# Learning Theory

## Preview

- "No free lunch" theorems
- Bias and variance
- PAC learning
- VC dimension
- Support vector machines

## "No Free Lunch" Theorems

$Acc_G(L)$ = Generalization accuracy of learner $L$
= Accuracy of $L$ on non-training examples
$\mathcal{F}$ = Set of all possible concepts, $y = f(\mathbf{x})$

**Theorem:** For any learner $L$, $\frac{1}{|\mathcal{F}|}\sum_{\mathcal{F}} Acc_G(L) = \frac{1}{2}$
(given any distribution $\mathcal{D}$ over $\mathbf{x}$ and training set size $n$)

*Proof sketch*: Given any training set $S$:
For every concept $f$ where $Acc_G(L) = \frac{1}{2} + \delta$,
there is a concept $f'$ where $Acc_G(L) = \frac{1}{2} - \delta$.
$\forall \mathbf{x} \in S, f'(\mathbf{x}) = f(\mathbf{x}) = y.\quad \forall \mathbf{x} \notin S, f'(\mathbf{x}) = \neg f(\mathbf{x})$.

**Corollary:** For any two learners $L_1$, $L_2$:
**If** $\exists$ learning problem s.t. $Acc_G(L_1) > Acc_G(L_2)$
**Then** $\exists$ learning problem s.t. $Acc_G(L_2) > Acc_G(L_1)$

## What Does This Mean in Practice?

- Don't expect your favorite learner to always be best
- Try different approaches and compare
- But how could (say) a multilayer perceptron be less accurate than a single-layer one?

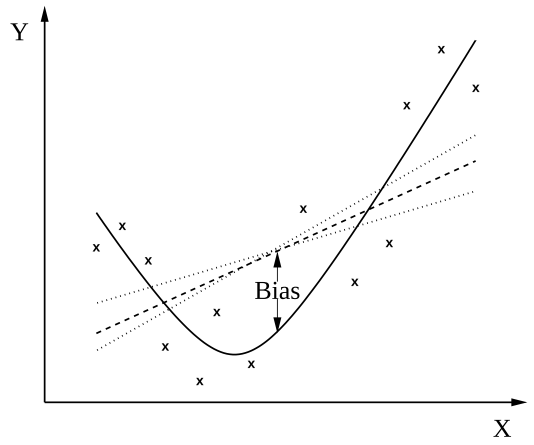
## Bias and Variance

- Bias-variance decomposition is key tool for understanding learning algorithms
- Helps explain why simple learners can outperform powerful ones
- Helps explain why model ensembles outperform single models
- Helps understand & avoid overfitting
- Standard decomposition for squared loss
- Can be generalized to zero-one loss

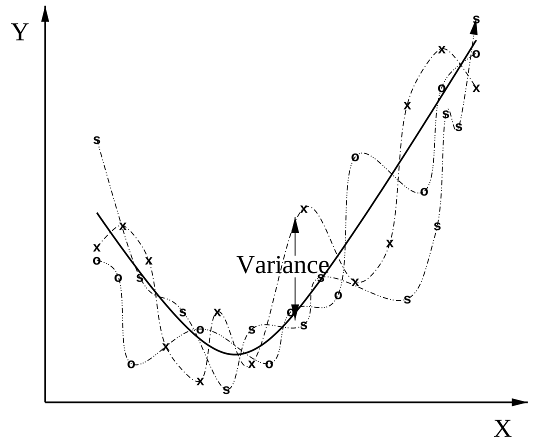
## Definitions

- Given training set: $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n)\}$
- Learner induces model: $y = f(\mathbf{x})$
- Loss measures quality of learner's predictions
  - Squared loss: $L(t, y) = (t - y)^2$
  - Absolute loss: $L(t, y) = |t - y|$
  - Zero-one loss: $L(t, y) = 0$ if $y = t$, 1 otherwise
  - Etc.
- Loss = Bias + Variance + Noise
  (This lecture: ignore noise; see paper)

## Bias




## Variance




## Decomposition for squared loss

$$\begin{aligned}(t-y)^2 &= (t-\overline{y}+\overline{y}-y)^2\\ &= (t-\overline{y})^2+(\overline{y}-y)^2+2(t-\overline{y})(\overline{y}-y)\end{aligned}$$

$$\begin{aligned}E[(t-y)^2] &= (t-\overline{y})^2+E[(\overline{y}-y)^2]\\ \text{Exp. loss} &= \text{Bias} \quad + \text{Variance}\end{aligned}$$

(Expectations are over training sets)

## How to generalize this to other loss funcs?

$$E[(t-y)^2] = (t-\overline{y})^2 + E[(\overline{y}-y)^2]$$

$$\begin{array}{rcll} (a-b)^2 & \rightarrow & L(a,b) & \\ E[(t-y)^2] & \rightarrow & E[L(t,y)] & \text{(Exp. loss)} \\ (t-\overline{y})^2 & \rightarrow & L(t,\overline{y}) & \text{(Bias)} \\ E[(\overline{y}-y)^2] & \rightarrow & E[L(\overline{y},y)] & \text{(Variance)} \end{array}$$

## But what should $\overline{y}$ be?

Define **Main Prediction**:
Prediction with min average loss relative to all predictions

$$\overline{y}_L = \underset{y'}{\operatorname{argmin}}\, E[L(y,y')]$$

- Squared loss: $\overline{y} = \text{Mean}$
- Absolute loss: $\overline{y} = \text{Median}$
- Zero-one loss: $\overline{y} = \text{Mode}$

## Generalized definitions

**Bias** = Loss incurred by main prediction = $L(t,\overline{y})$

**Variance** = Average loss incurred by prediction relative to main prediction = $E[L(\overline{y},y)]$

These definitions have all the required properties.

**For zero-one loss:**

$$\textbf{Bias} = \begin{cases} 0 \text{ if main prediction is correct} \\ 1 \text{ otherwise} \end{cases}$$

**Variance** = Prob(Prediction $\neq$ Main pred) = $P(y \neq \overline{y})$

## Can we decompose zero-one loss into these?

Assume two-class problem.

**Bias = 0** $\Rightarrow$ **Loss = Bias + Variance**

Loss $= P(y \neq t)$ Variance $= P(y \neq \overline{y})$

Bias $= 0 \Leftrightarrow \overline{y} = \mathrm{t}$

**Bias = 1** $\Rightarrow$ **Loss = Bias − Variance**

Loss $= P(y \neq t) = 1 - P(y = t) = 1 - P(y \neq \overline{y})$

because if $\overline{y} \neq t$ then $y = t \Leftrightarrow y \neq \overline{y}$.

Increasing variance can reduce loss!

## Can we generalize this further?

$$\boxed{\text{Loss} = \text{Bias} + c \text{ Variance}}$$

where c = 1 if Bias = 0, otherwise see below

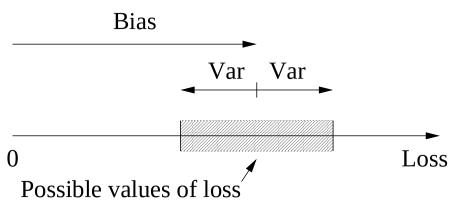
- **Applies to:**
  - Squared loss: $c = 1$
  - Two-class problems: $c = -1$
  - Multiclass problems: $c = -P(y = t | y \neq \overline{y})$
  - Variable costs: $c = -L(t, \overline{y})/L(\overline{y}, t)$

## Metric loss functions

- What about loss functions where decomposition does not apply?
- For any metric loss function:

  Loss $\leq$ Bias + Variance

  Loss $\geq$ Max {Bias − Var, Var − Bias}




## PAC Learning

- Overfitting happens because training error is bad estimate of generalization error

$\rightarrow$ Can we infer something about generalization error from training error?

- Overfitting happens when the learner doesn't see "enough" examples

$\rightarrow$ Can we estimate how many examples are enough?

## Problem Setting

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 sequence $D$ of training examples $\langle x, c(x) \rangle$, for some target concept $c \in C$

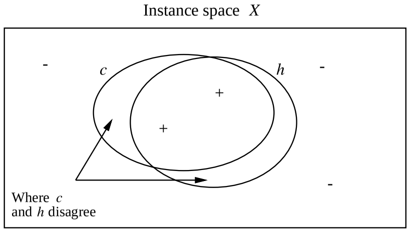
- Instances $x$ are drawn from distribution $\mathcal{D}$
- Teacher provides target value $c(x)$ for each

Learner must output a hypothesis $h$ estimating $c$

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

Note: probabilistic instances, noise-free classifications

## 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 $\mathcal{D}$.

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

## Two Notions of Error

**Training error** of hypothesis $h$ with respect to target concept $c$

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

**True error** of hypothesis $h$ with respect to $c$

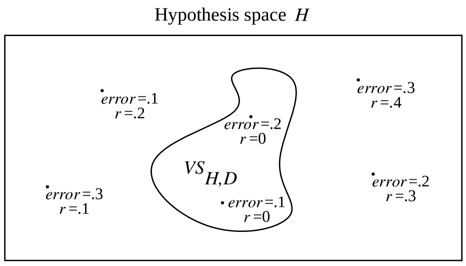
- How often $h(x) \neq c(x)$ over future random instances

**Our concern:**

- Can we bound the true error of $h$ given the training error of $h$?
- First consider when training error of $h$ is zero

## Version Spaces

**Version Space $VS_{H,D}$:**
Subset of hypotheses in $H$ consistent with training data $D$




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

## How Many Examples Are Enough?

**Theorem:**
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 $VS_{H,D}$ contains a hypothesis with error greater than $\epsilon$ is less than

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

*Proof sketch:*
Prob(1 hyp. w/ error $> \epsilon$ consistent w/ 1 ex.) $< 1 - \epsilon \leq e^{-\epsilon}$
Prob(1 hyp. w/ error $> \epsilon$ consistent with $m$ exs.) $< e^{-\epsilon m}$
Prob(1 of $|H|$ hyps. consistent with $m$ exs.) $< |H|e^{-\epsilon m}$

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

If we want this probability to be at most $\delta$

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

then

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

## Learning Conjunctions

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

Use our theorem:

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

Suppose $H$ contains conjunctions of constraints on up to $n$ Boolean attributes (i.e., $n$ literals). Then $|H| = 3^n$, and

$$\begin{aligned} m &\geq \frac{1}{\epsilon}(\ln 3^n + \ln(1/\delta)) \\ &\geq \frac{1}{\epsilon}(n \ln 3 + \ln(1/\delta)) \end{aligned}$$

## How About $PlayTennis$?

1 attribute with 3 values (outlook)
9 attributes with 2 values (temp, humidity, wind, etc.)
Language: Conjunction of features or null concept

$$|H| = 4 \times 3^9 + 1 = 78733$$

$$m \geq \frac{1}{\epsilon}(\ln 78733 + \ln(1/\delta))$$

If we want to ensure that with probability 95%,
$VS$ contains only hypotheses with $error_{\mathcal{D}}(h) \leq 10\%$,
then it is sufficient to have $m$ examples, where

$$m \geq \frac{1}{0.1}(\ln 78733 + \ln(1/.05)) = 143$$

(# examples in domain: $3 \times 2^9 = 1536$)

## 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 $H$ iff for all $c \in C$, distributions $\mathcal{D}$ over $X$, $\epsilon$ such that $0 < \epsilon < 1/2$, and $\delta$ 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 $size(c)$.

## Agnostic Learning

So far, assumed $c \in H$

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

- What can we say in this case?
  - 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 the sample complexity in this case?

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

## VC Dimension

- What about hypotheses with numeric parameters?
- Solution: Use VC dimension instead of $\ln |H|$

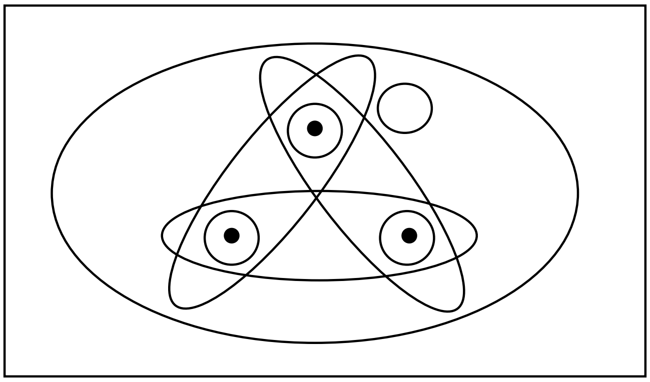
## Shattering a Set of Instances

*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.

## Three Instances Shattered

Instance space $X$



## 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 Dim. of Linear Decision Surfaces

(a) (b)

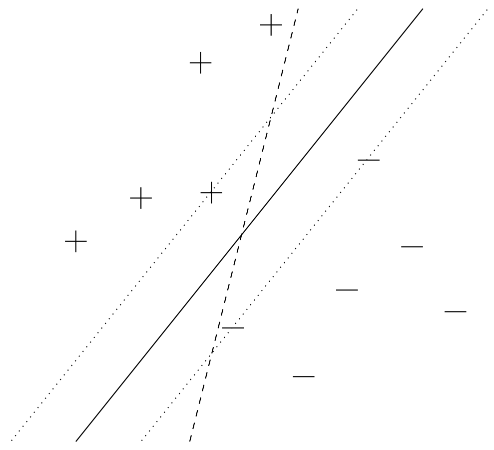
VC dim. of hyperplane in $d$-dimensional space is $d+1$

## Sample Complexity from VC Dimension

How many randomly drawn examples suffice to guarantee error of at most $\epsilon$ with probability at least $(1-\delta)$?

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

## Support Vector Machines



## Support Vector Machines

- Many different hyperplanes can separate positive and negative examples
- Choose hyperplane with maximum margin
- **Margin**: Min. distance between plane and example
- Bound on VC dimension decreases with margin
- **Support vectors:** Examples that determine the plane
- $E[error_{\mathcal{D}}(h)] \leq \frac{E[\#support\ vectors]}{\#training\ vectors - 1}$
- Noisy data: use slack variables
- Avoids overfitting even in very high-dimensional spaces (e.g., text)
- Non-linear: augment data with derived features

## Learning Theory: Summary

- "No free lunch" theorems
- Bias and variance
- PAC learning
- VC dimension
- Support vector machines