Never Stand Still

Learning Theory

COMP9417 Machine Learning & Data Mining

Term 3, 2019

Adapted from slides by Dr Michael Bain

Aims

This lecture will introduce you to some foundational results that apply in machine learning irrespective of any particular algorithm, and will enable you to define and reproduce some of the fundamental approaches and results from the computational and statistical theory. Following it you should be able to:

- describe a basic theoretical framework for sample complexity of learning
- describe the Probably Approximately Correct (PAC) learning framework
- describe the Vapnik-Chervonenkis (VC) dimension framework

Introduction

Machine learning: Have a computer solve problems by learning from data rather than being explicitly programmed.

- Regression
- Classification
- Clustering
- Ranking
- Reinforcement learning
- …

Computational Learning Theory

The goal of learning theory is to develop and analyse formal models that help us understand:

- what concepts we can hope to learn efficiently, and how much data is necessary to learn them
- what types of guarantees we might hope to achieve (error bounds, complexity bounds)
- why particular algorithms may or may not perform well under various conditions

Key Idea

Leaning theory aims at a body of theory that captures all important aspects of the fundamentals of the learning process and any algorithm or class of algorithms designed to do learning — i.e., we desire theory to capture the algorithm-independent aspects of machine learning.

BUT: we're not quite there yet ...

Computational Learning Theory

Inductive learning: learning from examples and all machine learning algorithms are a kind of inductive learning and there are some questions that we are interested to be able to answer in such framework:

- Probability of successful learning
- Number of training examples
- Complexity of hypothesis space
- Time complexity of learning algorithm
- Accuracy to which target concept is approximated
- Manner in which training examples presented

Computational Learning Theory

Some questions to ask, without focusing on any particular algorithm:

- Sample complexity
  - How many training examples required for learner to converge (with high probability) to a successful hypothesis ?
- Computational complexity
  - How much computational effort required for learner to converge (with high probability) to a successful hypothesis ?
- Hypothesis complexity
  - How do we measure the complexity of a hypothesis ?
  - How large is a hypothesis space?

Computational Learning Theory

What do we consider to be a successful hypothesis:

- identical to target concept ?
- mostly agrees with target concept ... ?
- ... does this most of the time ?

Computational Learning Theory

Instead of focusing on particular algorithms, learning theory aims to characterize classes of algorithms.

Probably Approximately Correct (PAC) is a framework for mathematical analysis of learning which was proposed by Valiant in 1948.

Probably Approximately Correct (PAC)

The framework of Probably Approximately Correct (PAC) learning can be used to address questions such as:

- How many training examples?
- How much computational effort required?
- How complex a hypothesis class needed?
- How to quantify hypothesis complexity?
- How many mistakes will be made ?

Sample Complexity

Given:

- set of instances X
- set of hypotheses H
- set of possible target concepts C
- training instances generated by a fixed, unknown probability distribution $\mathcal{D}$ over X

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

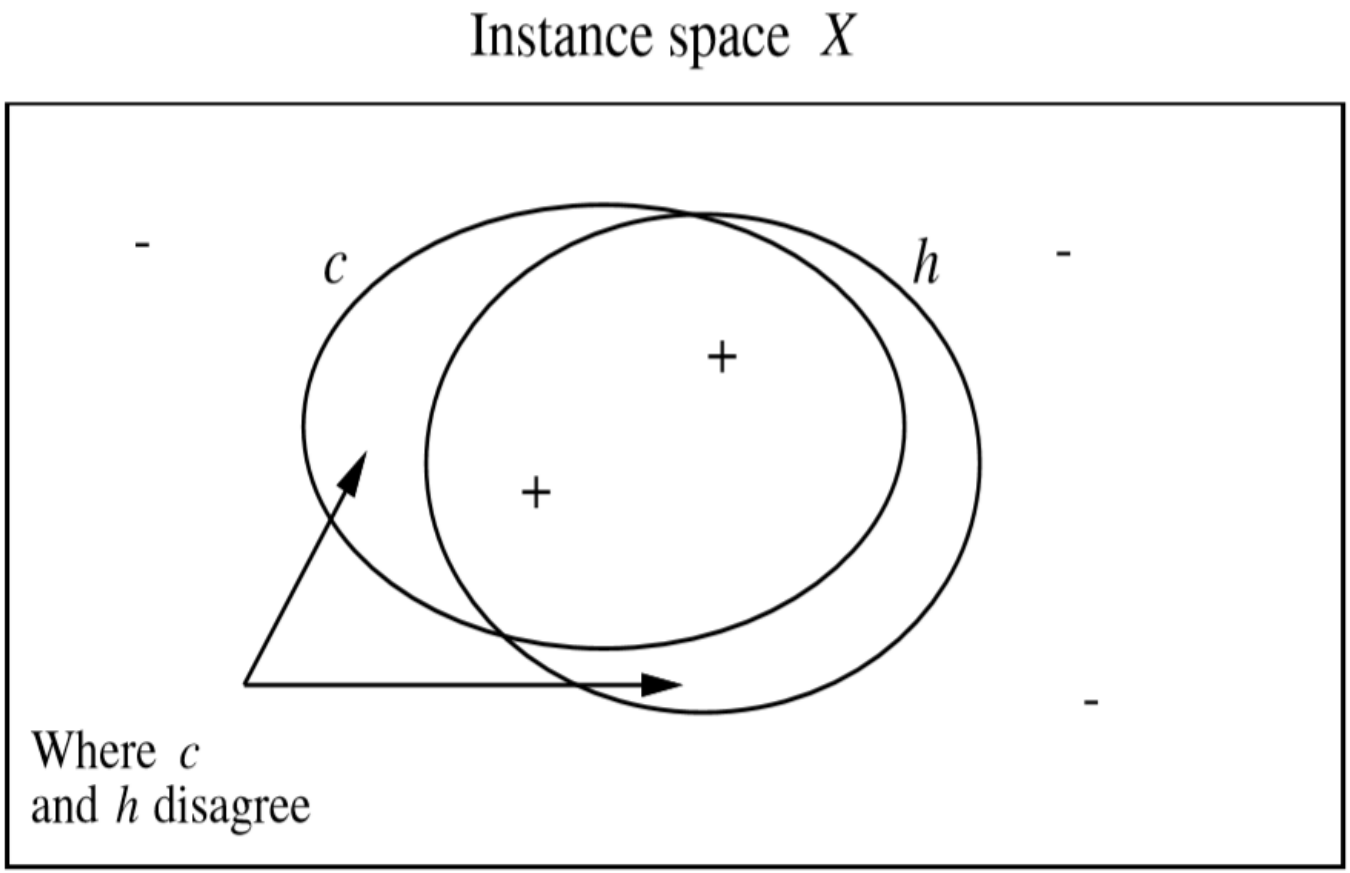
- Instances x are drawn from distribution D
- teacher provides target value $c(\mathrm{x})$ for each x

Learner must output a hypothesis h estimating c

- h is evaluated by its performance on subsequent instances drawn according to $\mathcal{D}$

Note: randomly drawn instances, noise-free classifications

True Error of a Hypothesis



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 $\mathcal{D}$.

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

Two Notions of Error

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

- How often $h(\mathrm{x}) \neq c(\mathrm{x})$ over training instances

True error of hypothesis h with respect to c:

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

Our concern:

- Can we bound the true error of h given the training error of h?
- First consider when training error of h is zero (i.e., $h \in V\ S_{H,D}$)

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 hypothesis space H.

Definition: 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 $0 < \delta < 1/2$, 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 $size(c)$.

Probably **A**pproximately **C**orrect **L**earning

L. Valiant, (1984; 2013).

PAC Learning

Given:

- C: concept class
- L : learner
- H: hypothesis space
- n: length of instances
- $|H|$, size of hypothesis space
- $\mathcal{D}$: distribution over inputs
- ϵ: error goal $(0 < \epsilon < ½)$
- δ: certainty goal $(0 < \delta < ½)$

C is PAC-learnable by L using H, if learner L, with probability $(1 - \delta)$, output a hypothesis $h \in H$, such that $error_{\mathcal{D}}(h) \leq \epsilon$ in time and sample polynomial in $1/\epsilon$, $1/\delta$, n and $size(c)$.

We start to look at PAC learning using Concept Learning.

Prototypical Concept Learning Task

Concept

	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Warm	Change	Yes

- A set of example days, and each is described by six attributes.
- The task is to learn to predict the value of EnjoySport for arbitrary day, based on the values of its attribute values.

Prototypical Concept Learning Task

Example :

Instances X: Possible days, each described by the attributes

Sky, AirTemp, Humidity, Wind, Water, *Forecast*

Target function c: *EnjoySport* : $X \rightarrow \{0,1\}$

Hypotheses H: Conjunctions of literals.

Training examples D: Positive and negative examples of target function

$$\langle \mathrm{x}_1, c(\mathrm{x}_1) \rangle, \ldots, \langle \mathrm{x}_m, c(\mathrm{x}_m) \rangle$$

Determine:

because the true is different

A hypothesis h in H such that $h(\mathrm{x}) = c(\mathrm{x})$ for all x in D?

A hypothesis h in H such that $h(\mathrm{x}) = c(\mathrm{x})$ for all x in X?

Concept Learning

Concept learning: can be formulated as:

"*problem of searching through a predefined space of potential hypothesis that best fits the training examples*"

Tom Mitchell

"*the search for and listing of attributes that can be used to distinguish exemplars from non exemplars of various categories*"

Bruner, Goodnow, & Austin (1967)

Concept Learning: Acquiring the definition of a general category from given positive and negative training examples of the category.

Concept Learning

Hypothesis representation for concept learning task:

- Each hypothesis consists of a conjunction of constraints on the instance attributes.
- Each hypothesis will be a vector of six constraints, specifying the values of the six attributes
- Each attribute will be:
  - ? - indicating any value is acceptable for the attribute (don't care)
  - single value – specifying a single required value (ex. Warm) (specific)
  - ∅ - indicating no value is acceptable for the attribute (no value)

Concept Learning

- *EnjoySport* concept learning task requires learning the sets of days for which $EnjoySport = yes$, describing this set by a conjunction of constraints over the instance attributes.
- One example of hypothesis:

  $< Sky = Sunny, AirTemp = ?, Humidity = ?, Wind = Strong, Water = ?, Forecast = same >$
- The most general hypothesis:

  $< Sky = ?, AirTemp = ?, Humidity = ?, Wind = ?, Water = ?, Forecast = ? >$
- The most specific hypothesis:

$< Sky = \emptyset, AirTemp = \emptyset, Humidity = \emptyset, Wind = \emptyset, Water = \emptyset, Forecast = \emptyset >$

Concept Learning as Search

Question: What can be learned?

Answer: (only) what is in the hypothesis space

- Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation.
- The goal of this search is to find the hypothesis that best fits the training examples.

EnjoySport - Hypothesis Space

How big is the hypothesis space for *EnjoySport* ?

- Instance space

$$Sky \times AirTemp \times \cdots \times Forecast = 3 \times 2 \times 2 \times 2 \times 2 \times 2$$
$$= 96 \text{ instances}$$

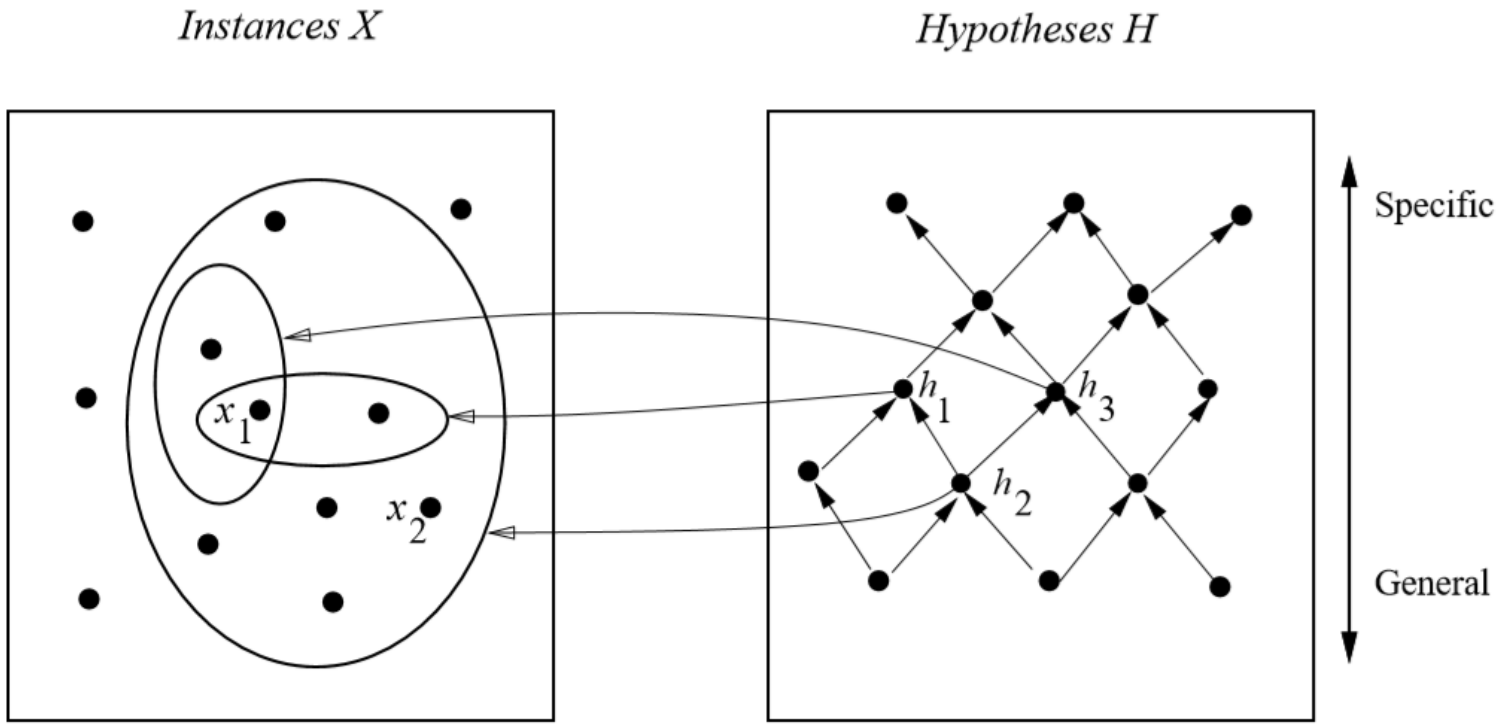
- Hypothesis space

$$Sky \times AirTemp \times \cdots \times Forecast = 5 \times 4 \times 4 \times 4 \times 4 \times 4$$
$$= 5120 \text{ synthetically distinct hypothesis}$$

$$Sky \times AirTemp \times \cdots \times Forecast = 1 + 4 \times 3 \times 3 \times 3 \times 3 \times 3$$
$$= 973 \text{ semantically distinct hypothesis}$$

(Any hypothesis with an ∅ constraint covers no instances, hence all are semantically equivalent)

Instances, Hypotheses, and More-General-Than



x_1 = <Sunny, Warm, High, Strong, Cool, Same>

x_2 = <Sunny, Warm, High, Light, Warm, Same>

h_1 = <Sunny, ?, ?, Strong, ?, ?>

h_2 = <Sunny, ?, ?, ?, ?, ?> more general

h_3 = <Sunny, ?, ?, ?, Cool, ?>

A generality order on hypotheses

Definition: Let h_j and h_k be Boolean-valued functions defined over instances X. Then h_j is **more-general-than-or-equal-to** h_k (written $h_j \geq_g h_k$) if and only if

$$(\forall x \in X)[(h_k(\mathrm{x}) = 1) \rightarrow (h_j(\mathrm{x}) = 1)]$$

Intuitively, h_j is **more-general-than-or-equal-to** h_k if any instance satisfying h_k also satisfies h_j.

h_j is (strictly) more general than h_k (written $h_j >_g h_k$) if and only if $(h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$.

h_j is more specific than h_k when k is more general than h_j.

Version Space

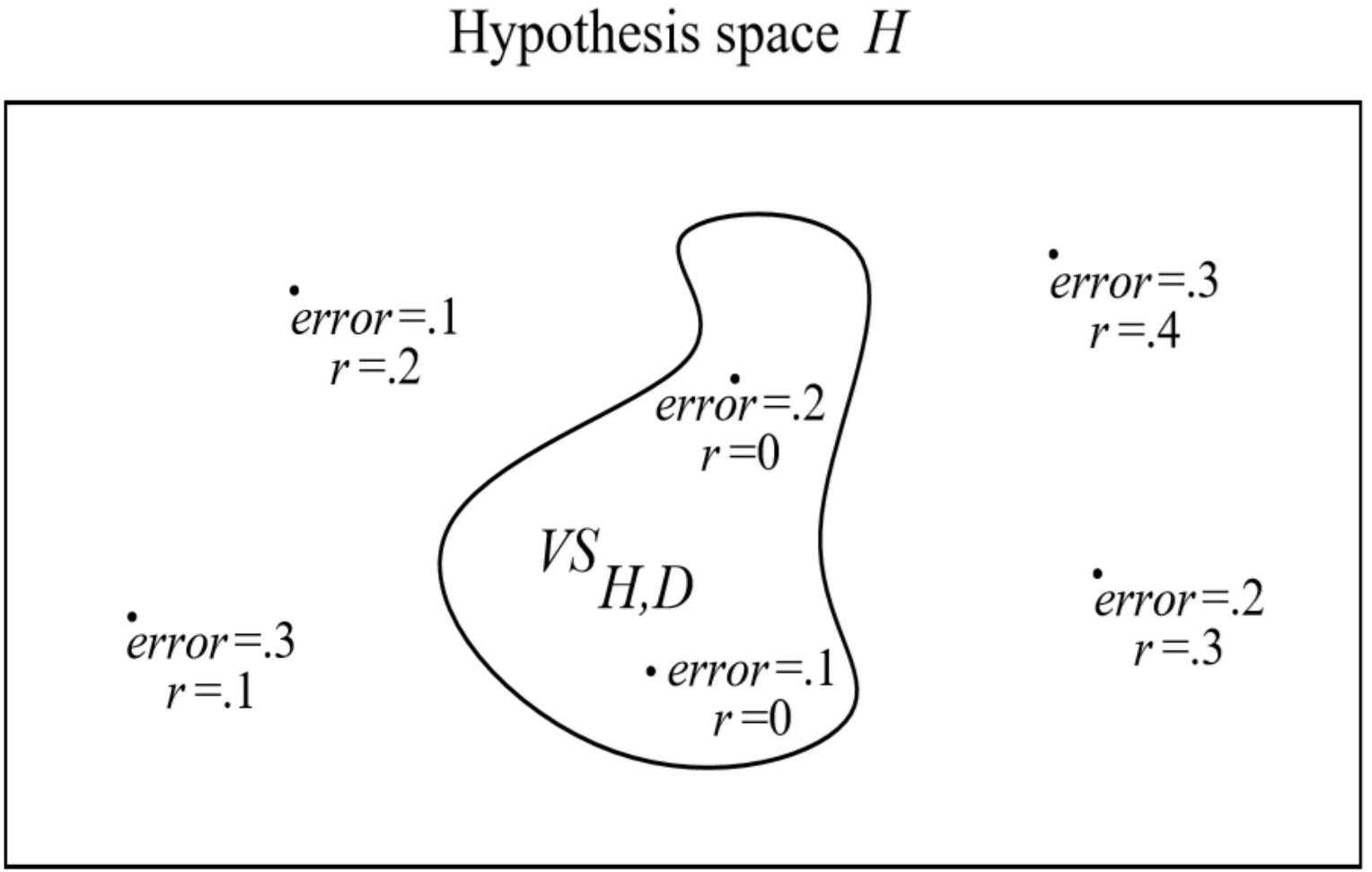
A hypothesis h is consistent with a set of training examples D of target concept c if and only if $h(\mathrm{x}) = c(\mathrm{x})$ for each training example $\langle \mathrm{x}, c(\mathrm{x}) \rangle$ in D.

$$Consistent(h, D) \equiv (\forall \langle \mathrm{x}, c(\mathrm{x}) \rangle \in D)\, h(\mathrm{x}) = c(\mathrm{x})$$

The **version space**, $V\,S_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D.

$$V\,S_{H,D} \equiv \{h \in H \mid Consistent(h, D)\}$$

Exhausting the Version Space



Exhausting the Version Space

Note: in the diagram

$$(r = \text{training error}, error = \text{true error})$$

Definition: The version space $VS_{H,D}$ is said to be ϵ-exhausted with respect to c and $\mathcal{D}$, if every hypothesis h in $VS_{H,D}$ has error less than ϵ with respect to c and $\mathcal{D}$.

$$(\forall\ h \in V\ S_{H,D})\ error_{\mathcal{D}}(h) < \epsilon$$

So $VS_{H,D}$ is not ϵ-exhausted if it contains at least one h with $error_{\mathcal{D}}(h) \geq \epsilon$

How many examples will -exhaust the VS?

Theorem [Haussler, 1988].

If the hypothesis space H is finite, and D is a sequence of $m \geq 1$ independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to H and D is not ϵ-exhausted (with respect to c) is less than:

$$|H|e^{-\epsilon m}$$

Interesting! This bounds the probability that any consistent learner will output a hypothesis h with $error(h) \geq \epsilon$.

How many examples will ϵ-exhaust the VS?

If we want this probability to be below δ

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

then

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

How many examples will ϵ-exhaust the VS?

How many examples are sufficient to assure with probability at least $(1-\delta)$ that every h in $V\ S_{H,D}$ satisfies $error_{\mathcal{D}}(h) \leq \epsilon$?

Use our theorem:

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

How About *EnjoySport*?

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

If H is as given in *EnjoySport* then $|H| = 973$, and

$$m \geq \frac{1}{\epsilon}\left(\ln 973 + \ln\left(\frac{1}{\delta}\right)\right)$$

How About *EnjoySport*?

... if want to assure that with probability 95%, VS contains only hypotheses with $error_{\mathcal{D}}(h) \leq 0.1$, then it is sufficient to have m examples, where

$$m \geq \frac{1}{0.1}\left(\ln 973 + \ln\left(\frac{1}{0.05}\right)\right)$$

$$m \geq 10\big(\ln 973 + \ln(20)\big)$$

$$m \geq 10(6.88 + 3)$$

$$m \geq 98.8$$

Agnostic PAC Learning

So far, assumed $c \in H$ — consistent learners (noise-free)

Agnostic learning setting: don't assume $c \in H$

- What do we want then?
  - The hypothesis h that makes fewest errors on training data

Hoeffding bounds:

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

For hypothesis space H:

$$Pr[error_{\mathcal{D}}(h_{best}) > error_{D}(h_{best}) + \epsilon] \leq |H|e^{-2m\epsilon^2}$$

- What is sample complexity in this case?

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

PAC Learning

So far, we have only considered cases with finite hypothesis space, however in reality many of the hypothesis spaces that we use have infinite dimension.

…what can we do/say about such hypothesis spaces?

Hypothesis Space Size

Question:

Which hypothesis spaces are infinite and which are finite?

- ❑ Decision trees (with discrete inputs)
- ❑ Neural networks
- ❑ Linear classifiers
- ❑ Decision trees (with continuous inputs)

Learners and Complexity

- Different learners have different complexity
- Complexity relates to the “representational power”
- Usual trade-off:
  - More power = represent more complex systems, may overfit
  - Less power: won’t overfit, but may not find “best learner”

Question: How can we quantify complexity?

  - Not easily…
  - One solution is VC (Vapnik-Chervonenkis) dimension

Learners and Complexity

- How complexity relates to PAC learning?
  - We want to be able to say something about the true error of the hypothesis based on the training error of the hypothesis
  - We know that:
    - in underfitting domain: those two errors are pretty similar
    - In overfitting domain: test error (which is a representation of true error) might be much worse!

Shattering a Set of Instances

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

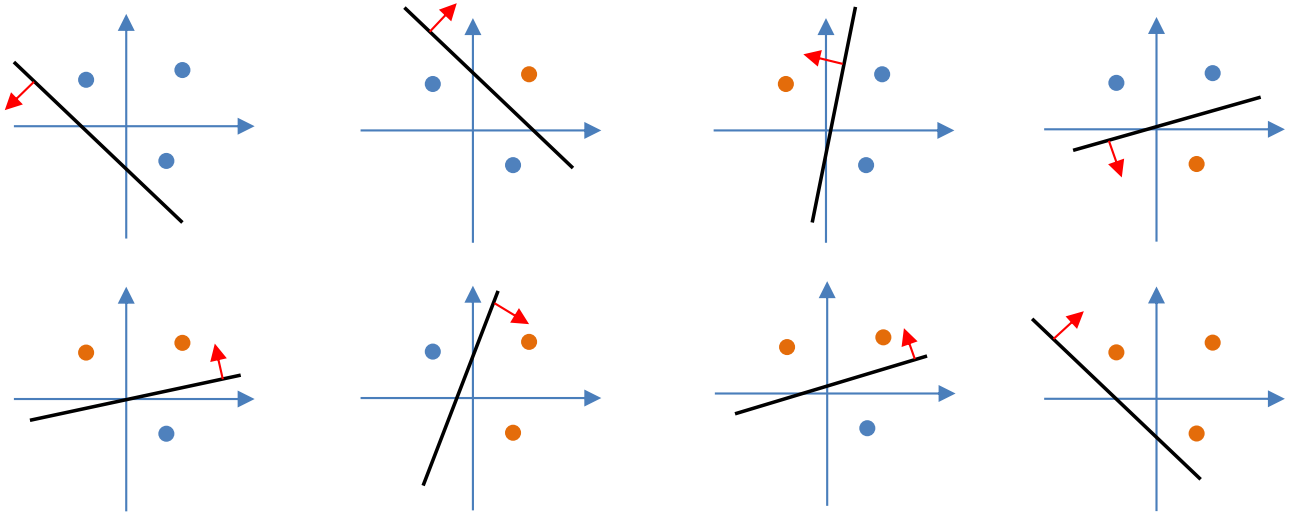
Definition: *a set of instances S is **shattered** by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy.*

Shattering

We say a classifier $f(\mathrm{x})$ can shatter points $\mathrm{x}_1, \ldots, \mathrm{x}_m$ iff for all possible $y_1, \ldots, y_m$, $f(\mathrm{x})$ can achieve zero error on training data $(\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_m, y_m)$. (i.e., there exist some θ that gets zero error)

Example:

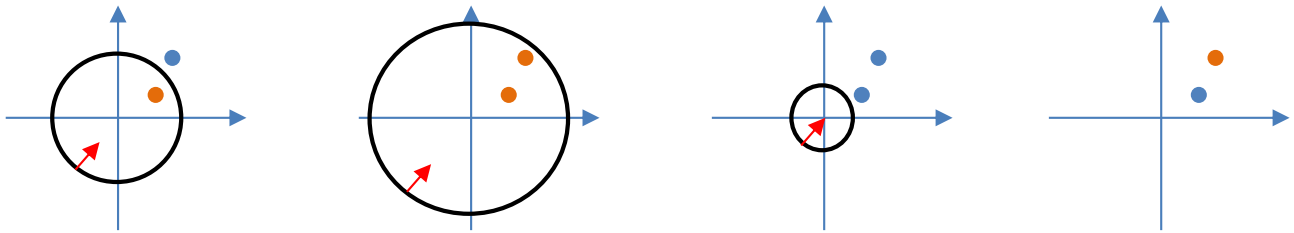
Can $f(\mathrm{x}) = \mathrm{sign}(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$ shatter these points?



Shattering

Example:

Can $f(\mathrm{x}) = \mathrm{sign}(\mathrm{x}^T\mathrm{x} - \theta)$ shatter these points?



This configuration can not be classified by $f(\mathrm{x})$

This particular classifier *can not shatter* these two points.

The Vapnik-Chervonenkis Dimension

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

VC dimension: the largest set of inputs that the hypothesis class can shatter (label in all possible ways).

Note: the VC dimension can be defined for an infinite hypothesis space H since it depends only on the size of finite subsets of X.

The Vapnik-Chervonenkis Dimension

Example:

$X = \{1, \ldots, 10\}$

$H = \{h(\mathrm{x}) = \mathrm{x} \geq \theta\}$

For one input (e.g. $S = \{6\}$), the hypothesis class can label it in all possible ways.

What about any pair of input (e.g. $S = \{5,7\}$)?

- No. So, $VC\ dimension = 1$

Although H, here, is an infinite hypothesis space, but it is a weak hypothesis space and not very expressive!

The Vapnik-Chervonenkis Dimension

Question:

$X = \mathbb{R}$

$H = \{h(\mathrm{x}) = \mathrm{x} \in [a, b]\}$

- What is the size of parameter?
- What is the size of hypothesis space?
- What is the size of $VC\ dimension$?

The Vapnik-Chervonenkis Dimension

Question:

$X = \mathbb{R}$

$H = \{h(\mathrm{x}) = \mathrm{x} \in [a, b]\}$

- What is the size of parameter? 2 parameters a and b
- What is the size of hypothesis space? infinite
- What is the size of $VC\ dimension$? $VC\ dimension = 2$

The Vapnik-Chervonenkis Dimension

Question:

$X = \mathbb{R}^2$

$H = \{h(\mathrm{x}) = w^T \mathrm{x} \geq \theta\}$

- What is the size of parameter?
- What is the size of hypothesis space?
- What is the size of $VC\ dimension$?

The Vapnik-Chervonenkis Dimension

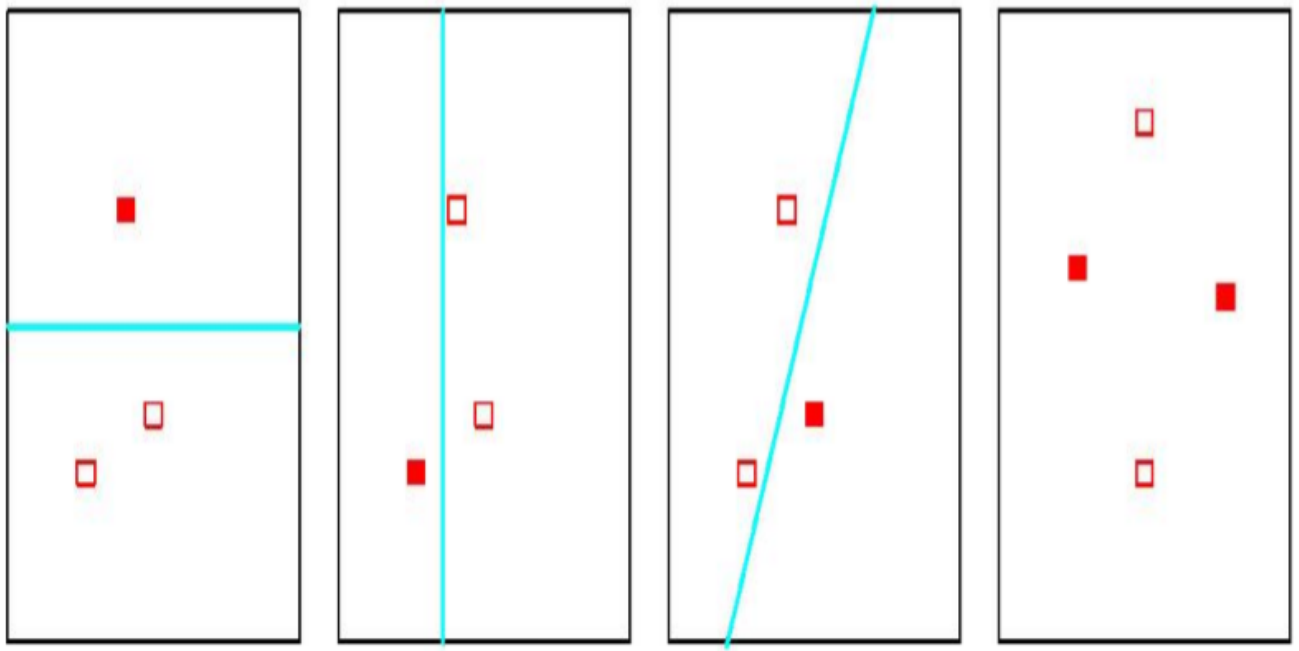
Question:

$X = \mathbb{R}^2$

$H = \{h(\mathrm{x}) = w^T \mathrm{x} \geq \theta\}$

- What is the size of parameter? 3 parameters
- What is the size of hypothesis space? infinite
- What is the size of $VC\ dimension$? See next 3 slides

VC Dimension of Linear Decision Surfaces



From the left, shown are three dichotomies of the same three instances. Can a linear classifier be found for the other five dichotomies? On the right, this set of four instances clearly cannot be shattered by a hypothesis space of linear classifiers.

VC Dimension of Linear Decision Surfaces

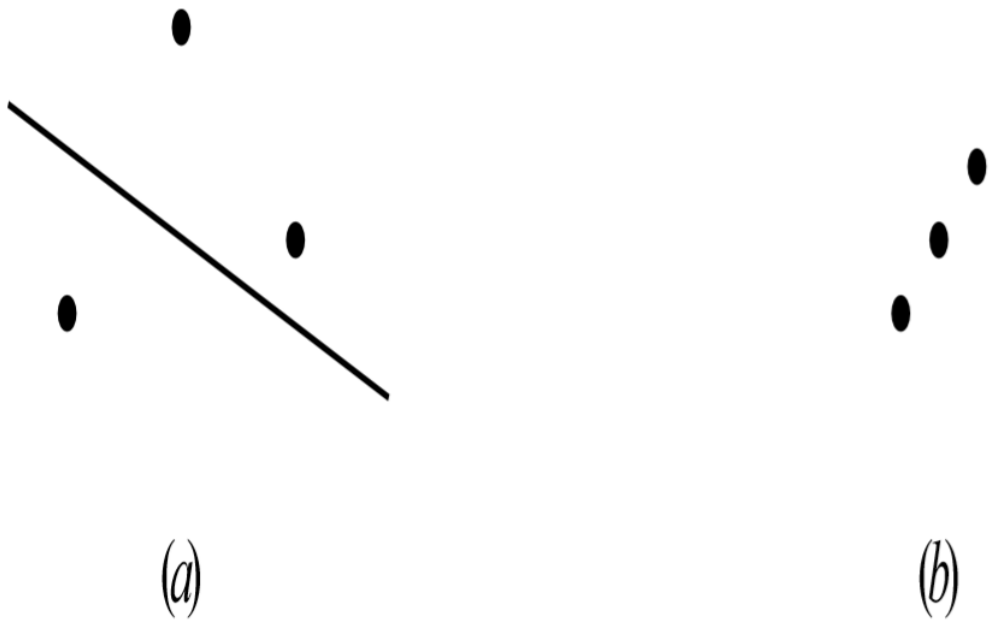
Clearly, for a subset of 2 instances we can find a linear classifier for all possible dichotomies.

The same argument as for 2 instances applies (see first three examples on previous slide, and case (a) on next slide), as long as the instances are not collinear (case (b) on next slide). So the VC dimension is at least 3.

However, in this setting, there is no set of 4 points that can be shattered.

In general, for linear classifiers in d dimensions (d is number of features) the VC dimension is $d + 1$.

VC Dimension of Linear Decision Surfaces



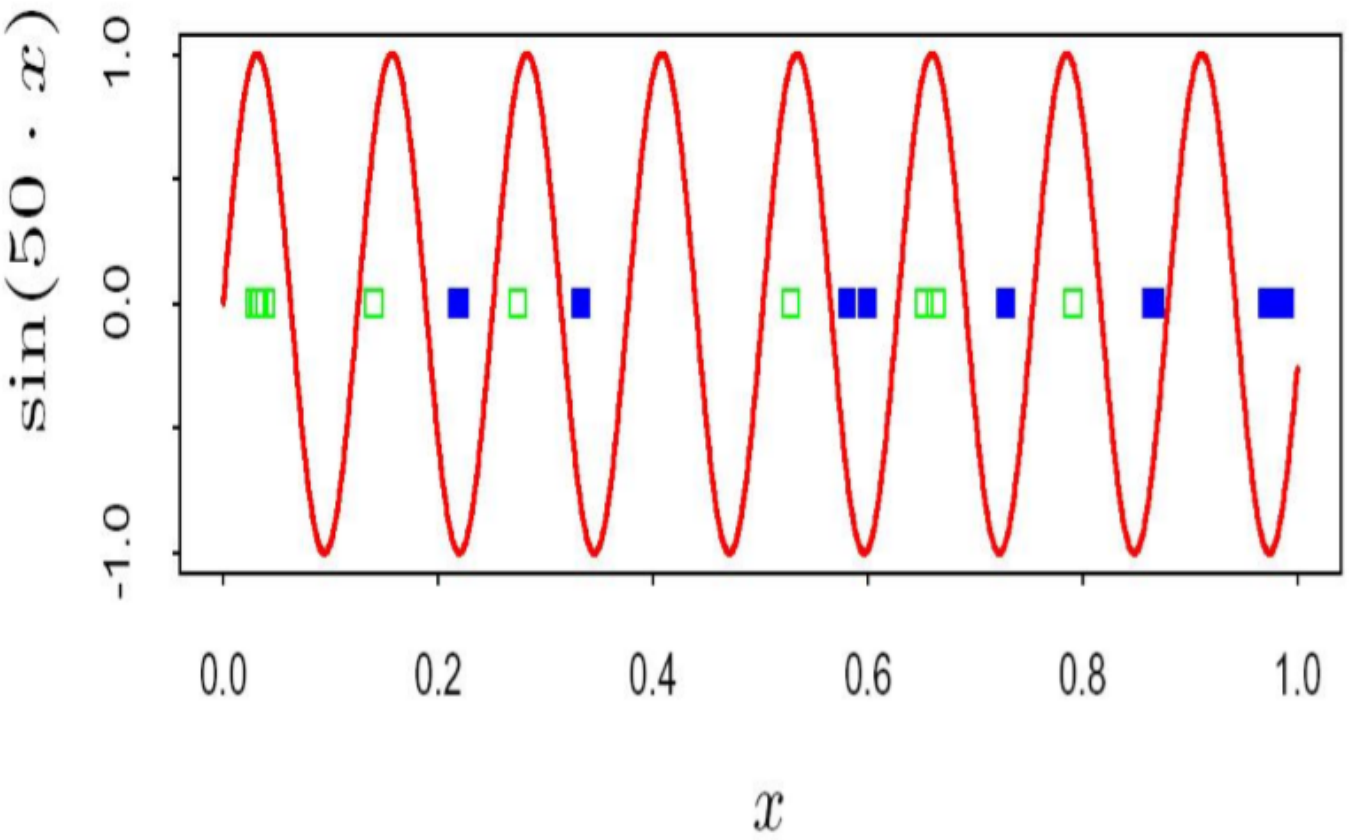
VC dimension and Number of parameters

Note that VC dimension does not necessarily equal the number of parameters!

Number of parameters does not necessarily equal complexity:

- Can define a classifier with lot of parameters but not much power!
- Can define a classifier with one parameter, but lots of power! (look at the example in next slide)

How *Complex* is a Hypothesis ?



How Complex is a Hypothesis ?

The solid curve is the function $\sin(50x)$ for $x \in [0,1]$.

The blue (solid) and green (hollow) points illustrate how the associated indicator function $I(\sin(\alpha\, x) > 0)$ can shatter (separate) an arbitrarily large number of points by choosing an appropriately high frequency α.

Classes separated based on $\sin(\alpha\, x)$, for frequency α, a single parameter.

Sample Complexity from VC Dimension

We can now generalize the PAC-learning result obtained earlier to answer the question: how many randomly drawn examples suffice to ϵ-exhaust $VS_{H,D}$ with probability at least $(1-\delta)$?

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

So we see that the concept of the VC dimension of a hypothesis class gives us a general framework for characterizing the complexity or capacity of hypotheses in terms of their ability to express all possible target concepts in a particular learning setting.

Sample Complexity and VC dimension

- For finite hypothesis space:

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

- For infinite hypothesis space with finite $VC\ dimension$:

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

VC dimension of finite H

If $d = VC(H)$, then in means that there are at least 2^d hypothesis in the hypothesis space that can shatter d points in all possible ways (which is 2^d), so:

$$2^d < |\mathrm{H}|$$

$$d < \log_2 |H|$$

Theorem:

H is is PAC-learnable *if and only* if VC dimension is finite.

Summary

- PAC learning
- Sample complexity
- Version space
- VC dimension and hypothesis space complexity
- VC dimension capture PAC-learnability

Acknowledgements

- Material derived from slides for the book “Elements of Statistical Learning (2nd Ed.)” by T. Hastie, R. Tibshirani & J. Friedman. Springer (2009) http://statweb.stanford.edu/~tibs/ElemStatLearn/
- Material derived from slides for the book “Machine Learning: A Probabilistic Perspective” by P. Murphy MIT Press (2012) http://www.cs.ubc.ca/~murphyk/MLbook
- Material derived from slides for the book “Machine Learning” by P. Flach Cambridge University Press (2012) http://cs.bris.ac.uk/~flach/mlbook
- Material derived from slides for the book “Bayesian Reasoning and Machine Learning” by D. Barber Cambridge University Press (2012) http://www.cs.ucl.ac.uk/staff/d.barber/brml
- Material derived from slides for the book “Machine Learning” by T. Mitchell McGraw-Hill (1997) http://www-2.cs.cmu.edu/~tom/mlbook.html
- Material derived from slides for the course “Machine Learning” by A. Srinivasan BITS Pilani Goa Campus, India (2016)