Dayananda Sagar College of Engineering

Department of Electronics and Communication Engineering

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru – 560 078.

(An Autonomous Institute affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment and Accreditation Council (NAAC) with 'A' grade

Assignment

Program: B.E.	Branch: ECE
Course: Machine Learning	Semester : 6
Course Code:18EC6DECML	Date:02.07.2021

A Report on

Find S Algorithm

Machine Learning Assignment

Submitted by

USN	NAME
1DS18EC143	**Sameer Gautam**

Faculty In-charge

Dr. K N Pushpalatha

Signature of Faculty In-charge

OVERVIEW

The find-S algorithm is a basic concept learning algorithm in machine learning. The find-S algorithm finds the most specific hypothesis that fits all the positive examples. We have to note here that the algorithm considers only those positive training example. The find-S algorithm starts with the most specific hypothesis and generalizes this hypothesis each time it fails to classify an observed positive training data. Hence, the Find-S algorithm moves from the most specific hypothesis to the most general hypothesis.

Important Representation :

- ? indicates that any value is acceptable for the attribute.
- specify a single required value (e.g., Cold) for the attribute.
- Φ indicates that no value is acceptable.
- The most general hypothesis is represented by: {?, ?, ?, ?, ?, ?}
- The most specific hypothesis is represented by: {ϕ, ϕ, ϕ, ϕ, ϕ, ϕ}

Steps Involved In Find-S :

1. Start with the most specific hypothesis h = {ϕ, ϕ, ϕ, ϕ, ϕ, ϕ}
2. Take the next example and if it is negative, then no changes occur to the hypothesis.
3. If the example is positive and we find that our initial hypothesis is too specific then we update our current hypothesis to a general condition.
4. Keep repeating the above steps till all the training examples are complete.
5. After we have completed all the training examples we will have the final hypothesis when can use to classify the new examples.

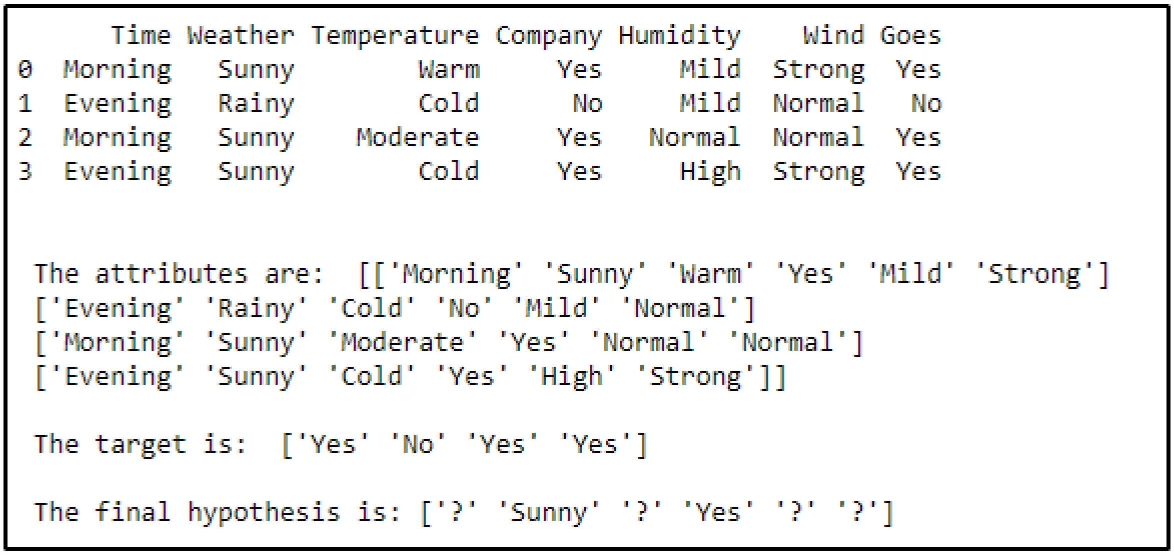
ALGORITHM

```
1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
   For each attribute constraint a, in h
      If the constraint a, is satisfied by x
      Then do nothing
      Else replace a, in h by the next more general constraint that is satisfied by x
3. Output hypothesis h
```

PROGRAM

```
import pandas as pd
import numpy as np

#to read the data in the csv file
file=pd.read_csv(r"F:\goes.csv")
print(file)

# making an array of all the attributes
d = np.array(file)[:, :-1]

# segragating the target that has positive and negative examples
target = np.array(file)[:, -1]

# training function to implement find-s algorithm
def train(c, t):
    for i, val in enumerate(t):
        if val == "Yes":
            specific_hypothesis = c[i].copy()
            break

    for i, val in enumerate(c):
        if t[i] == "Yes":
            for x in range(len(specific_hypothesis)):
                if val[x] != specific_hypothesis[x]:
                    specific_hypothesis[x] = '?'
                else:
                    pass

    return specific_hypothesis

print(" ")

# obtaining the final hypothesis
print("n The final hypothesis is:", train(d, target))
```

RESULTS



```
      Time Weather Temperature Company Humidity    Wind Goes
0  Morning   Sunny        Warm     Yes     Mild  Strong  Yes
1  Evening   Rainy        Cold      No     Mild  Normal   No
2  Morning   Sunny    Moderate     Yes   Normal  Normal  Yes
3  Evening   Sunny        Cold     Yes     High  Strong  Yes


 The attributes are:  [['Morning' 'Sunny' 'Warm' 'Yes' 'Mild' 'Strong']
 ['Evening' 'Rainy' 'Cold' 'No' 'Mild' 'Normal']
 ['Morning' 'Sunny' 'Moderate' 'Yes' 'Normal' 'Normal']
 ['Evening' 'Sunny' 'Cold' 'Yes' 'High' 'Strong']]

 The target is:  ['Yes' 'No' 'Yes' 'Yes']

 The final hypothesis is: ['?' 'Sunny' '?' 'Yes' '?' '?']
```

Looking at the data set, we have six attributes and a final attribute that defines the positive or negative example. In this case, yes is a positive example, which means the person will go for a walk.

So now, the general hypothesis is:

h_0 = {‘Morning’, ‘Sunny’, ‘Warm’, ‘Yes’, ‘Mild’, ‘Strong’}

This is our general hypothesis, and now we will consider each example one by one, but only the positive examples.

h_1= {‘Morning’, ‘Sunny’, ‘?’, ‘Yes’, ‘?’, ‘?’}

h_2 = {‘?’, ‘Sunny’, ‘?’, ‘Yes’, ‘?’, ‘?’}

We replaced all the different values in the general hypothesis to get a resultant hypothesis.

REFERENCES

1. geeksforgeeks.org
2. edureka.com

Dayananda Sagar College of Engineering

Department of Electronics and Communication Engineering

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru – 560 078.

(An Autonomous Institute affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment and Accreditation Council (NAAC) with 'A' grade

Assignment

Program: B.E.	Branch: ECE
Course: Machine Learning	Semester : 6
Course Code: 18EC6DECML	Date:02.07.2021

A Report on

Candidate Elimination Algorithm

Machine Learning Assignment

Submitted by

USN	NAME
1DS18EC143	Sameer Gautam

Faculty In-charge

Dr. K N Pushpalatha

Signature of Faculty In-charge

OVERVIEW

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

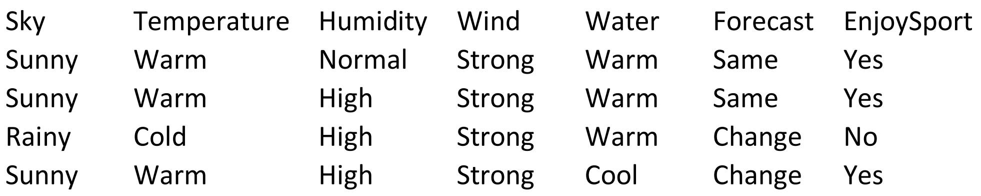
```
1. Load Data set
2. Initialize General Hypothesis  and Specific  Hypothesis.
3. For each training example
4. If example is positive example
       if attribute_value == hypothesis_value:
                 Do nothing
       else:
                 replace attribute value with '?' (Basically generalizing it)
5. If example is Negative example
           Make generalize hypothesis more specific.
```

PROGRAM

```
import csv
a = []
print("\n The Given Training Data Set \n")

with open(("F:\enjoysport.csv"), 'r') as csvFile:
    reader = csv.reader(csvFile)
    for row in reader:
        a.append (row)
        print(row)
num_attributes = len(a[0])-1 # we don't want last col which is target concet ( yes/no)
```

```
print("\n The initial value of hypothesis: ")
S = ['0'] * num_attributes
G = ['?'] * num_attributes
print ("\n The most specific hypothesis S0 : [0,0,0,0,0,0]\n")
print (" \n The most general hypothesis G0 : [?,?,?,?,?,?]\n")

for j in range(0,num_attributes):
       S[j] = a[0][j];



# Comparing with Remaining Training Examples of Given Data Set

print("\n Candidate Elimination algorithm \n")
temp=[]

for i in range(0,len(a)):
    if a[i][num_attributes]=='Yes':
        for j in range(0,num_attributes): #Specific hypothesis
            if a[i][j]!=S[j]:
                S[j]='?'

        for j in range(0,num_attributes):
            for k  in range(0,len(temp)):
                if temp[k][j] != '?' and temp[k][j] != S[j]:
                    del temp[k] #remove it if it's not matching with the specific hypothesis


        print(" For Training Example No :{0} the hypothesis is S{0}  ".format(i+1),S)
if (len(temp)==0):
            print(" For Training Example No :{0} the hypothesis is G{0} ".format(i+1),G)
        else:
            print(" For Training Example No :{0} the hypothesis is G{0}".format(i+1),temp)
    if a[i][num_attributes]=='No':
        for j in range(0,num_attributes):
            if S[j] != a[i][j] and S[j]!= '?':  #if not  matching with the specific Hypothesis take it seperately and store it
                G[j]=S[j]
                temp.append(G) # this is the version space to store all Hypotheses
                G = ['?'] * num_attributes

        print(" For Training Example No :{0} the hypothesis is S{0} ".format(i+1),S)
        print(" For Training Example No :{0} the hypothesis is G{0}".format(i+1),temp)
```

RESULTS



Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
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

```
Instances are:
[[‘sunny’ ‘warm’ ‘normal’ ‘strong’ ‘warm’ ‘same’]
[‘sunny’ ‘warm’ ‘high’ ‘strong’ ‘warm’ ‘same’]
[‘rainy’ ‘cold’ ‘high’ ‘strong’ ‘warm’ ‘change’]
[‘sunny’ ‘warm’ ‘high’ ‘strong’ ‘cool’ ‘change’]]

Target Values are: [‘yes’ ‘yes’ ‘no’ ‘yes’]

Initialization of specific_h and general_h
```

```
Specific Boundary: [‘sunny’ ‘warm’ ‘normal’ ‘strong’ ‘warm’ ‘same’]

Generic Boundary: [[‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’]]

Instance 1 is [‘sunny’ ‘warm’ ‘normal’ ‘strong’ ‘warm’ ‘same’] Instance is Positive

Specific Bundary after 1 Instance is [‘sunny’ ‘warm’ ‘normal’ ‘strong’ ‘warm’ ‘same’]

Generic Boundary after 1 Instance is [[‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’]]

Instance 2 is [‘sunny’ ‘warm’ ‘high’ ‘strong’ ‘warm’ ‘same’] Instance is Positive

Specific Bundary after 2 Instance is [‘sunny’ ‘warm’ ‘?’ ‘strong’ ‘warm’ ‘same’]

Generic Boundary after 2 Instance is [[‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’]]

Instance 3 is [‘rainy’ ‘cold’ ‘high’ ‘strong’ ‘warm’ ‘change’] Instance is Negative

Specific Bundary after 3 Instance is [‘sunny’ ‘warm’ ‘?’ ‘strong’ ‘warm’ ‘same’]

Generic Boundary after 3 Instance is [[‘sunny’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘warm’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘same’]]

Instance 4 is [‘sunny’ ‘warm’ ‘high’ ‘strong’ ‘cool’ ‘change’] Instance is Positive

Specific Bundary after 4 Instance is [‘sunny’ ‘warm’ ‘?’ ‘strong’ ‘?’ ‘?’]

Generic Boundary after 4 Instance is [[‘sunny’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘warm’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’]]
```

```
Final Specific_h: [‘sunny’ ‘warm’ ‘?’ ‘strong’ ‘?’ ‘?’]

Final General_h: [[‘sunny’, ‘?’, ‘?’, ‘?’, ‘?’, ‘?’], [‘?’, ‘warm’, ‘?’, ‘?’, ‘?’, ‘?’]]
```

REFERENCES

1. geeksforgeeks.org
2. vtupulse.com