Machine Learning 10-601

Based on slide from Tom M. Mitchell

Computational Learning Theory

Computational Learning Theory

- What general laws constrain inductive learning?
- Want theory to relate
  - Number of training examples
  - Complexity of hypothesis space
  - Accuracy to which target function is approximated
  - Manner in which training examples are presented
  - Probability of successful learning

Sample Complexity

How many training examples suffice to learn target concept

1. If learner proposes instances as queries to teacher?
   - learner proposes x, teacher provides f(x)

2. If teacher (who knows f(x)) proposes training examples?
   - teacher proposes sequence $\{<x^1, f(x^1)>, \ldots <x^n, f(x^n)>$

3. If some random process (e.g., nature) proposes instances, and teacher labels them?
   - instances drawn according to *P(X)*

Sample Complexity (cont.)

The true (but unknown) function that labels objects

Problem setting:

- Set of instances X
- Set of hypotheses $H = \{h : X \to \{0,1\}\}$
- Set of possible target functions $C = \{c : X \to \{0,1\}\}$
- Sequence of training instances drawn at random from $P(X)$
- Teacher provides noise-free label $c(x)$

Learner outputs a hypothesis $h \in H$ such that

$$h = \arg\min_{h \in H} \; error_{train}(h)$$

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 D

$$error_{train} \equiv \Pr_{x \in D}[hx \neq c(x)] = \sum_{x \in D} \frac{\delta(h(x) \neq c(x))}{|D|}$$

training examples D

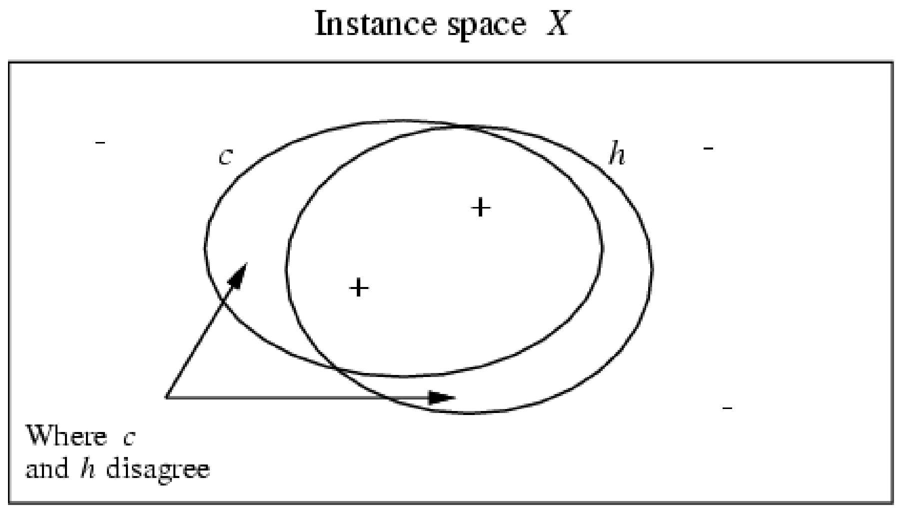
True error of hypothesis h with respect to c

- How often $h(x) \neq c(x)$ over future instances drawn at random from $\mathcal{D}$

$$error_{true}(h) \equiv \Pr_{x \sim P(X)}[h(x) \neq c(x)]$$

Probability distribution P(X)

True Error of a Hypothesis



The *true error* of h is the probability that it will misclassify an example drawn at random from $P(X)$

$$error_{true}(h) \equiv \Pr_{x \sim P(X)}[h(x) \neq c(x)]$$

Overfitting

Consider a hypothesis *h* and its

- Error rate over training data: $error_{train}(h)$
- True error rate over all data: $error_{true}(h)$

easy to compute

unknown

We say *h* overfits the training data if

$$error_{true}(h) > error_{train}(h)$$

Amount of overfitting =

$$error_{true}(h) - error_{train}(h)$$

Can we bound $error_{true}(h)$

in terms of $error_{train}(h)$??

$$error_{train} \equiv \Pr_{x \in D}[h(x) \neq c(x)] = \sum_{x \in D} \frac{\delta(h(x) \neq c(x))}{|D|}$$

training examples

$$error_{true}(h) \equiv \Pr_{x \sim P(X)}[h(x) \neq c(x)]$$

Probability distribution P(x)

if D was a set of examples drawn from $P(X)$ and ***independent*** of h, then we could use standard statistical confidence intervals to determine that with 95% probability $error_{true}(h)$ lies in the interval:

$$error_{D}(h) \pm 1.96 \sqrt{\frac{error_{D}(h)\,(1 - error_{D}(h)\,)}{n}}$$

but D is the ***training data*** for h ….

Version Spaces

c: X → {0,1}

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

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

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

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

Example: H is Conjunction of up to N Boolean Literals

Consider classification problem f:X→Y:

- instances: $X = <X_1\ X_2\ X_3\ X_4>$ where each X_i is boolean
- Each hypothesis in H is a rule of the form:
  - IF $<X_1\ X_2\ X_3\ X_4> = <0,?,1,?>$, THEN Y=1, ELSE Y=0
  - i.e., rules constrain any subset of the X_i

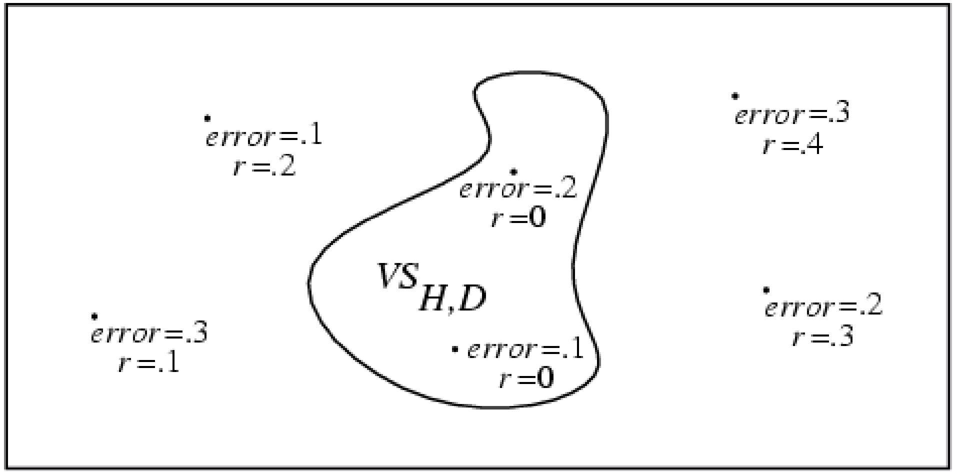
Example 2: H is Decision Tree with depth=2

Consider classification problem f:X→Y:

- instances: $X = \langle X_1 \ldots X_N \rangle$ where each X_i is boolean
- learned hypotheses are decision trees of depth 2 (so each can use only three variables).

Exhausting the Version Space

Hypothesis space H



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

Definition: The version space $VS_{H,D}$ with respect to training data D is said to be **ϵ-exhausted** if every hypothesis h in $VS_{H,D}$ has true error less than ϵ.

$$(\forall h \in VS_{H,D})\ error_{true}(h) < \epsilon$$

How many examples will ϵ-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

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

How many examples will ϵ-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 $error(h) \geq \epsilon$

Any(!) learner that outputs a hypothesis consistent with all training examples (i.e., an h contained in $VS_{H,D}$)

What it means

[Haussler, 1988]: probability that the version space is not ε-exhausted after *m* training examples is at most $|H|e^{-\epsilon m}$

$$\Pr[(\exists h \in H) s.t. (error_{train}(h) = 0) \wedge (error_{true}(h) > \epsilon)] \ \leq \ |H|e^{-\epsilon m}$$

↑

Suppose we want this probability to be at most δ

1. How many training examples suffice?

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

2. If $error_{train}(h) = 0$ then with probability at least (1-δ):

$$error_{true}(h) \leq \frac{1}{m}(\ln|H| + \ln(1/\delta))$$

Example: Simple decision trees

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

Consider Boolean classification problem

- instances: X = <X_1 ... X_N> where each X_i is Boolean
- Each hypothesis in H is a decision tree of depth 1

How many training examples *m* suffice to assure that with probability at least 0.99, *any* consistent learner using H will output a hypothesis with true error at most 0.05?

Example: Simple decision trees

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

Consider Boolean classification problem

- instances: $X = \langle X_1 \ldots X_n \rangle$ where each X_i is boolean
- Each hypothesis in H is a decision tree of depth 1

How many training examples *m* suffice to assure that with probability at least 0.99, *any* consistent learner using H will output a hypothesis with true error at most 0.05?

|H| = 2n,   epsilon = 0.05, delta = 0.01

Example: H is Conjunction of Boolean Literals

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

Consider classification problem f:X→Y:

- instances: $X = \langle X_1\ X_2\ X_3\ X_4 \rangle$ where each X_i is boolean
- Each hypothesis is a rule of the form:
  - IF $\langle X_1\ X_2\ X_3\ X_4 \rangle = \langle 0,?,1,? \rangle$, THEN Y=1, ELSE Y=0
  - i.e., rules constrain any subset of the X_i

How many training examples *m* suffice to assure that with probability at least 0.99, *any* consistent learner will output a hypothesis with true error at most 0.05?

Example: H is Decision Tree with depth=2

Consider classification problem f:X→Y:

- instances: $X = \langle X_1 \ldots X_N \rangle$ where each X_i is boolean
- learned hypotheses are decision trees of depth 2, using only two variables

How many training examples *m* suffice to assure that with probability at least 0.99, *any* consistent learner will output a hypothesis with true error at most 0.05?

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 $size(c)$.

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 <u>time that is polynomial</u> in $1/\epsilon$, $1/\delta$, n and $size(c)$.

Sufficient condition:

Holds if learner L requires only a polynomial number of training examples, and processing per example is polynomial

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

Question: If H = {h | h: X → Y} is infinite, what measure of complexity should we use in place of |H| ?

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

Question: If H = {h | h: X → Y} is infinite, what measure of complexity should we use in place of |H| ?

Answer: The largest subset of X for which H can <u>guarantee</u> zero training error (regardless of the target function c)

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

Question: If H = {h | h: X → Y} is infinite, what measure of complexity should we use in place of |H| ?

Answer: The largest subset of X for which H can guarantee zero training error (regardless of the target function c)

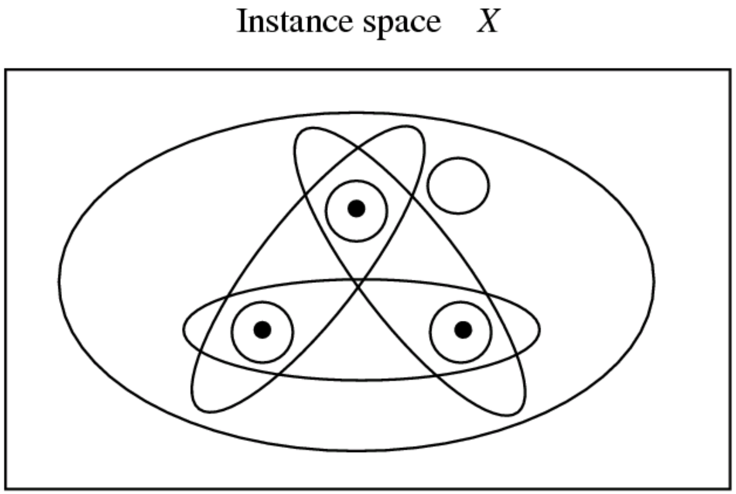
VC dimension of H is the size of this subset

Shattering a Set of Instances

a labeling of each member of S as positive or negative

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.



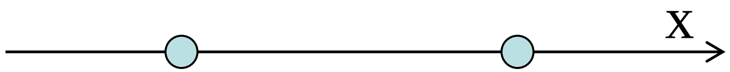
The Vapnik-Chervonenkis Dimension

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: examples

Consider X = <, want to learn c:X→{0,1}

What is VC dimension of



- Open intervals:

H1: if $x > a$ then $y = 1$ else $y = 0$

H2: if $x > a$ then $y = 1$ else $y = 0$
or, if $x > a$ then $y = 0$ else $y = 1$

- Closed intervals:

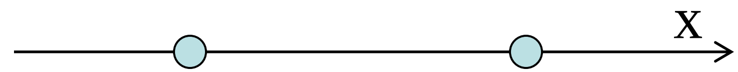
H3: if $a < x < b$ then $y = 1$ else $y = 0$

H4: if $a < x < b$ then $y = 1$ else $y = 0$
or, if $a < x < b$ then $y = 0$ else $y = 1$

VC dimension: examples

Consider X = <, want to learn c:X→{0,1}

What is VC dimension of



- Open intervals:

H1: if $x > a$ then $y = 1$ else $y = 0$ VC(H1)=1

H2: if $x > a$ then $y = 1$ else $y = 0$
or, if $x > a$ then $y = 0$ else $y = 1$ VC(H2)=2

- Closed intervals:

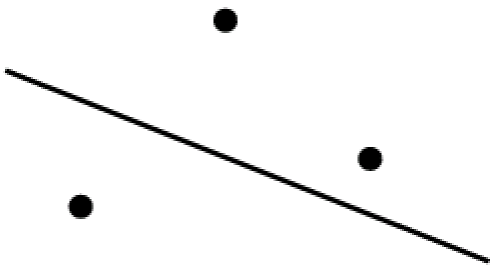
H3: if $a < x < b$ then $y = 1$ else $y = 0$ VC(H3)=2

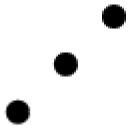
H4: if $a < x < b$ then $y = 1$ else $y = 0$
or, if $a < x < b$ then $y = 0$ else $y = 1$ VC(H4)=3

VC dimension: examples

What is VC dimension of lines in a plane?

- $H_2 = \{ ((w_0 + w_1x_1 + w_2x_2)>0 \rightarrow y=1) \}$





VC dimension: examples

What is VC dimension of

- $H_2 = \{ ((w_0 + w_1 x_1 + w_2 x_2) > 0 \rightarrow y=1) \}$
  - $VC(H_2)=3$
- For H_n = linear separating hyperplanes in n dimensions, $VC(H_n)=n+1$

Sample Complexity based on VC dimension

How many randomly drawn examples suffice to ε-exhaust $VS_{H,D}$ with probability at least $(1-\delta)$?

ie., to guarantee that any hypothesis that perfectly fits the training data is probably $(1-\delta)$ approximately (ε) correct

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

Compare to our earlier results based on $|H|$:

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

For any finite hypothesis space H, can you give an upper bound on VC(H) in terms of |H| ? (hint: yes)

More VC Dimension Examples to Think About

- Logistic regression over n continuous features
  - Over n boolean features?

- Linear SVM over n continuous features

- Decision trees defined over n boolean features

  F: $< X_1, \dots X_n > \rightarrow Y$

- Decision trees of depth 2 defined over n features

- How about 1-nearest neighbor?