BITS F464: Machine Learning

12 Concept Learning Part-1

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 12, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/ml

Concept learning

Concept[1] learning. Inferring a boolean-valued function (*hypothesis*) from training examples of its input and output.

Consider a dataset D with following attributes

- **Wind:** Strong/Weak
- **Water:** Warm/Cool
- **Forecast:** Same/Change
- **Sky:** Sunny/cloudy/Rainy
- **AirTemp:** Warm/Cold
- **Humidity:** Normal/High

SN	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	Cool	Change	Yes

[1] Note that "concept" is true function. $h(x) = c(x)$

The hypothesis Function

- For each attribute, the hypothesis will either
  - Indicate by a "**?**" that any value is acceptable for this attribute,
  - Specify a single required value (*e.g.*, Warm) for the attribute, or
  - Indicate by a "ϕ" that no value is acceptable.
- If some instance x satisfies all the constraints of hypothesis h, then h classifies x as a positive example ($h(x) = 1$).
- A hypothesis that favorite sport is enjoyed only on cold days with high humidity (independent of the values of the other attributes) is represented by the expression $(?, Cold, High, ?, ?, ?)$
- The **most general** hypothesis that specifies *"every day is positive"* is represented by
$$(?, ?, ?, ?, ?, ?)$$
- **Most specific** hypothesis that *"no day is positive"* is given by [2]
$$(\phi, \phi, \phi, \phi, \phi, \phi)$$

[2] Specifies more.

Inductive Learning Hypothesis

- The only information available about c is its value over the training examples
- Therefore, algorithms can at best guarantee that the output hypothesis fits the target concept over the training data.

Inductive learning hypothesis

Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

Essentially we are focusing on lower side of efficiency vs. model-complexity plot.

Concept learning as search

- Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation.
- In EnjoySport learning task contains $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$ distinct instances.
- So $5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120$ syntactically distinct hypotheses
- Hypothesis containing one or more ϕ represents the empty set of instances; that is, it classifies every instance as negative
- Although semantically distinct hypotheses are only $1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3) = 973$

General-to-Specific Ordering

- A very useful structure that exists for any concept learning problem is a general-to-specific ordering
- Used for exhaustive search even with infinite hypothesis without explicitly enumerating every hypothesis.

$$h1 = (Sunny, ?, ?, Strong, ?, ?)$$
$$h2 = (Sunny, ?, ?, ?, ?, ?)$$

- As h_2 imposes fewer constraints, it classifies more instances as positive. Any instance classified positive by h_1 will also be classified positive by h_2. Therefore, h_2 is more general than h_1

$h_j \geq_g h_k$

Let h_j and h_k be boolean-valued functions defined over X. Then h_j is more-general-than-or-equal-to h_k if

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

More-general-than Ordering

It is more useful to consider cases where one hypothesis is strictly more general than the other.

$h_j >_g h_k$

If $h_j \geq_g h_k$ and $h_k \not\geq_g h_j$

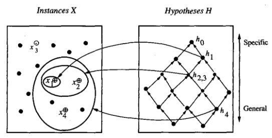
- Sometimes we also say h_j is **more-specific-than** h_k when h_k is more-general-than h_j. [3]

[3]Recall: If $h_j \geq_g h_k$ Then h_j is more-general-than-or-equal-to h_k

Finding a maximally specific hypothesis

Algorithm 1: FIND-S

```
Initialize h to the most specific hypothesis in H
for each positive training instance x do
    for each attribute constraint a_i in h do
        if constraint a_i is NOT satisfied by x then
            replace a_i in h by next more general constraint that is satisfied by x
return hypothesis h
```



$h_0 =< \phi\ \phi\ \phi\ \phi\ \phi\ \phi >$

E1: $x_1 =<$ *Sunny Warm Normal Strong Warm Same* $>, +$

$h_1 =<$ *Sunny Warm Normal Strong Warm Same* $>$

E2: $x_2 =<$ *Sunny Warm High Strong Warm Same* $>, +$

$h_2 =<$ *Sunny Warm ? Strong Warm Same* $>$

E3: $x_3 =<$ *Rainy Cold High Strong Warm Change* $>, -$

$h_3 =<$ *Sunny Warm ? Strong Warm Same* $>$

E4: $x_4 =<$ *Sunny Warm High Strong Cool Change* $>, +$

$h_4 =<$ *Sunny Warm ? Strong ? ?* $>$

FIND-S

- FIND-S provides a way to use more-general-than partial ordering to organize the search.
- It is guaranteed to output the most specific hypothesis within H that is consistent with the positive training examples.
- No way to determine whether it has found the only hypothesis
- It is unclear whether we should prefer this hypothesis over, say, the most general, or some other hypothesis of intermediate generality.
- If training examples will contain some errors or noise then it can severely mislead FIND-S

LIST-THEN-ELIMINATE ALGORITHM

- Outputs a description of the set of all hypotheses consistent with the training examples.
- **Consistent hypothesis.** A hypothesis h is consistent with a set of training examples D if and only if $h(x) = c(x)$ for each example $(x, c(x))$ in D
- **Version space.** Subset of H containing only consistent hypothesis.

Algorithm 2: List-Then-Eliminate

```
VersionSpace ← a list containing every hypothesis in H
for each training example <x, c(x)> ∈ D do
    remove all hypothesis h from the VersionSpace for which h(x) ≠ c(x)
return List of hypothesis in VersionSpace
```


CANDIDATE-ELIMINATION ALGORITHM

- Version space is represented by its most general and least general members.
- **General boundary G,** with respect to hypothesis space H and training data D, is the set of maximally general members of H consistent with D.
- **Specific boundary S,** with respect to hypothesis space H and training data D, is the set of minimally general (*i.e.*, maximally specific) members of H consistent with D.

S: {<Sunny,Warm,?,Strong,?,?>}

<Sunny,?,?,Strong,?,?> <Sunny,Warm,?,?,?,?> <?,Warm,?,Strong,?,?>

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>}

Version Space Representation Theorem

Version space representation theorem

Let X be an arbitrary set of instances and let H be a set of boolean-valued hypotheses defined over X. Let $c : X \rightarrow \{0, 1\}$ be an arbitrary target concept defined over X, and let D be an arbitrary set of training examples $\{(x, c(x))\}$. For all X, H, c, and D such that S and G are well defined,

$$VS_{H,D} = \{h \in H | (\exists g \in G)(\exists s \in S)(g \geq_g h \geq_g s)\}$$

Proof Sketch. It suffices to show that

1. Every h satisfying the right-hand side of the above expression is in $VS_{H,D}$, and
2. Every member of $VS_{H,D}$ satisfies the right-hand side of the expression.

Version space representation theorem

[1] Every h satisfying the right-hand side; is in $VS_{H,D}$

Let $g \in G$ and $s \in S$ such that $g \geq_g h \geq_g s$.
Then by definition s must satisfy all positive examples in D.
Since $h \geq_g s$, h would also satisfy all positive examples in D.
Similarly, by definition of g it cannot satisfy any negative example in D, and because of $g \geq_g h$ it also cannot satisfy any negative example in D. So **this h satisfies all positive and no negative examples of D** therefore, it is in $VS_{H,D}$

[2] Every member of $VS_{H,D}$ satisfies the right-hand side of the expression

It can be proven by assuming some $h \in VS_{H,D}$ that does not satisfy the right-hand side of the expression. Than showing that this leads to an inconsistency.

CANDIDATE-ELIMINATION ALGORITHM

Algorithm 3: Candidate-Elimination

```
Initialize G to the set of maximally general hypothesis in H, G_0 = {?,?,?,?,?,?}
Initialize S to the set of maximally specific hypothesis in H, S_0 = {φ,φ,φ,φ,φ,φ}
for each training example d do
    if d is a positive example then
        Remove from G any hypotheses inconsistent with d
        for each hypothesis s in S that is not consistent with d do
            Remove s from S
            Add to S all minimum generalizations h of s such that – h is consistent
            with d, and some member of G is more general than h
            Remove from S any hypothesis that is more general than another
            hypothesis in S
    if d is a negative example then
        Remove from S any hypotheses inconsistent with d
        for each hypothesis g in G that is not consistent with d do
            Remove g from G
            Add to G all minimum specializations h of g such that – h is consistent
            with d, and some member of S is more specific than h
            Remove from G any hypothesis that is less general than another
            hypothesis in G
```


Example

Training Example

1. $<$ *Sunny*, *Warm*, *Normal*, *Strong*, *Warm*, *Same* $>$ Enjoy Sport = Yes
2. $<$ *Sunny*, *Warm*, *High*, *Strong*, *Warm*, *Same* $>$ Enjoy Sport = Yes

S_0: $\{<\phi,\phi,\phi,\phi,\phi,\phi>\}$

S_1: $\{<$ *Sunny*, *Warm*, *Normal*, *Strong*, *Warm*, *Same* $>\}$

S_2: $\{<$ *Sunny*, *Warm*, ?, *Strong*, *Warm*, *Same* $>\}$

G_0, G_1, G_2: $\{<?,?,?,?,?,?>\}$

Example

Training Example

3. $<$ *Rainy*, *Cold*, *High*, *Strong*, *Warm*, *Change* $>$ Enjoy Sport = No [4]

S_2, S_3: {<Sunny,Warm,?,Strong, Warm, Same>}

G_3: {<Sunny,?,?,?,?,?> <?,Warm,?,?,?,?> <?,?,?,?,?,Same>}

G_2: {<?,?,?,?,?,?>}

[4] $<$ *Sunny*, ?, ?, ?, ?, ? $>$, $<$ *Cloudy*, ?, ?, ?, ?, ? $>$, $<$?, *Warm*, ?, ?, ?, ? $>$, $<$?, ?, *Normal*, ?, ?, ? $>$, $<$?, ?, ?, *Weak*, ?, ? $>$, $<$?, ?, ?, ?, *Cool*, ? $>$, $<$?, ?, ?, ?, ?, *Same* $>$

Example

Training Example

4. $<$ *Sunny*, *Warm*, *High*, *Strong*, *Cool*, *Change* $>$ Enjoy Sport = Yes

S_3: {<Sunny, Warm, ?, Strong, Warm, Same>}

S_4: {<Sunny, Warm, ?, Strong, ?, ?>}

G_4: $\{<$ *Sunny*, ?, ?, ?, ?, ? $>$ $<$?, *Warm*, ?, ?, ?, ? $>\}$

G_3: $\{<$ *Sunny*, ?, ?, ?, ?, ? $>$ $<$?, *Warm*, ?, ?, ?, ? $>$ $<$?, ?, ?, ?, ?, *Same* $>\}$

Example

Training is independent of data ordering.

S_4: {<Sunny,Warm,?,Strong,?,?>}

<Sunny,?,?,Strong,?,?> <Sunny,Warm,?,?,?,?> <?,Warm,?,Strong,?,?>

G_4: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>}

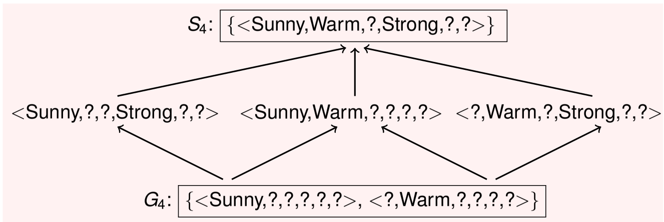
Convergence:

1. If NO errors in training data
2. There exist some hypotheses in the hypothesis space that correctly describes the target concept

Active learning can help

- If learner is allowed to query (instead of teacher providing examples)
- It could choose contradicting instance that would be classified positive by some hypotheses and negative by others.
- Optimal query strategy is to generate instances that satisfy exactly half the hypotheses in the current version space.
- Size of the version space is reduced by half with each new example, and the correct target concept can therefore be found with only $\lceil \log(|VS|) \rceil$ number of experiments.

Partially Learned Concept



Classify

1. $A = (Sunny, Warm, Normal, Strong, Cool, Change)$ $+ve$
2. $B = (Rainy, Cold, Normal, Light, Warm, Same)$ $-ve$
3. $C = (Sunny, Warm, Normal, Light, Warm, Same)$?
4. $D = (Sunny, Cold, Normal, Strong, Warm, Same)$ $-ve$

Biased hypothesis space

When hypothesis space is restricted to include only conjunctions of attribute then we may have a problem.

$$h = \{\phi, \phi, \phi, \phi, \phi, \phi\}$$

Tr Example $(Sunny, Warm, Normal, Strong, Cool, Change)$ as $+ve$

$$h = (Sunny, Warm, Normal, Strong, Cool, Change)$$

Tr Example $(Cloudy, Warm, Normal, Strong, Cool, Change)$ as $+ve$

$$h = (?, Warm, Normal, Strong, Cool, Change)$$

Tr Example $(Rainy, Warm, Normal, Strong, Cool, Change)$ as $-ve$

$$h = \phi$$

OOPS! NO hypothesis for this training data.

Unbiased Learner

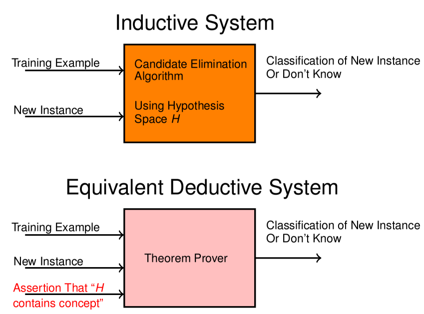
- Obvious solution is to make hypothesis space capable of representing every possible subset of the instances X.
- Use set of all subsets of a set X that is called the power set of X.
- In current example, *EnjoySport* learning task contains $3 \times 2 \times 2 \times 2 \times 2 = 96$ distinct instances therefore hypothesis space becomes 2^{96} or 10^{28} as opposed to 973 in case of conjunctions.
- "Sky = Sunny or Sky = Cloudy" is represented as $(Sunny, ?, ?, ?, ?, ?) \vee (Cloudy, ?, ?, ?, ?, ?)$
- While eliminating problems of expressibility it introduce problem in learning algorithm that is now **unable to generalize**
- With three positive (x_1, x_2, x_3) and two negative (x_4, x_5) it looks like $S : \{(x_1 \vee x_2 \vee x_3)\}$ and $G : \{\neg(x_4 \vee x_5)\}$

Futility of Bias-Free Learning

- Is Bias-Free Learning useless?
- **Fundamental property of inductive inference:** a learner that makes NO a priori assumptions regarding the identity of the target concept has no rational basis for classifying any unseen instances.
- Inductive learning requires inductive bias
- Let classification of x_i by algorithm L trained on D_c be $L(x_i, D_c)$
- **Inductive inference** is represented as $(D_c \wedge x_i) \succ L(x_i, D_c)$
- Let us represent **inductive bias** by B, then $(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$
- Inductive bias B, is the assumption that $c \in H$
- This assumption enables deduction (proof)

Inductive bias of a learner is defined as the set of additional assumptions B, sufficient to justify its inductive inferences as deductive inferences.

Inductive and Deductive Learning



This assertion is the inductive bias. These two systems will produce identical outputs for every input set of training examples and every new instance in X.

Comparison of different learners based on bias

- **ROTE-LEARNER:** Learning corresponds to simply storing each observed training example in memory. New instances are classified by looking them up in memory. If the instance is found, the stored classification is returned. Otherwise, the system refuses to classify the new instance. (no inductive bias)
- **CANDIDATE-ELIMINATION:** New instances are classified only in the case where all members of the current version space agree on the classification. Otherwise, the system refuses to classify the new instance. (bias: target concept can be represented in its hypothesis space)
- **FIND-S:** Finds the most specific hypothesis consistent with the training examples. It then uses this hypothesis to classify all subsequent instances. (bias: target concept can be described in its hypothesis space + all instances are negative instances unless the opposite is entailed by its other knowledge)

Thank You!

Thank you very much for your attention! (Ref [5])

Queries ?

[5] [1] Book - *Machine Learning*, ch-2, Tom M. Mitchell.