Assignment - 2

ML – Candidate Elimination Algorithm

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.

- You can consider this as an extended form of Find-S algorithm.
- Consider both positive and negative examples.
- Actually, positive examples are used here as Find-S algorithm (Basically they are generalizing from the specification).
- While the negative example is specified from generalize form.

Terms Used:

- **Concept learning:** Concept learning is basically learning task of the machine (Learn by Train data)
- **General Hypothesis:** Not Specifying features to learn the machine.
- **G = {'?', '?','?','?'...}:** Number of attributes
- **Specific Hypothesis:** Specifying features to learn machine (Specific feature)
- **S= {'pi','pi','pi'...}:** Number of pi depends on number of attributes.
- **Version Space:** It is intermediate of general hypothesis and Specific hypothesis. It not only just written one hypothesis but a set of all possible hypothesis based on training data-set.

Algorithm:

Step1: Load Data set
Step2: Initialize General Hypothesis and Specific Hypothesis.
Step3: For each training example
Step4: If example is positive example
    if attribute value == hypothesis value:
      Do nothing
    else:
      replace attribute value with '?' (Basically generalizing it)
Step5: If example is Negative example
    Make generalize hypothesis more specific.

Example:

Consider the dataset given below:

Sky	Temperature	Humid	Wind	Water	Forest	Output
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

Algorithmic steps:

Initially : G = [[?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?],
[?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?]]
S = [Null, Null, Null, Null, Null, Null]

For instance 1 : <'sunny','warm','normal','strong','warm ','same'> and positive output.
G1 = G
S1 = ['sunny','warm','normal','strong','warm ','same']

For instance 2 : <'sunny','warm','high','strong','warm ','same'> and positive output.
G2 = G
S2 = ['sunny','warm',?,'strong','warm ','same']

For instance 3 : <'rainy','cold','high','strong','warm ','change'> and negative output.
G3 = [['sunny', ?, ?, ?, ?, ?], [?, 'warm', ?, ?, ?, ?], [?, ?, ?, ?, ?, ?],
[?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, 'change']]
S3 = S2

For instance 4 : <'sunny','warm','high','strong','cool','change'> and positive output.
G4 = G3
S4 = ['sunny','warm',?,'strong', ?, ?]

At last, by synchronizing the G4 and S4 algorithm produce the output.

Candidate Elimination

↳ Concept learning
  ↳ General hypothesis
  ↳ Specific hypothesis
  ↳ Version Space

↳ Uses version space
↳ Considers both +ve and -ve results.

↳ We have both specific and general hypothesis
↳ For a +ve example:
  We tend to generalize specific hypothesis

↳ For a -ve example:
  We tend to make general hypothesis more specific.

Version space ⇒ It is intermediate space between specific hyp. and General hyp.

$S = \{\phi, \phi, \ldots \phi\}$

$G = \{?, ?, \ldots ?\}$

$S = \{\phi, \phi, \phi, \phi, \phi\}$ ↓

$G = \{?, ?, ?, ?, ?\}$ ↑

Algorithm :-

→ Initialize G and S as most general and specific hypothesis.

→ For each example, e :-
  if e is +ve:
    → make specific hypothesis more general
  else
    → make general hypothesis more specific

Exp	Sky	Airtemp	Humidity	Wind	Water	forecast	Enjoy sport
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

S_0 $\{\phi, \phi, \phi, \phi, \phi, \phi\}$

S_1 Sunny, Warm, Normal, Strong, Warm, Same

S_2 Sunny, Warm, ?, Strong, Warm, Same

$S_3 = S_2$

$S_4 \Rightarrow$ Sunny, Warm, ?, Strong, ?, ?

$\Rightarrow$ Version Space

Sunny, Warm, ?, ?, ?, ?

Sunny, ?, ?, Strong, ?, ?

Sunny, Warm, ?, ?, ?, ?

?, Warm, ?, Strong, ?, ?

$G_4 \Rightarrow$ Sunny, ?, ?, ?, ?, ? ; ?, Warm, ?, ?, ?, ?

G_3 Sunny, ?, ?, ?, ?, ? ; ?, Warm, ?, ?, ?, ? ; ?, ?, Normal, ?, ?, ? ; ?, ?, ?, ?, Cool, ? ; ?, ?, ?, ?, ?, Same

$G_2 = G_1 = G_0$

$G_1 = G_0$

G_0 ?, ?, ?, ?, ?, ?

Assignment-2

Objective: **For a given set of training examples, stored in a .csv file, implement and demonstrate Candidate Elimination Algorithm. Output a description of the set of all hypothesis consistent with the training examples. for finding the most specific hypothesis based on a given training data samples.**

Step-1: Importing necessory libraries.

- numpy for numerical operations.
- pandas for reading csv.

In [4]:
```
import numpy as np
import pandas as pd
```

Step-2: Reading csv file.

In [8]:
```
df = pd.read_csv("[redacted]EnjoySports.csv")
```

Step-3: Inspection of data and understanding it.

In [9]:
```
df.head()
```

Out[9]:

	f1	f2	f3	f4	f5	f6	target
0	Sunny	warm	normal	strong	warm	same	Yes
1	Sunny	warm	high	strong	warm	same	Yes
2	Rainy	cold	high	strong	warm	change	No
3	Sunny	warm	high	strong	cool	same	Yes

About Code (Just for knowledge):

- This df.head() code prints first 5 lines of your dataframe.

About Data:

- Total 7 columns are there.
- f1,f2,f3,f4,f5 and f6 are features.
- target is our output column.
- target column has decision Yes and No

In [10]:
```
df.shape
```

```
(4, 7)
```

ut[10]:

This shows that in our dataframe total 14 raws and 6 columns are available.

n [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   f1      4 non-null      object
 1   f2      4 non-null      object
 2   f3      4 non-null      object
 3   f4      4 non-null      object
 4   f5      4 non-null      object
 5   f6      4 non-null      object
 6   target  4 non-null      object
dtypes: object(7)
memory usage: 352.0+ bytes
```

This shows details of your dataframe:

- Data type and non null value of each feature and target.
- memory usage of this dataframe.

Step-4: Seperating features and target

In [19]:
```
# features is all features where [:,:-1] represents [rows, columns] i.e. all rows and a
concepts= np.array(df)[:,:-1]
# target is last column where [:,-1] represents [rows, columns] i.e. all rows and last
target = np.array(df)[:,-1]
```

Setp-5: Implementing Candidate Elimination algorithm

- Read/ Scan target of first training data set.
- If it is "Yes", then update Specific_h and it is "No", then update generah_h.
- Specific_h is equals to first positive event.
- Repeat step-1 for all the training sets.
- for all features check, if this hypothesis matches with previous one, keep is same else replace it with '?' means generic.
- at the end generate version space using final set of specific_h and general_h, which will be our result.

In [24]:
```
def learn(concepts, target):
    specific_h = concepts[0]
    print("\nInitialization of specific_h and general_h")
    specific_h = ["0" for i in range(len(specific_h))]
    print("\n So: ", specific_h)
    general_h = ["?" for i in range(len(specific_h))]
    print("\n Go: ", general_h)

    # print("\nSpecific Boundary: ", specific_h)
    general_h = [["?" for i in range(len(specific_h))] for i in range(len(specific_h))]
```

```
for i, h in enumerate(concepts):
    print("\nInstance", i+1 , "is ", h)
    if target[i] == "yes":
        print("Instance is Positive ")
        specific_h = concepts[0]
        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("S", i+1, "is", specific_h)
       print("G", i+1, "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

# calling function
s_final, g_final = learn(concepts, target)

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

```
Initialization of specific_h and general_h

 So:  ['0', '0', '0', '0', '0', '0']

 Go:  ['?', '?', '?', '?', '?', '?']

Instance 1 is  ['Sunny' 'warm' '?' 'strong' '?' 'same']
Instance is Positive
S 1 is ['Sunny' 'warm' '?' 'strong' '?' 'same']
G 1 is [['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?',
'?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?', '?']]


Instance 2 is  ['Sunny' 'warm' 'high' 'strong' 'warm' 'same']
Instance is Positive
S 2 is ['Sunny' 'warm' '?' 'strong' '?' 'same']
G 2 is [['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?',
'?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?', '?']]


Instance 3 is  ['Rainy' 'cold' 'high' 'strong' 'warm' 'change']
Instance is Negative
S 3 is ['Sunny' 'warm' '?' 'strong' '?' 'same']
G 3 is [['Sunny', '?', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?', '?'], ['?', '?', '?', '?', '?',
'?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', 'same']]
'?'], ['?', '?', '?', '?', '?', '?',
```