Concept Learning

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Courtesy: Prof. Pabitra Mitra, CSE, IIT Kharagpur

References

- 1. Chapter 2 and 7 of "Machine learning" by Tom M. Mitchel.
- 2. Chapter 2 of "Introduction to Machine Learning" by Ethem Alpaydin.

Which days does one come out to enjoy sports?

- Sky condition
  - Rainy / Cloudy / Sunny
- Humidity
  - High / Normal
- Temperature
  - Warm / Cold
- Wind
  - Strong / Weak
- Water
  - Warm / Cool
- Forecast
  - Same / Change

Attributes of a day: takes on values

Enjoy sports (?): Yes / No

Learning Task

- To make a hypothesis about the day on which a person comes out to enjoy sports.
  - in the form of a boolean function on the the attributes of the day.

- Find the right hypothesis/function from historical data
  - Training Examples (TE)

Training Examples for EnjoySport

	Sky	Temp	Humid	Wind	Water	Forecst		EnjoySpt
c(	Sunny	Warm	Normal	Strong	Warm	Same	)=1	Yes
c(	Sunny	Warm	High	Strong	Warm	Same	)=1	Yes
c(	Rainy	Cold	High	Strong	Warm	Change	)=0	No
c(	Sunny	Warm	High	Strong	Cool	Change	)=1	Yes

c is the target concept

- Negative and positive learning examples.
- To learn the target concept c.
  - A Boolean function

Concept learning

- To derive a Boolean function from training examples.
  - Many “hypothetical” Boolean functions
    - find h such that h = c.
- Generate hypotheses for concept from TE’s

Representing a Hypothesis

- A hypothesis as conjunction of constraints.
  - Each constraint: a Boolean condition on attribute values.
  - Three forms
    - Specific value : Water = *Warm*
    - Don't-care value: Water = ?
      - Any value satisfies condition.
    - No value allowed : Water $= \emptyset$
      - i.e., no permissible value given values of other attributes
  - Represented in the form of a vector.

Example of a hypothesis

- Represented in the form of a vector:
  - <sky, temp, humid, wind, water, forecast>
  - h=<Sunny ? ? Strong ? Same>

$$h(x) = 1 \quad \text{if } h \text{ is true on } x$$
$$= 0 \quad \text{otherwise}$$

- x is also represented as a vector, an element in the 6-D space.
  - x= <Sunny, Warm, Normal, Strong, Warm, Same>
    - $h(x)$=1
  - x= <Sunny, Warm, Normal, Strong, Warm, Change>
    - $h(x)$=0

Space of Hypotheses

- H: A set of all possible hypotheses
  - Finite number of combinations.
- Size of input space X
  - X = Sky x Temp x Humid x Wind x Water x Forecast
  - |X|= 3 x 2 x 2 x 2 x 2 x 2 = 96
- Size of H
  - Each attribute A can have |A|+|{∅,?}| conditions.
    - E.g. for Sky: 3+2=5
  - |H|= 5 x 4 x 4 x 4 x 4 x 4= 5120
    - But every h with ∅ : empty set of instances (all negatives)
  - No. of distinct hypotheses: 1+4x3x3x3x3x3=973

Concept Learning: Task

TASK T: predicting when a person will enjoy sport

- **Target function** c: EnjoySport : $X \rightarrow \{0, 1\}$
- Cannot, in general, know Target function c
  - Adopt hypotheses H about c
- Form of hypotheses H :
  - Conjunctions of literals
    - $\langle ?, \text{Cold}, \text{High}, ?, ?, ? \rangle$

Concept Learning: Experience

- **EXPERIENCE E**
  - **Instances** X: possible days described by attributes *Sky*, *Temp*, *Humidity*, *Wind*, *Water*, *Forecast*
  - **Training examples** D: Positive / negative examples of target function $\{\langle x_1, c(x_1)\rangle, \ldots \langle x_m, c(x_m)\rangle\}$

Concept Learning: Performance Measure

PERFORMANCE MEASURE P:

The Hypothesis h in H such that $h(x) = c(x)$ for all x in D (Training Examples).

- may exist several alternative hypotheses that fit examples.
- Assumption of inductive learning on h being true for unseen examples.

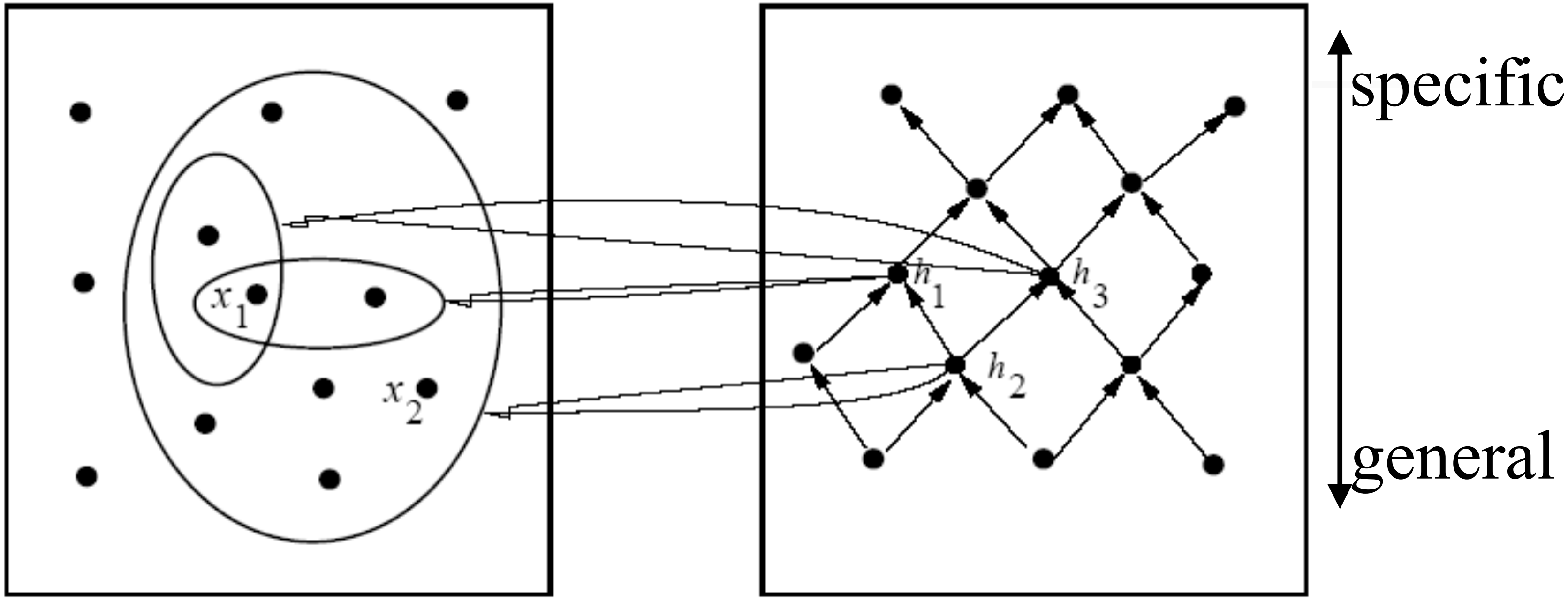
Inductive Learning Hypothesis

- Any hypothesis h found to approximate the target function c well over a sufficiently large set of training examples D will also approximate the target function well over other unobserved examples (i.e. in population distribution $\mathcal{D}$) .

$$\forall x \in D,\ h(x) \approx c(x) \rightarrow \forall x \in \mathcal{D},\ h(x) \approx c(x)$$

Ordering on Hypotheses

Instances X Hypotheses H



x_1=⟨*Sunny Warm High Strong Cool Same*⟩

x_2=⟨*Sunny Warm High Light Warm Same*⟩

h_1=⟨*Sunny ? ? Strong ? ?*⟩

h_2=⟨*Sunny ? ? ? ? ?*⟩

h_3=⟨*Sunny ? ? ? Cool ?*⟩

- h is more general than h' ($h \geq_g h'$) if for each instance x,

$$h'(x) = 1 \rightarrow h(x) = 1$$

- Which is the most general/most specific hypothesis?

Learning as a search problem

- Search a hypothesis h in the space H to best fit examples.
- If examples are error free, h should satisfy all of them
  - Not unique.
  - several alternative hypotheses may fit examples.
  - May not exist any solution at all!
    - Satisfying all +ve and –ve examples.
    - Constraints may have other form
      - e.g. Sky condition could be (rainy OR cloudy), but not admitted in H.

Approaches to learning algorithms

- Approach 1: Search based on ordering of hypotheses.
- Approach 2: Search based on finding all possible hypotheses using a good representation of hypothesis space.
  - All hypotheses that fit data

The choice of the hypothesis space reduces the number of hypotheses.

Assumes

- There is a hypothesis h in describing target function c.
- There are no errors in the TE's.

Find-S Algorithm

1. Initialize *h* to the most specific hypothesis in *H* *(what is this?)*
2. For each *positive* training instance *x*

   For each attribute constraint a_i in *h*

   If the constraint a_i in *h* is satisfied by *x*

   do nothing

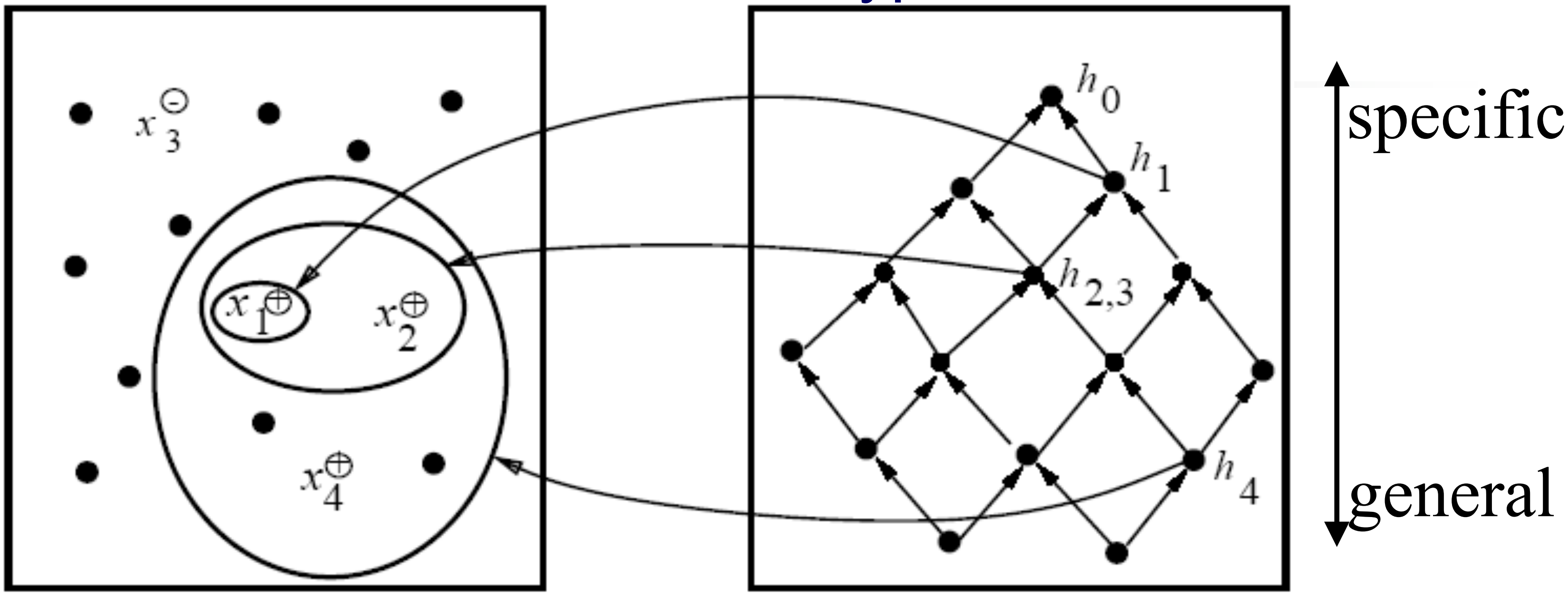
   Else

   replace a_i in *h* by the next more general constraint that is satisfied by *x*
3. Output hypothesis *h.*

As no change for a negative example, it is ignored.

Example of Find-S

Instances X Hypotheses H



x_1=⟨*Sunny Warm Normal Strong Warm Same*⟩ +

x_2=⟨*Sunny Warm High Strong Warm Same*⟩ +

x_3=⟨*Rainy Cold High Strong Warm Change*⟩ –

x_4=⟨*Sunny Warm High Strong Cool Change*⟩ +

h_0=⟨∅ ∅ ∅ ∅ ∅ ∅⟩

h_1=⟨*Sunny Warm Normal Strong Warm Same*⟩

h_2=⟨*Sunny Warm ? Strong Warm Same*⟩

h_3=⟨*Sunny Warm ? Strong Warm Same*⟩

h_4=⟨*Sunny Warm ? Strong ? ?*⟩

Problems with Find-S

- Problems:
  - Throws away information!
    - Negative examples
  - Can't tell whether it has learned the concept
    - Depending on *H*, there might be several h's that fit TEs!
    - Picks a maximally specific *h.* (why?)
  - Can't tell when training data is inconsistent
    - Since ignores negative TEs

- Advantages
  - Simple
  - Outcome independent of order of examples
    - Why?

- Any alternative?
  - Keep *all* consistent hypotheses!
    - Candidate elimination algorithm

Consistent Hypothesis

- if $h(x) = c(x)$ for each training example $\langle x, c(x) \rangle$ in *D.*
  - consistent with a set of training examples *D* of target concept *c*
  - Note that consistency is with respect to specific *D*.
- Notation:

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

Agnostic hypothesis:

May label erroneously a training sample.

$$Agnostic(h, D) \equiv \exists \langle x, c(x) \rangle \in D :: h(x) \neq c(x)$$

Version Space

- $VS_{H,D}$:The subset of hypotheses from H consistent with D
  - with respect to hypothesis space H and training examples D
- Notation:
  $VS_{H,D} = \{h \mid h \in H \wedge Consistent(h, D)\}$

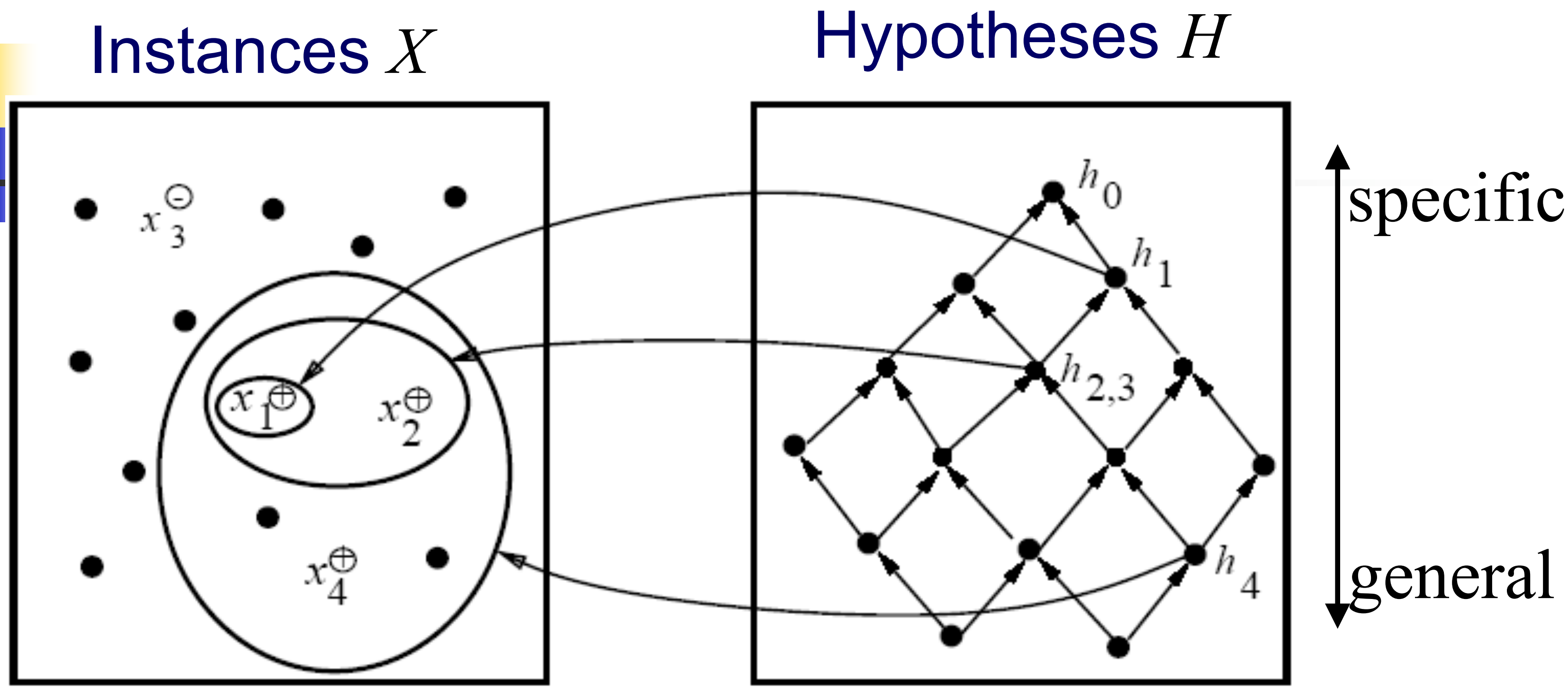
List-Then-Eliminate Algorithm

1. *VersionSpace* $\leftarrow$ list of all hypotheses in *H*
2. For each training example $\langle x, c(x) \rangle$

   remove from *VersionSpace* any hypothesis *h* for which $h(x) \neq c(x)$.
3. Output the list of hypotheses in *VersionSpace*.

Essentially a brute force procedure.

Example of Find-S, Revisited



$h_0=\langle\varnothing\ \varnothing\ \varnothing\ \varnothing\ \varnothing\ \varnothing\rangle$

$x_1=\langle Sunny\ Warm\ Normal\ Strong\ Warm\ Same\rangle$ +

$h_1=\langle Sunny\ Warm\ Normal\ Strong\ Warm\ Same\rangle$

$x_2=\langle Sunny\ Warm\ High\ Strong\ Warm\ Same\rangle$ +

$h_2=\langle Sunny\ Warm\ ?\ Strong\ Warm\ Same\rangle$

$\boxed{x_3=\langle Rainy\ Cold\ High\ Strong\ Warm\ Change\rangle -}$

$h_3=\langle Sunny\ Warm\ ?\ Strong\ Warm\ Same\rangle$

$x_4=\langle Sunny\ Warm\ High\ Strong\ Cool\ Change\rangle$ +

$\boxed{h_4=\langle Sunny\ Warm\ ?\ Strong\ ?\ ?\rangle}$

h_5: consistent? $h_5=\langle Sunny\ Warm\ ?\ ?\ ?\ ?\rangle$

h_4: Least general

Restriction on most general hypothesis looking at the –ve sample!

Version Space for this Example

Except +ve samples all -ve.

S $\{\langle$Sunny Warm ? Strong ? ?$\rangle\}$

$\langle$Sunny ? ? Strong ? ?$\rangle$ $\langle$Sunny Warm ? ? ? ?$\rangle$ $\langle$? Warm ? Strong ? ?$\rangle$

Except –ve samples all +ve.

G $\{\langle$Sunny ? ? ? ? ?$\rangle$, $\langle$? Warm ? ? ? ?$\rangle\}$

Compact Representation of the Version Space

- Store the most and the least general boundaries of space.
  - Generalize from most specific boundaries
    - Use +ve samples.
  - Specialize from most general boundaries
    - Use –ve samples.
- Generate all intermediate h's in VS
  - any h in VS must be consistent with all TE's

Compact Representation of the Version Space $VS_{H,D}$

- The general boundary, *G*,
  - the set of its maximally general members consistent with D
    - Summarizes the negative examples;
    - Anything more general covers wrongly a negative TE
- The specific boundary, *S*,
  - the set of its maximally specific members consistent with D
    - Summarizes the positive examples;
    - Anything more specific fails to cover a positive TE

Theorem

Every member of the version space lies between the S,G boundary

$$VS_{H,D} = \{h \mid h \in H \wedge \exists s \in S\ \exists g \in G\ (g \geq h \geq s)\}$$

- Must prove:
  - 1) every h satisfying RHS is in $VS_{H,D;}$
  - 2) every member of $VS_{H,D}$ satisfies RHS.

Proof

Every member of the version space lies between the S,G boundary

$$VS_{H,D} = \{h \mid h \in H \wedge \exists s \in S\ \exists g \in G\ (g \geq h \geq s)\}$$

- Must prove:
  - 1) every h satisfying RHS is in $VS_{H,D}$;
  - 2) every member of $VS_{H,D}$ satisfies RHS.

- For 1), let g, h, s be arbitrary members of G, H, S respectively with g>h>s Prove that h is consistent.
  - s must be satisfied by all + TEs and so must h because it is more general;
  - g cannot be satisfied by any – TEs, and so nor can h
  - h is in $VS_{H,D}$ since satisfied by all + TEs and no – TEs
- For 2),
  - Since h satisfies all + TEs and no – TEs, h $\geq$ s, and $g \geq h$.

Candidate Elimination Algorithm

G ← maximally general hypotheses in *H*
S ← maximally specific hypotheses in *H*

For each training example *d*, do

- If *d* is positive
  - Remove from *G* every hypothesis inconsistent with *d*
  - For each hypothesis *s* in *S* that is inconsistent with *d*
    - Remove *s* from *S*
    - Add to *S* all minimal generalizations *h* of *s* such that
      1. *h* is consistent with *d*, and
      2. some member of *G* is more general than *h*
  - Remove from *S* every hypothesis that is more general than another hypothesis in *S*

Candidate Elimination Algorithm (cont)

- If *d* is a negative example
  - Remove from *S* every hypothesis inconsistent with *d*
  - For each hypothesis *g* in *G* that is inconsistent with *d*
    - Remove *g* from *G*
    - Add to *G* all minimal specializations *h* of *g* such that
      1. *h* is consistent with *d*, and
      2. some member of *S* is more specific than *h*
  - Remove from *G* every hypothesis that is less general than another hypothesis in *G*

- Essentially use
  - Pos TEs to generalize S
  - Neg TEs to specialize G
- Independent of order of TEs

- Convergence guaranteed if:
  - ***no errors***
  - ***there is h in H describing c***.

Example

Recall : If d is positive

Remove from *G* every hypothesis inconsistent with *d*

For each hypothesis s in S that is inconsistent with d

- Remove s from S
- Add to S all minimal generalizations h of s that are specializations of a hypothesis in G
- Remove from S every hypothesis that is more general than another hypothesis in S

S_0 $\boxed{\{\langle \varnothing\ \varnothing\ \varnothing\ \varnothing\ \varnothing\ \varnothing \rangle\}}$

G_0 $\boxed{\{\langle ?\ ?\ ?\ ?\ ?\ ? \rangle\}}$

$\langle \textit{Sunny Warm Normal Strong Warm Same} \rangle$ +

S_1 $\boxed{\{\langle \text{Sunny Warm Normal Strong Warm Same} \rangle\}}$

G_1 $\boxed{\{\langle ?\ ?\ ?\ ?\ ?\ ? \rangle\}}$

Example (contd)

S_1 $\boxed{\{\langle \text{Sunny Warm Normal Strong Warm Same} \rangle\}}$

G_1 $\boxed{\{\langle ?\ ?\ ?\ ?\ ?\ ? \rangle\}}$

$\langle \textit{Sunny Warm High Strong Warm Same} \rangle$ +

S_2 $\boxed{\{\langle \text{Sunny Warm ? Strong Warm Same} \rangle\}}$

G_2 $\boxed{\{\langle ?\ ?\ ?\ ?\ ?\ ? \rangle\}}$

If d is a negative example

- Remove from S every hypothesis inconsistent with d
- For each hypothesis g in G that is inconsistent with d
  - Remove g from G
  - Add to G all minimal specializations h of g that generalize some hypothesis in S
  - Remove from G every hypothesis that is less general than another hypothesis in G

Example (contd)

S_2 $\boxed{\{\langle \text{Sunny Warm ? Strong Warm Same} \rangle\}}$

G_2 $\boxed{\{\langle \text{? ? ? ? ? ?} \rangle\}}$

$\langle \textit{Rainy Cold High Strong Warm Change} \rangle$ –

Current G boundary is incorrect
So, need to make it more specific.

S_3 $\boxed{\{\langle \text{Sunny Warm ? Strong Warm Same} \rangle\}}$

G_3 $\boxed{\{\langle \text{Sunny ? ? ? ? ?} \rangle, \langle \text{? Warm ? ? ? ?} \rangle, \langle \text{? ? ? ? ? Same} \rangle\}}$

Example (contd)

- Why are there no hypotheses left relating to:
  - ⟨ *Cloudy ? ? ? ? ?*⟩
    - Inconsistent with S.
- The following specialization using the third value

  ⟨*? ? Normal ? ? ?*⟩,

  is not more general than the specific boundary

  {⟨Sunny Warm ? Strong Warm Same⟩}

- The specializations ⟨*? ? ? Weak ? ?*⟩, ⟨*? ? ? ? Cool ?*⟩ are also inconsistent with S

Example (contd)

S_3 $\boxed{\{\langle \text{Sunny Warm ? Strong Warm Same} \rangle\}}$

G_3 $\boxed{\{\langle \text{Sunny ? ? ? ? ?} \rangle, \langle \text{? Warm ? ? ? ?} \rangle, \langle \text{? ? ? ? ? Same} \rangle\}}$

$\langle \textit{Sunny Warm High Strong Cool Change} \rangle$ +

$S4$ $\boxed{\{\langle \text{Sunny Warm ? Strong ? ?} \rangle\}}$

$G4$ $\boxed{\{\langle \text{Sunny ? ? ? ? ?} \rangle, \langle \text{? Warm ? ? ? ?} \rangle\}}$

Example (contd)

$\langle$*Sunny Warm High Strong Cool Change*$\rangle$ +

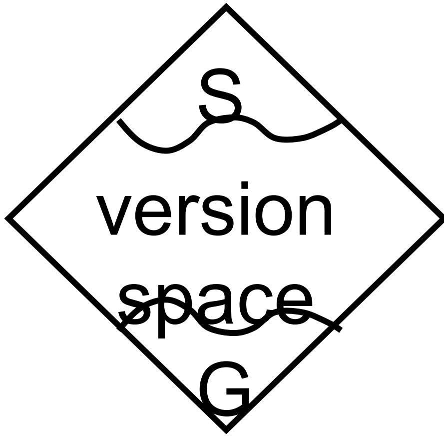
- Why does this example remove a hypothesis from G?:
  - $\langle$*? ? ? ? ? Same*$\rangle$
- This hypothesis
  - Cannot be specialized, since would not cover new TE.
  - Cannot be generalized, because more general would cover negative TE.
  - Hence must drop hypothesis.

Version Space of the Example

S {⟨Sunny Warm ? Strong ? ?⟩}

⟨Sunny ? ? Strong ? ?⟩ ⟨Sunny Warm ? ? ? ?⟩ ⟨? Warm ? Strong ? ?⟩

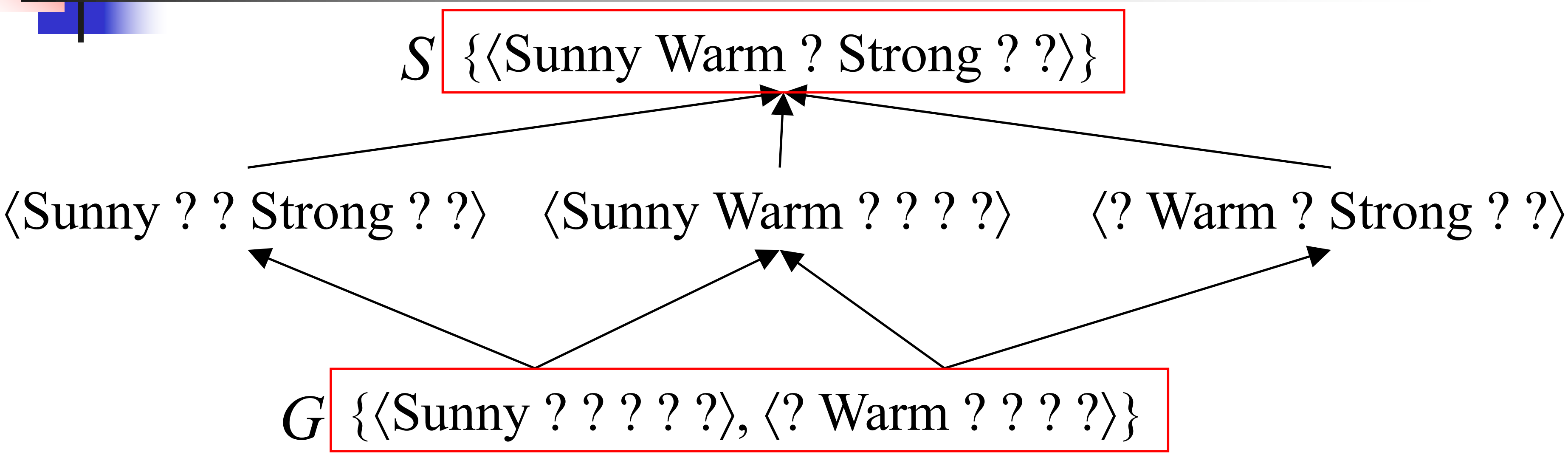
G {⟨Sunny ? ? ? ? ?⟩, ⟨? Warm ? ? ? ?⟩}



Convergence of algorithm

- Convergence guaranteed if:
  - ***no errors***
  - ***there is h in H describing c*.**
- Ambiguity removed from VS when S = G
  - Containing single h
  - When have seen enough TEs
- For any false negative TE, algorithm will remove every h consistent with TE, and hence may remove correct target concept from VS
  - If observed enough, TEs will find that S, G boundaries converge to empty VS

Which Next Training Example?



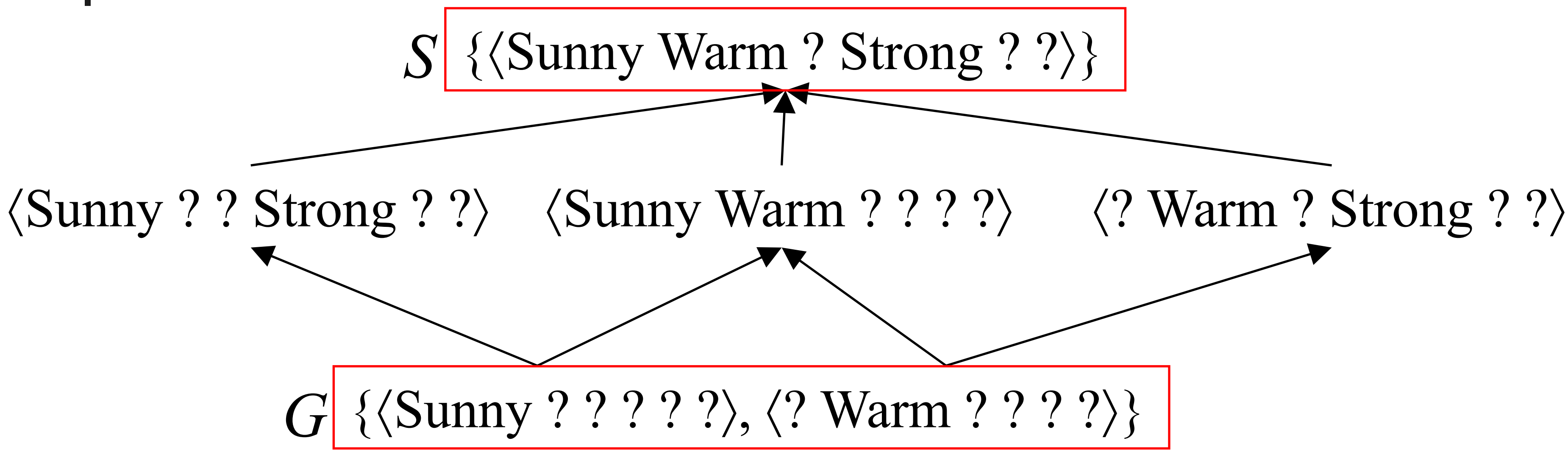
Assume learner can choose the next TE

- Should choose d such that
  - Reduces maximally the number of hypotheses in VS
  - Best TE: satisfies precisely 50% hypotheses;
    - Can't always be done

Order of examples matters for intermediate sizes of S,G; not for the final S, G

- Example:
  - ⟨Sunny Warm Normal Weak Warm Same⟩ ?
  - If pos, generalizes S
  - If neg, specializes G

Classifying new cases using VS



- Use *voting procedure* on following examples:
  - ⟨Sunny Warm Normal Strong Cool Change⟩
  - ⟨Rainy Cool Normal Weak Warm Same⟩
  - ⟨Sunny Warm Normal Weak Warm Same⟩
  - ⟨Sunny Cold Normal Strong Warm Same ⟩

Effect of incomplete hypothesis space

- Preceding algorithms work if target function is in H
  - Will generally not work if target function *not* in H
- Consider following examples which represent target function

  "sky = sunny or sky = cloudy":

  - ⟨Sunny Warm Normal Strong Cool Change⟩ Y
  - ⟨Cloudy Warm Normal Strong Cool Change⟩ Y
  - ⟨Rainy Warm Normal Strong Cool Change⟩ N

Effect of incomplete hypothesis space

"sky = sunny or sky = cloudy":

⟨Sunny Warm Normal Strong Cool Change⟩ Y
⟨Cloudy Warm Normal Strong Cool Change⟩ Y
⟨Rainy Warm Normal Strong Cool Change⟩ N

- If apply CE algorithm as before, end up with empty VS
  - After first two TEs,
    - S= ⟨? Warm Normal Strong Cool Change⟩
  - New hypothesis is overly general
    - it covers the third negative TE!
- Our H does not include the appropriate c.

Need more expressive hypotheses

Unbiased Learners

- if no limits on representation of hypotheses (i.e., full logical representation: *and, or, not*), can only learn examples…no generalization possible!
  - Say, 5 TEs {x1, x2, x3, x4, x5}, with x4, x5 negative TEs
- Apply CE algorithm
  - S :disjunction of +ve examples
    - S={x1 OR x2 OR x3}
  - G :negation of disjunction of -ve examples
    - G={*not* (x4 or x5)}
  - Need to use all instances to learn the concept!
- Cannot predict usefully:
  - TEs have unanimous vote
  - other x's have 50/50 vote!
    - For every h in H that predicts +, there is another that predicts -

Inductive Bias

- As constraints on representation of hypotheses
  - Example of limiting connectives to conjunctions
  - Allows learning of generalized hypotheses
  - Introduces bias that depends on hypothesis representation
- Needs formal definition of inductive bias of learning algorithm

Inductive system as an equivalent deductive system

- Inductive bias made explicit in *equivalent deductive system*
  - Logically represented system that produces same outputs (classification) from inputs (TEs, instance x, bias B)
    - *E.g.* The CE procedure

Equivalent deductive system

- Inductive bias (IB) of learning algorithm L:
  - any minimal set of assertions B used to logically infer the value c(x) of any instance x from B, D, and x for any target concept c and training examples D.
    - for a rote learner, B = {}, and there is no IB.
- Difficult to apply in many cases, but a useful guide

Inductive bias and specific learning algorithms

- Rote learners:
  - no IB
- Version space candidate elimination (CE) algorithm:
  - The target concept c can be represented in H
- Find-S:
  - The target concept c can be represented in H;
  - all instances that are not positive are negative.

Computational Complexity of VS

- The *S* set for conjunctive feature vectors
  - linear in the number of features and the number of training examples.
- The *G* set for conjunctive feature vectors
  - exponential in the number of training examples.
- In more expressive languages,
  - both *S* and *G* can grow exponentially.
- The order of processing examples significantly affect computational complexity.

Size of S and G?

- n Boolean attributes
- 1 positive example: (T, T, .., T)
- n/2 negative examples:
  - (F,F,T,..T)
  - (T,T,F,F,T..T)
  - (T,T,T,T,F,F,T..T)
  - ..
  - (T,..T,F,F)
- |S|=1
- $|G|=2^{n/2}$

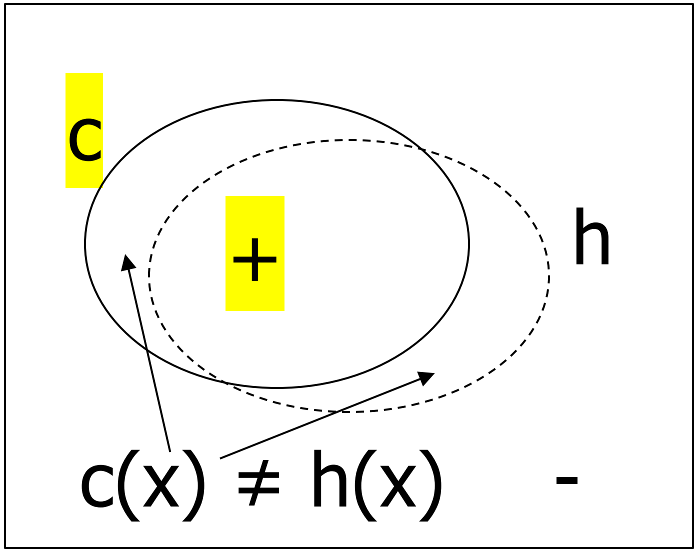
G0: (?,?,...?)

G1: (T,?,..,?), (?,T,?,...?)

G2: (T,?,T,?..,?), (T,?,?,T..,?), (?,T,T,?,...?), (?,T,?,T,...?)

Probably Approximately Correct (PAC) learning model

- A consistent hypothesis
  - Training error: 0
  - True error $\neq$ 0
    - $error_{\mathcal{D}}(h) = P(c(x) \neq h(x))$
    - $\mathcal{D}$: Population distribution



Is it possible to bound true error by minimizing training error?

Probably Approximately Correct (PAC) learning model

- C: Concept class defined over a set of instances X of length n
- L: A learner using hypothesis space H.
- C PAC learnable.
  - If $\forall c \in C$, distribution $\mathcal{D}$ over X, $0 < \varepsilon < ½$ and $0 < \delta < ½$
  - learner L with probability at least $(1- \delta)$ outputs a hypothesis $h \in H$, such that $error_{\mathcal{D}}(h) \leq \varepsilon$,
  - in time that is polynomial in $1/\varepsilon$, $1/\delta$, n and size(c).

to relate sample complexity, running time, and results.

Sample complexity of a learner

- How many training samples required to get a reliable hypothesis with a high probability with a reasonable amount of computation?
  - low true error, minimum number of samples required, polynomial time complexity
  - Equivalently, how many samples required so that the version space consists of every consistent hypothesis bounded by an error ε.
    - $VS_{H,D}$ for a target concept c having such property called ε-exhausted.

Theorem of ε-exhausting the VS

- Given finite H of a target concept c, and m training samples independently randomly drawn forming data D, for any $0 \leq \varepsilon \leq 1$,
  - $P(VS_{H,D}$ is not ε-exhausted$) < |H|e^{-\varepsilon m}$

Proof:
For any h with error $> \varepsilon$, it appears consistent if all m samples are correctly labeled with at most prob. $(1-\varepsilon)^m$.
Let there be k such hypotheses with error $> \varepsilon$.
Prob. that at least one of them would be consistent $\leq k(1-\varepsilon)^m$
As $1-\varepsilon < e^{-\varepsilon} \Rightarrow k(1-\varepsilon)^m < k\, e^{-\varepsilon m} < |H|\, e^{-\varepsilon m}$

Sample complexity of a PAC learner

- Maximum Prob. of providing a consistent hypothesis with error > ε : δ
- Hence, for a PAC learner
  - $|H|e^{-\varepsilon m} \leq \delta$
  - ➔ $m \geq (1/\varepsilon)(\ln |H| + \ln (1/\delta))$
    - An overestimate as size of version space is much smaller than |H|.
- m grows linearly with $1/\varepsilon$ and logarithmically with $1/\delta$
  - Also grows logarithmically with |H|

Example

- C= Target functions of n Boolean attributes in conjunctive forms.
  - Each literal can have three values true, false, and ignore (always 1).
- H=C
  - A hypothesis in the same conjunctive form of a literal
- |H|= ?
  - 3^n
  - Hence, $m \geq (1/\varepsilon)(n \ln 3 + \ln(1/\delta))$
  - Suppose, n=10, ε=0.1 and δ=5%
  - $m \geq (1/.1)(10 \ln 3 + \ln(1/.05)) = 139.82$, i.e. 140

Example

- Learn a concept in the form of **any boolean function** over n variables.
  - Are such concepts PAC-learnable by a consistent learner?
- Hypothesis Space H: all possible functions.
- $|H|$=?
  - 2^{2^n}
- $m \geq (1/\epsilon)$ (ln $|H|$ + ln $(1/\delta)$) =?
  - $(1/\epsilon)$ (2^n+ ln $(1/\delta)$)
- Is it PAC-learnable?
  - NO (Sample complexity not polynomial)

Sample complexity for infinite Hypothesis space

- PAC learners bound Bound for finite hypothesis space not applicable.
- Other sample complexity measure
  - Vapnik-Chervonenkis (VC) dimension

VC Dimension: Sample complexity of infinite H

- Dichotomy on a set of instances S
  - Partitioning into two sets (+ve and –ve examples).
  - No. of all possible dichotomies: $2^{|S|}$
- Shattering by a hypothesis space H
  - If there exists a consistent h for every dichotomy of S.
    - Classification problem
  - Can H distinguish all subsets of S?
    - for any bi-partition ($S1$,$S2$) of S, there exists one h in H such that $h(s)=0$ for each $s \in S1$ and $h(s)=1$ for each $s \in S2$.
- Vapnik-Chervonenkis (VC) dimension:
  - The size of the largest finite subset in X shattered by H.
  - Sufficient to have at least one such instance

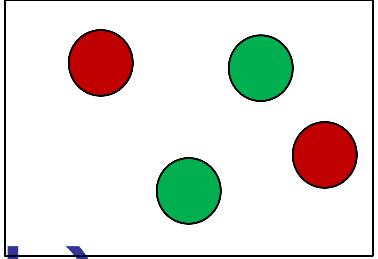
A simple upper bound on VC(H)

- Vapnik-Chervonenkis (VC) dimension:
  - The size of the largest finite subset in X shattered by H.
  - $VC(H) \leq \log_2 |H|$
    - Proof: H requires 2^d distinct hypothesis to shatter d instances.
    - ➔ $2^d < |H|$. Hence, $d=VC(H) < \log_2 |H|$

A few examples

- H= Set of intervals [a,b], in real axis.
  - h(x): 1 if x in [a,b], else 0.
- Shatters any pair of distinct points, e.g., p and q, $p<q$
  - e.g. $[p-2, p-1]$, $[p-\varepsilon, p+\varepsilon]$, $[q-\varepsilon, q+\varepsilon]$ $[q+1,q+2]$
  - Existence of any hypothesis sufficient.
  - Existence of any pair of points being shattered sufficient.
- Say three points $p < q < r$
  - No h shattering {p,r|q} dichotomy.
- VC(H)=2.

VC(H)=3



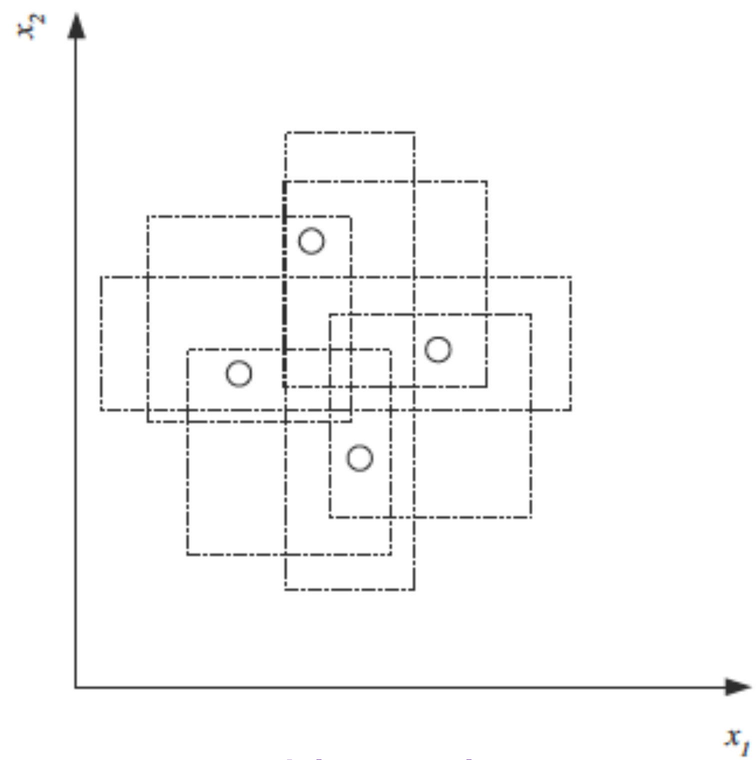
A few examples (Contd.)

In an d-dimensional space with (d-1)-dimensional hyperplanes, VC(H)=d+1.

- H= Set of straight lines in a plane.
  - h(x): 1 if x lies in right half, else 0.
- Shatters any pair of distinct points.
  - All points in one half, two points in two different halves.
- Shatters three non-collinear distinct points
  - All three in one half, two in one half and the other point in the opposite half.
  - Need not be true for all instances of three points.
- No set of 4 distinct points could be shattered
  - There exists a dichotomy, each containing a pair of points, non-separable by a straight line.

Another example

- H= All axis aligned rectangles.
- Shatters maximum 4 points.
- VC(H)=4.
- Enough to find any set of 4 points for shattering.
  - Not required for all 4 points in the space.
  - 4 points in a straight line not shattered.



Not possible to place 5 points anywhere for shattering.

Courtesy: "Introduction to Machine" Learning by Ethem Alpaydin (Chapter 2, Fig. 2.6)

Sample complexity infinite space: A few results

- Upper bound for ϵ-exhausted version space
  - $m \geq (1/\epsilon)(4\log(2/\delta) + 8VC(H)\log(13/\epsilon))$

- Theorem on lower bound:
  - Consider any concept class C such that $VC(C) \geq 2$, any learner L, and any $\epsilon \in (0,1/8)$ and any $\delta \in (0,1/100)$.
  - Then there exists a distribution D and a target concept in C such that if L observes examples fewer than

    $$\max[(1/\epsilon)\log(1/\delta), (VC(C)-1)/(32\epsilon)],$$

    then with probability at least δ, L outputs a hypothesis h having $ED(h) > \epsilon$.

Example: Rectangle learning

- In a 2-dimensional space, consider a class C of concept of form $(a \leq x \leq b) \wedge (c \leq y \leq b)$, where a, b, c, d are real values.
  - Find a number of training examples drawn randomly to assure that for any target in C, any consistent learner using $H = C$ will, with probability at least 95%, output a hypothesis with error at most 0.15.
- Compute VC(H)
  - 4
- Use $m \geq (1/\epsilon)(4\log(2/\delta) + 8VC(H)\log(13/\epsilon))$
  - ϵ =0.15, and δ = .05
  - m $\geq$1515.2 = 1516

VC-dimension: Significance

- Measure of sample complexity.
  - LUT: Rote learner: Infinite VC dimension
  - Sample complexity proportional to VC dimension
- A bit pessimistic measure.
  - Does not consider probability distribution in feature space.
  - A simple model may discern classes (with data points much larger than the VC dimension).

Exercise

- In a 2-dimensional space, consider a class C of concept of form $(a \leq x \leq b) \wedge (c \leq y \leq d)$, where a,b,c,d are integers in [0,99].
  - Find a number of training examples drawn randomly to assure that for any target in C, any consistent learner using $H=C$ will, with probability at least 95%, output a hypothesis with error at most 0.15.

Solution:

- Finite hypothesis space.
- $|H|=?$
  - ${}^{n}C_{2} \times {}^{n}C_{2}$ where $n=100$.
  - $= 24502500$
- $m \geq (1/\varepsilon)(\ln(|H|) + \ln(1/\delta))$
  - $\epsilon = 0.15$, $\delta = .05$, $|H| = 24502500$
  - $m \geq 133.4 = 134$

Handling noise in data

- Three major sources:
  - Imprecision in measurement of features.
  - Error in labeling (Teacher noise).
  - Missing additional attributes in representation (hidden or latent attributes).
- Noise may not provide consistent hypothesis.
- Tolerate training error within a limit to use simpler model.

Effect of inductive bias

- As training data is a small segment of the input space.
  - Smaller the proportion greater the inductive bias.
  - Low training error still may provide high errors on unseen inputs.
    - Generalization error.
  - Higher the proportion of training samples in the input space, better is model fitting and lower generalization error.

To what extent a model trained on the training set predicts the correct output for new instances is called *generalization.*

Matching complexities

- Complexities of model to be matched with the underlying process generating data.
  - Lower complex model → Higher training and generalization error.
    - Underfitting
  - Higher complex model → Low training error, but may have high generalization error.
    - Overfitting
      - Even the chosen complexity is matched, model fitting requires more data point.

Occam’s razor

- Given comparable empirical error, a simple (but not too simple) model would generalize better than a complex model.
  - simpler explanations more plausible and any unnecessary complexity to be shaved off.

Triple trade-off

- A trade-off between three factors in any data driven learning algorithm:
  - the complexity of the hypothesis
  - the amount of training data, and
  - the generalization error on new examples.

Dietterich, T. G. 2003. “Machine Learning.” In Nature Encyclopedia of Cognitive Science. London: Macmillan.

Model selection

- Empirical choice of model complexity
  - Number of parameters
  - Degree of a polynomial for regression
- Divide input in 3 sets:
  - Training, Validation and Test.
  - Increase model complexity by keeping training and validation error low.
    - May adopt cross-validation.
  - Check on generalization error.
- There exist other information theoretic / likelihood ratio based approaches.

Summary

- Concept learning as search through *H*
- General-to-specific ordering over *H*
- Version space candidate elimination algorithm
- *S* and *G* boundaries characterize learner's uncertainty
- Learner can generate useful queries
- Inductive leaps possible only if learner is biased!
- Inductive learners can be modeled as equiv deductive systems
- Concept learning algorithms: unable to handle data with errors
  - Allow forming hypothesis with low training error.
    - e.g. learning decision trees

Summary

- Learning allowing low error
  - Supervised learning of a model given labelled data.
    - Classification
    - Regression
- Three trade-offs of learning
  - Model capacity and complexity.
    - VC-Dimension
      - Maximum number of points shattered by a hypothesis space.
  - Number of labelled samples.
  - Generalization error.
- Empirical choice of model
  - Training, validation and test sets.
  - Three types of errors.
  - Choose model by keeping training and validation errors low.
  - Generalization error indicated by test error.

Thank you!