Sardar Patel Institute of Technology,Mumbai
Department of Electronics and Telecommunication Engineering
B.E. Sem-VII
EC433 - Artificial Intelligence and Machine Learning
**Experiment: Candidate Elimination algorithm**

**Name: Abhinav Pallayil** **Date: 19/09/2022**

**Objective: Implement and demonstrate the Candidate-Elimination algorithm to output adescription of the set of all hypotheses consistent with the training example that are givenas per .CSV file.**

**Outcomes:**

1. Representation of version space.
2. Apply the Candidate Elimination algorithm on the given data to get version space.
3. Interpret the output of Candidate Elimination.

**System Requirements:**

Linux OS with Python and libraries or R or windows with MATLAB

**Theory:**

The candidate elimination algorithm incrementally builds the version space given a hypothesis space H and a set E of examples. The examples are added one by one; each example possibly shrinks the version space by removing the hypotheses that are inconsistent with the example. The candidate elimination algorithm does this by updating the general and specific boundary for each new example.

**Procedure** CandidateEliminationLearner(X,Y,E,H)
Inputs
X: set of input features, X={X1,...,Xn}
Y: target feature
E: set of examples from which to learn
H: hypothesis space
Output
general boundary G⊆H
specific boundary S⊆H consistent with E
Local
G: set of hypotheses in H
S: set of hypotheses in H
Let G={true}, S={false};
for each e∈E do
if ( e is a positive example) then
Elements of G that classify e as negative are removed from G;
Each element s of S that classifies e as negative is removed and replaced by the minimal generalizations of s that classify e as positive and are less

general than some member of G;
Non-maximal hypotheses are removed from S;

else

Elements of S that classify e as positive are removed from S;
Each element g of G that classifies e as positive is removed and replaced by the minimal specializations of g that classifies e as negative and are more general than some members of S.
Non-minimal hypotheses are removed from G.

**Properties of Candidate Elimination**

FIND-S outputs a hypothesis from H that is consistent with the training examples; this is just one of many hypotheses from H that might fit the training data equally well.
The key idea in the Candidate-Elimination algorithm is to output a description of the set of all hypotheses consistent with the training examples.

– Candidate-Elimination algorithm computes the description of this set without explicitly enumerating all of its members.
– This is accomplished by using the more-general-than partial ordering and maintaining a compact representation of the set of consistent hypotheses.

**Problem Statement:** To predict whether it is a malignant tumor or not.

**Dataset:**
Number of Instances:9
Number of Attributes: 5

| | shape | size | color | surface | thickness | Malignant |
|---|---|---|---|---|---|---|
| 0 | circular | large | light | smooth | thick | yes |
| 1 | circular | large | light | irregular | thick | yes |
| 2 | circular | large | light | smooth | thick | yes |
| 3 | oval | large | dark | smooth | thin | no |
| 4 | oval | large | light | irregular | thick | yes |
| 5 | oval | large | dark | smooth | thin | no |
| 6 | circular | small | light | smooth | thick | yes |
| 7 | oval | small | light | irregular | thick | yes |
| 8 | circular | small | light | smooth | thick | yes |

Attribute Information:
1. shape: 2 (circular, oval)
2. size: 2 (large, small)
3. color: 2 (light, dark)
4. surface: 2 (smooth, irregular)
5. thickness: 2 (thick, thin)

**Code:**

```
import numpy as np
import pandas as pd
data=pd.read_csv('C:/Users/Dell/Downloads/A.csv')
concepts = np.array(data.iloc[:,0:-1])
print("\nInstances are:\n",concepts)
target = np.array(data.iloc[:,-1])
print("\nTarget Values are: ",target)

def learn(concepts, target):
    specific_h = concepts[0].copy()
    print("\nInitialization of specific_h and genearal_h")
    print("\nSpecific Boundary: ", specific_h)
    general_h = [["?" for i in range(len(specific_h))] for i in range(len(specific_h))]
    print("\nGeneric Boundary: ",general_h)

    for i, h in enumerate(concepts):
        print("\nInstance", i+1 , "is ", h)
        if target[i] == "yes":
            print("Instance is Positive ")
            for x in range(len(specific_h)):
                if h[x]!= specific_h[x]:
                    specific_h[x] ='?'
                    general_h[x][x] ='?'

        if target[i] == "no":
            print("Instance is Negative ")
            for x in range(len(specific_h)):
                if h[x]!= specific_h[x]:
                    general_h[x][x] = specific_h[x]
                else:
                    general_h[x][x] = '?'

        print("Specific Bundary after ", i+1, "Instance is ", specific_h)
        print("Generic Boundary after ", i+1, "Instance is ", general_h)
        print("\n")

    indices = [i for i, val in enumerate(general_h) if val == ['?', '?', '?', '?', '?', '?']]
    for i in indices:
        general_h.remove(['?', '?', '?', '?', '?', '?'])
    return specific_h, general_h

s_final, g_final = learn(concepts, target)

print("Final Specific_h: ", s_final, sep="\n")
print("Final General_h: ", g_final, sep="\n")
```

**Output:**

```
Initialization of specific_h and genearal_h

Specific Boundary:  ['circular' 'large' 'light' 'smooth' 'thick']

Generic Boundary:  [['?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?']]

Instance 1 is  ['circular' 'large' 'light' 'smooth' 'thick']
Instance is Positive
Specific Bundary after  1 Instance is  ['circular' 'large' 'light' 'smooth'
'thick']
Generic Boundary after  1 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?']]



Instance 2 is  ['circular' 'large' 'light' 'irregular' 'thick']
Instance is Positive
Specific Bundary after  2 Instance is  ['circular' 'large' 'light' '?' 'thick']
Generic Boundary after  2 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?']]



Instance 3 is  ['circular' 'large' 'light' 'smooth' 'thick']
Instance is Positive
Specific Bundary after  3 Instance is  ['circular' 'large' 'light' '?' 'thick']
Generic Boundary after  3 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?']]



Instance 4 is  ['oval' 'large' 'dark' 'smooth' 'thin']
Instance is Negative
Specific Bundary after  4 Instance is  ['circular' 'large' 'light' '?' 'thick']
Generic Boundary after  4 Instance is  [['circular', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?'], ['?', '?', 'light', '?', '?'], ['?', '?', '?', '?', '?'],
['?', '?', '?', '?', 'thick']]



Instance 5 is  ['oval' 'large' 'light' 'irregular' 'thick']
Instance is Positive
Specific Bundary after  5 Instance is  ['?' 'large' 'light' '?' 'thick']
```

```
Generic Boundary after  5 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', 'light', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', 'thick']]



Instance 6 is  ['oval' 'large' 'dark' 'smooth' 'thin']
Instance is Negative
Specific Bundary after  6 Instance is  ['?' 'large' 'light' '?' 'thick']
Generic Boundary after  6 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', 'light', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', 'thick']]



Instance 7 is  ['circular' 'small' 'light' 'smooth' 'thick']
Instance is Positive
Specific Bundary after  7 Instance is  ['?' '?' 'light' '?' 'thick']
Generic Boundary after  7 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', 'light', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', 'thick']]



Instance 8 is  ['oval' 'small' 'light' 'irregular' 'thick']
Instance is Positive
Specific Bundary after  8 Instance is  ['?' '?' 'light' '?' 'thick']
Generic Boundary after  8 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', 'light', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', 'thick']]



Instance 9 is  ['circular' 'small' 'light' 'smooth' 'thick']
Instance is Positive
Specific Bundary after  9 Instance is  ['?' '?' 'light' '?' 'thick']
Generic Boundary after  9 Instance is  [['?', '?', '?', '?', '?'], ['?', '?',
'?', '?', '?'], ['?', '?', 'light', '?', '?'], ['?', '?', '?', '?', '?'], ['?',
'?', '?', '?', 'thick']]


Final Specific_h:
['?' '?' 'light' '?' 'thick']
Final General_h:
[['?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?'], ['?', '?', 'light', '?',
'?'], ['?', '?', '?', '?', '?'], ['?', '?', '?', '?', 'thick']]
```

**Conclusion**: Version Space by Candidate Elimination Algorithm for given data set is:

```
[['?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?'], ['?', '?', 'light', '?',
'?'], ['?', '?', '?', '?', '?'], ['?', '?', '?', '?', 'thick']], ['?' '?'
'light' '?' 'thick']
```

For the general hypothesis in case of negative samples, previous general hypothesis is compared with the current specific and common ones are rejected while different ones are listed in the general hypothesis. We do this for all the samples one by one.

**Questions:**

Q1) compare between inductive and deductive learning.

Ans) Inductive learning creates a generalisation from the input data. We basically try to map the relationship between the input and output. It is a bottom-up approach and is used in various industrial applications. In this method we derive conclusions or generalize it from the given facts i.e. the data. It is similar to supervised learning. One example is Find-S algorithm where only positive samples are used to generalise to the data.
Deductive learning is similar to inductive learning. It also tries to fit in a hypothesis, but in addition it also relies on already available information to derive facts and information to give a valid conclusion. It is a top-down approach. The one major thing to note is that in deductive learning, the results are certain i.e, it is either yes or no. Whereas it's probability-based on inductive learning i.e, it can range from strong to weak.

Q2) justify candidate elimination is deductive learning.

Ans) In Candidate learning algorithm, both positive and negative samples are considered because it uses all available information and facts to provide a conclusion. It has positive samples for better generalisation and negative samples to prevent overgeneralisation. It is a step by step process where each sample is compared with previous specific hypothesis if it is a positive one and with previous general if it is a negative one. Thus we get final specific and general hypothesis from the algorithm and then we can find the version space by getting all common values from the final hypothesis.