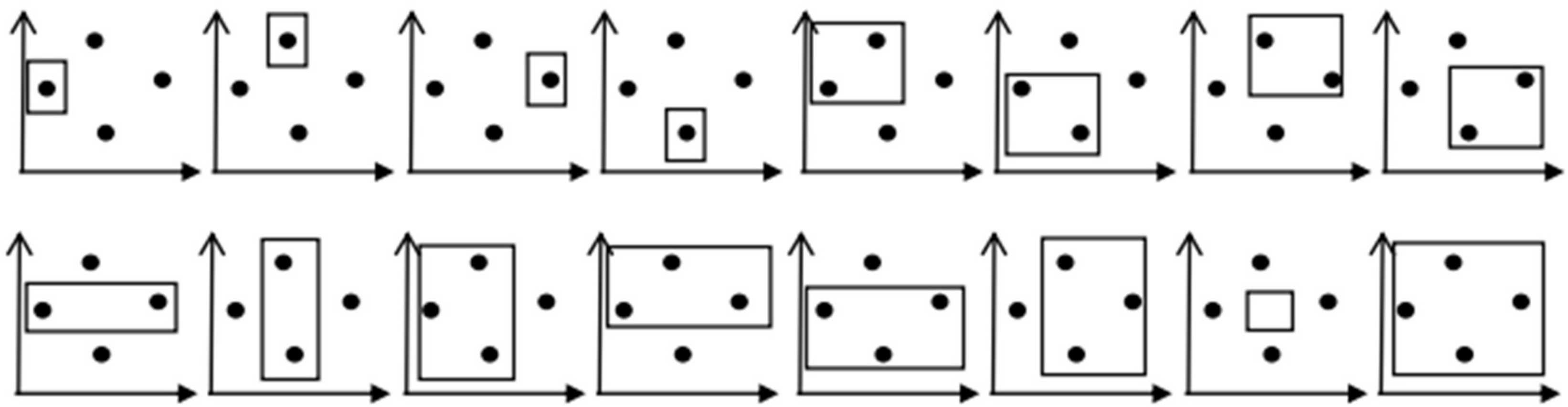


VC Dimension

- There exists a **set of size d+1** in a **d–dimensional space** that can be shattered by a **linear separator**, <u>but not a set of size d+2</u>
- **The larger the subset of X that can be shattered, the more expressive the hypothesis space is.**

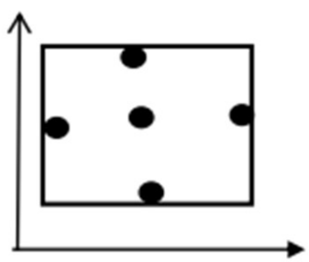
VC Dimension - Axis Parallel Rectangles

- Consider axis-parallel rectangles in the real-plane, i.e., conjunctions of intervals on two real-valued features. Some 4 instances can be shattered.



VC Dimension - Axis Parallel Rectangles

- No five instances can be shattered since there can be at most 4 distinct extreme points (min and max on each of the 2 dimensions) and these 4 cannot be included without including any possible 5th point.



- Therefore $VC(H) = 4$
- Generalizes to axis-parallel hyper-rectangles (conjunctions of intervals in n dimensions): $VC(H) = 2n$.

VC Dimension

Other Examples

- VC dimension of one-level decision trees over real vectors of length 2?
  - **Three**
- VC dimension of linear separators through the origin?
  - **Two**
- VC Dimension of axis-aligned triangles n 2D?
  - **Four**

VC Dimension

VC dimension of a hypothesis H equals to n if:

- **ANY** arrangement of n points can be shattered
- **NO** arrangement of n+1 points can be shattered.

- **Note: you do not need to shatter all arrangements of size n**
- Just showing one arrangement can be shattered is enough.

Sample Complexity and the VC Dimension

- Previously, we have asked "*How many randomly drawn training examples suffice to probably approximately learn any target concept in C?*"
    - How many examples suffice to ε-exhaust the version space with probability $(1-\delta)$?
- Using VC(H) as a measure for the complexity of H, it is possible to derive an alternative answer to this question…
- This new bound (*Blumer et al. 1989*) is:

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

Upper Bound on Sample Complexity with VC

- Sample Complexity using VC(H):

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

- Compared to the previous result using $ln|\mathrm{H}|$, this bound has some **extra constants** and an ***extra*** $\boldsymbol{log_2\left(\frac{1}{\varepsilon}\right)}$ factor. Since $\mathrm{VC(H)} \leq log_2|H|$, this can provide a tighter upper bound on the number of examples needed for PAC learning.

No Free Lunch Theorem

- There is no single algorithm or technique that is best for all situations and data sets
- If an algorithm is highly effective on certain types of problems, it likely will perform poorly on others.
- Select models based on the characteristics of the data and problem, rather than assuming one model is best in all situations.
  - **Data Size:** KNN or NB (small data) vs ANN or Ensemble (large data)
  - **Dimensionality:** Linear models (high) vs DT or KNN (low)
  - **Noise and Outliers:** Random Forest (robust) vs KNN or LR (sensitive to noise)
  - **Linearity:** LR or SVM vs DT and ANN
  - **Interpretability**: DT and Linear models are easily explainable
  - **Training time Constraint**: NB and LR (fast train) vs ANN and Ensemble (more time)

Computational Learning Theory

- The PAC model provides a theoretical framework for analyzing the effectiveness of learning algorithms.
- The sample complexity for any consistent learner using some hypothesis space H can be determined from a measure of its expressiveness |H| or VC(H), quantifying bias and relating it to generalization.
- If sample complexity is tractable, then the computational complexity of finding a consistent hypothesis in H governs its PAC learnability.
- Experimental results suggest that theoretical sample complexity bounds over-estimate the number of training instances needed in practice since they are worst-case upper bounds.

Readings

- **Machine Learning by Tom Mitchell – Chapter 7**