Version Spaces

Instructor: Dr. Muhammad Fahim

Contents

- Version Spaces
- The List-Then-Eliminate Algorithm
- Representing Version Spaces
- Candidate Elimination Algorithm
- Example of Candidate Elimination Algorithm

Version Spaces

- 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 *<x,c(x)>* in D.

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

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

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

The List-Then-Eliminate Algorithm

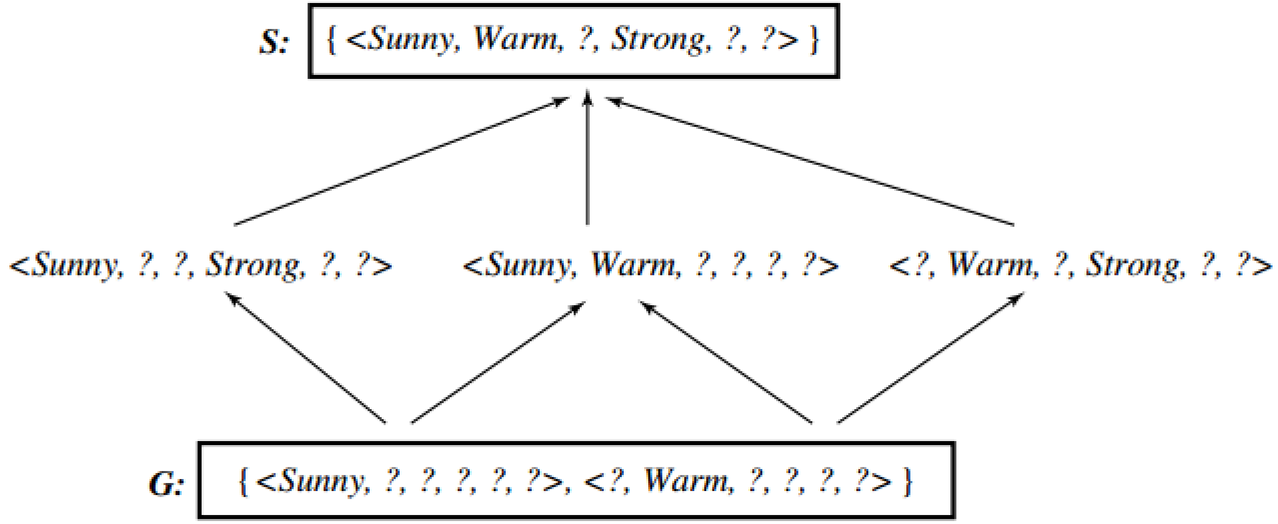
1. VersionSpace ← a list containing every hypothesis in H

2. For each training example <x,c(x)>
   remove from VersionSpace any hypothesis that is inconsistent with any training example $h(x) \neq c(x)$

3. Output the list of hypotheses in VersionSpace

Example Version Space

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes



Representing Version Spaces

- The General boundary G, of version space $VS_{H,D}$ is the set of its maximally general members

- The Specific boundary S, of version space $VS_{H,D}$ is the set of its maximally specific members

- Every member of the version space lies between these boundaries

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

Observations!!

- If insufficient data is available to narrow the version space to a single hypothesis, then the algorithm can output the entire set of hypotheses consistent with the observed data.

Observations!!

- In principle, the List-Then-Eliminate algorithm can be applied whenever the hypothesis space H is finite.
- It has many advantages, including the fact that it is guaranteed to output all hypotheses consistent with the training data.
- Unfortunately, it requires exhaustively enumerating all hypotheses in H

Candidate Elimination Algorithm

Candidate Elimination Algorithm (CEA)

- It is another approach to concept learning

The key idea in the Candidate Elimination algorithm is to output a set of all hypotheses consistent with the training examples

Candidate Elimination Algorithm (CEA)

- It begins by initializing the version space to the set of all hypotheses, by initializing the
  - G as {?, ?, …, ?, ?} and
  - S as {∅, ∅, … , ∅, ∅} respectively

- As each training example is considered, the S boundary is generalized and the G boundary is specialized, to eliminate from the version space any hypotheses found inconsistent with the new training example.

Candidate Elimination Algorithm

Initialize G to the set of maximally general hypotheses in H
Initialize S to the set of maximally specific hypotheses in H
For each training example d, do

- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to S all minimal 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
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal 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

CEA – Example

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Step 1

- $S_0 = \{\varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing\}$ Set of maximally specific hypothesis
- $G_0 = \{?, ?, ?, ?, ?, ?\}$ Set of maximally general hypothesis

CEA – Example

Step 2 If d is positive example

S_0: {<∅, ∅, ∅, ∅, ∅, ∅>}

↓

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

↓

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

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

Training examples:

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

CEA – Example

Step 3

If d is negative example

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

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

Training Example:

3. <Rainy, Cold, High, Strong, Warm, Change>, EnjoySport=No

CEA – Example

Step 3 If d is negative example

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

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

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

Training Example:

3. <*Rainy, Cold, High, Strong, Warm, Change*>, *EnjoySport=No*

CEA – Example

Step 2 Repeat

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

↓

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

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

↑

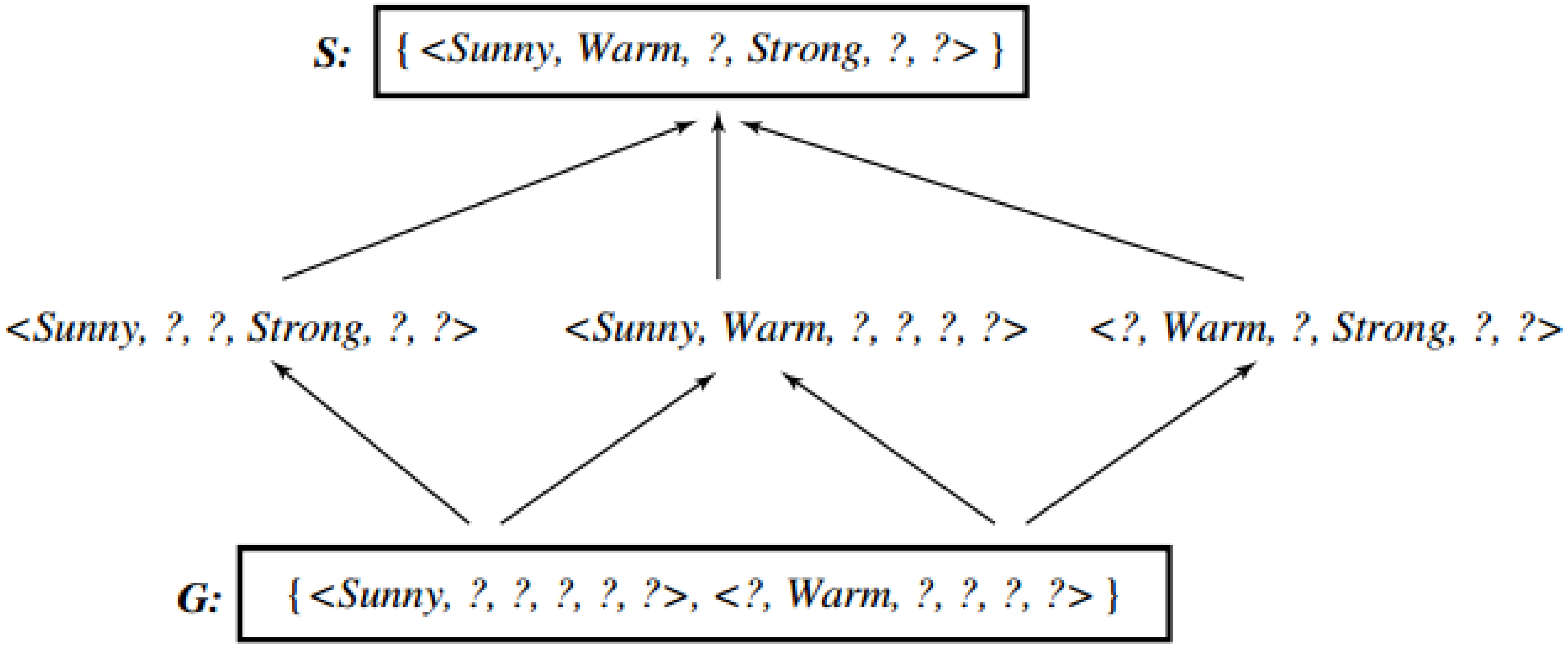
G_3: {<*Sunny, ?, ?, ?, ?, ?*> <*?, Warm, ?, ?, ?, ?*> <*?, ?, ?, ?, ?, Same*>}

Training Example:

4. <*Sunny, Warm, High, Strong, Cool, Change*>, *EnjoySport* = *Yes*

CEA – Example

Final version space is:



CEA – Properties

- If there is a consistent hypothesis then the algorithm will converge to $S = G = \{h\}$ when enough examples are provided

- False examples may cause the removal of the correct h

- If the examples are inconsistent, S and G become empty

- This can also happen, when the concept to be learned is not in H

Remarks on CEA!

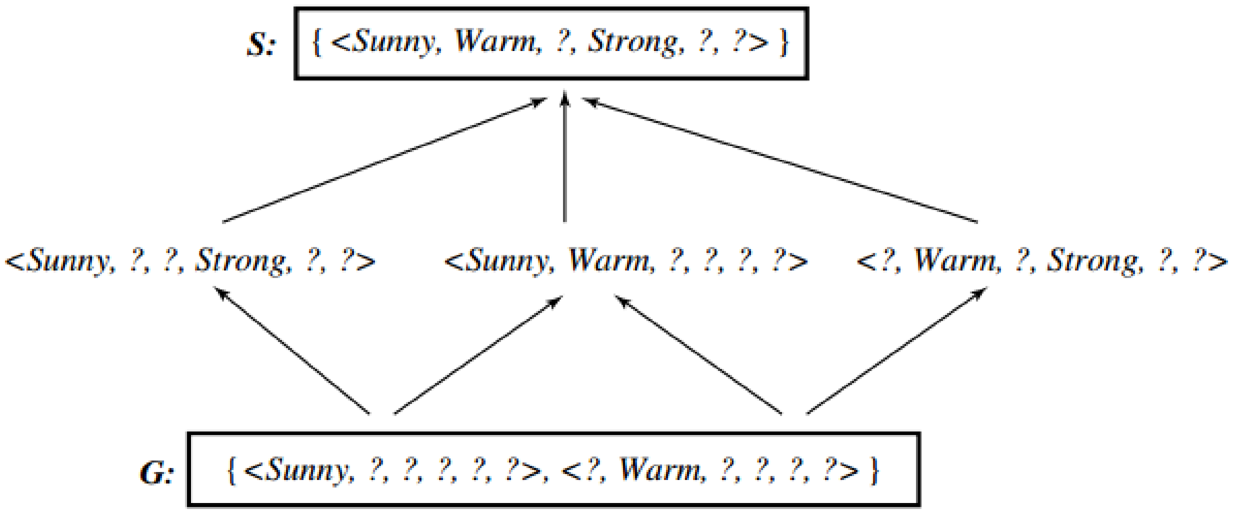
- Will CEA converge to the correct hypothesis?

- **Answer**
- The version space learned by the CEA will converge toward the hypothesis that correctly describes the target concept, provided that
  - there are no errors in the training examples
  - there is some hypothesis in H that correctly describes the target concept.

Classification of Unseen Data

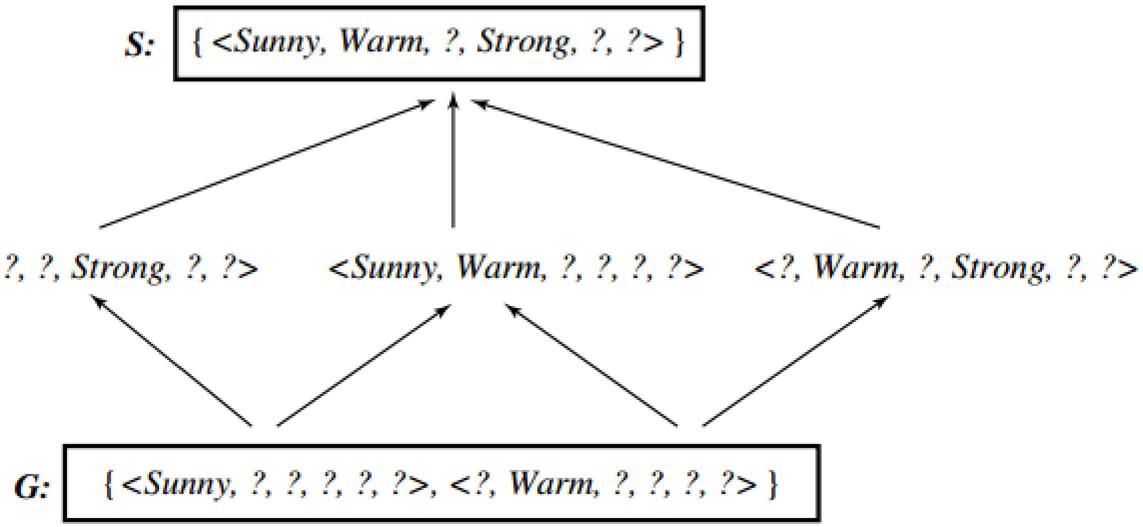
- Classify a new example as positive or negative, if all hypotheses in the version space agree in their classification

- **Otherwise:**
  - Rejection *or*
  - Majority vote



NOTE: The VS can be represented more simply with the S and G boundaries.

Classification of Unseen Data



Instance	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*	
A	Sunny	Warm	Normal	Strong	Cool	Change	?	+ 6/0
B	Rainy	Cold	Normal	Light	Warm	Same	?	- 0/6
C	Sunny	Warm	Normal	Light	Warm	Same	?	? 3/3
D	Sunny	Cold	Normal	Strong	Warm	Same	?	? 2/4

A Biased Hypothesis Space

Day	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*WaterSport*
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No
							class

A Biased Hypothesis Space

x_1 = <Sunny Warm Normal Strong Cool Change> +
x_2 = <Cloudy Warm Normal Strong Cool Change> +

S : { <?, Warm, Normal, Strong, Cool, Change> }

x_3 = <Rainy Warm Normal Light Warm Same> -

S : {}

Given our previous choice of the hypothesis space representation, **no hypothesis is consistent with the above examples**: we have ***BIASED*** the learner to consider only conjunctive hypotheses

An Unbiased Learner

- In order to solve the problem caused by the bias of the hypothesis space, we can remove this bias and allow the hypotheses to represent every possible subset of instances.

- The previous examples could then be expressed as:
  <Sunny, ?,?,?,?,?> v <Cloudy,?,?,?,?,?,?>

- ***However, such an unbiased learner is not able to generalize beyond the observed examples!!!!***

The Futility of Bias-Free Learning

- **Fundamental Property of Inductive Learning** A learner that makes no a priori assumptions regarding the identity of the target concept has no rational basis for classifying any unseen instances.

- **We constantly have recourse to inductive biases** *Example:* we all know that the sun will rise tomorrow. Although we cannot *deduce* that it will do so based on the fact that it rose today, yesterday, the day before, etc., we do take this **leap of faith** or use this **inductive bias**, naturally!

Ranking Inductive Learners according to their Biases

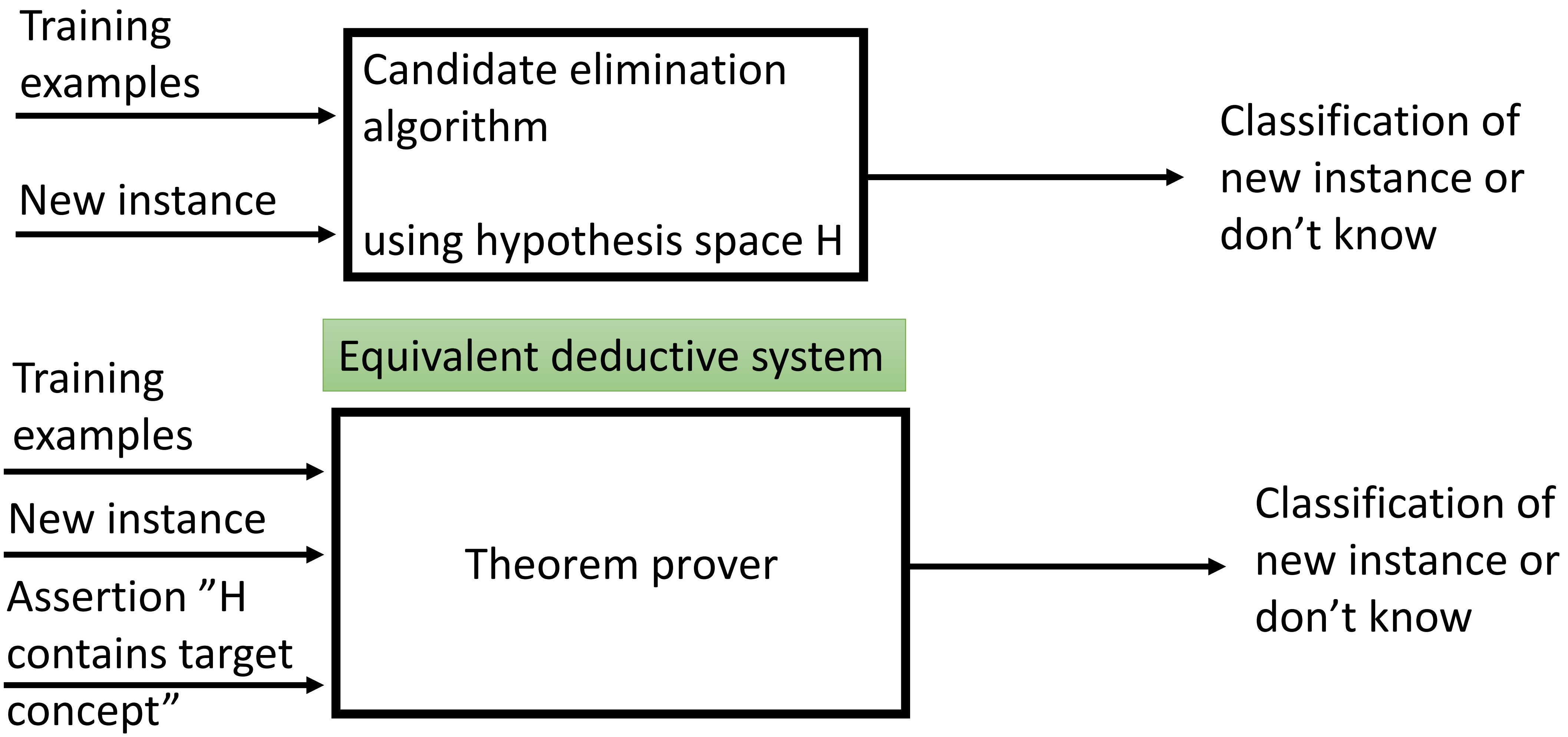
Weak

Bias Strength

- **Rote-Learner:** This system simply memorizes the training data and their classification--- No generalization is involved (No inductive bias)
- **Candidate-Elimination:** New instances are classified only if all the hypotheses in the version space agree on the classification
- **Find-S:** New instances are classified using the most specific hypothesis consistent with the training data

Strong

Inductive Systems and Equivalent Deductive Systems



Announcement

Lecture # 03

Reference

- 2nd chapter of Tom M. Mitchell's book

Thank You ☺