

**Submitted in the partial fulfilment of Assignment 1 in Machine Learning BITS F464.**

**Submitted by**

<b>1.Sai Sanketh Vedula</b>	<b>-2013A7PS180H</b>
<b>2.Abhishek Kanniganti</b>	<b>-2013AAPS308H</b>
<b>3.Surya Teja Cheedella</b>	<b>-2013A3PS340H</b>
<b>4.PS Phani Teja</b>	<b>-2013AAPS102H</b>

## Assignment-1

This report contains the Implementation of Candidate elimination algorithm. The program is generic to work on any dataset.

The algorithm is implemented using C++ .The “Balloon” data set from UCI machine learning repository. The link for the given dataset is given below

<https://archive.ics.uci.edu/ml/datasets/Balloons>

The zip file contains the following files:

**Source code:** The source code includes the entire implementation of candidate elimination algorithm in C++

### Report

**Training Dataset-** Balloon dataset from UCI machine learning repository and the example dataset given in assignment 1 have been used as training datasets.

**Source code:**

**Input Format:**

The number of attributes and number of examples to be recorded are given as inputs. Last attribute is the target attribute which is either “Yes” or “No”/ “T” or “F” depending on the data.

**Test Data -1:**

6 4	(number of features , number of training examples)
3	(number of attributes for 1st feature)
Sunny	(names of the attributes)
Rainy	
Cloudy	
2	(number of attributes for 2nd feature)
Warm	(names of the attributes)

Cold  
 2 (number of attributes for 3rd feature)  
 Normal (names of the attributes)  
 High  
 2 (number of attributes for 4th feature)  
 Strong (names of features)  
 Weak  
 2 (number of attributes for 5th feature)  
 Warm (names of features)  
 Cool  
 2 (number of features for 6th feature)  
 Same (names of features)  
 Change  
 2 (number of target attributes)  
 Yes (names of attributes) Assumption: Always give positive target attribute as the first attribute  
 No  
 Sunny Warm Normal Strong Warm Same Yes (training examples)  
 Sunny Warm High Strong Warm Same Yes  
 Rainy Cold High Strong Warm Change No  
 Sunny Warm High Strong Cool Change Yes

## Test Data-2: ( Balloon data)

4 4 (number of features , number of training examples)  
 2 (number of attributes for 1st feature)  
 YELLOW (names of the features)  
 PURPLE  
 2 (number of attributes for 2nd feature)  
 SMALL (names of the attributes)  
 LARGE  
 2 (number of features for 3rd feature)  
 STRETCH (names of the attributes)  
 DIP  
 2 (number of features for 3rd feature)  
 ADULT (names of the attributes)  
 CHILD  
 2 (Target attributes)  
 T (names of target attributes) Assumption: Always give positive target attribute as the first attribute.  
 F  
 YELLOW SMALL STRETCH ADULT T  
 YELLOW SMALL STRETCH CHILD T  
 YELLOW SMALL DIP ADULT T  
 YELLOW SMALL DIP CHILD F

The input can be found in the file *data.txt*.

Copy-Paste the input for each case from the *data.txt* into the terminal.

Execute using:

```
$ g++ candidate_elimination.cpp
```

```
$ ./a.out
```

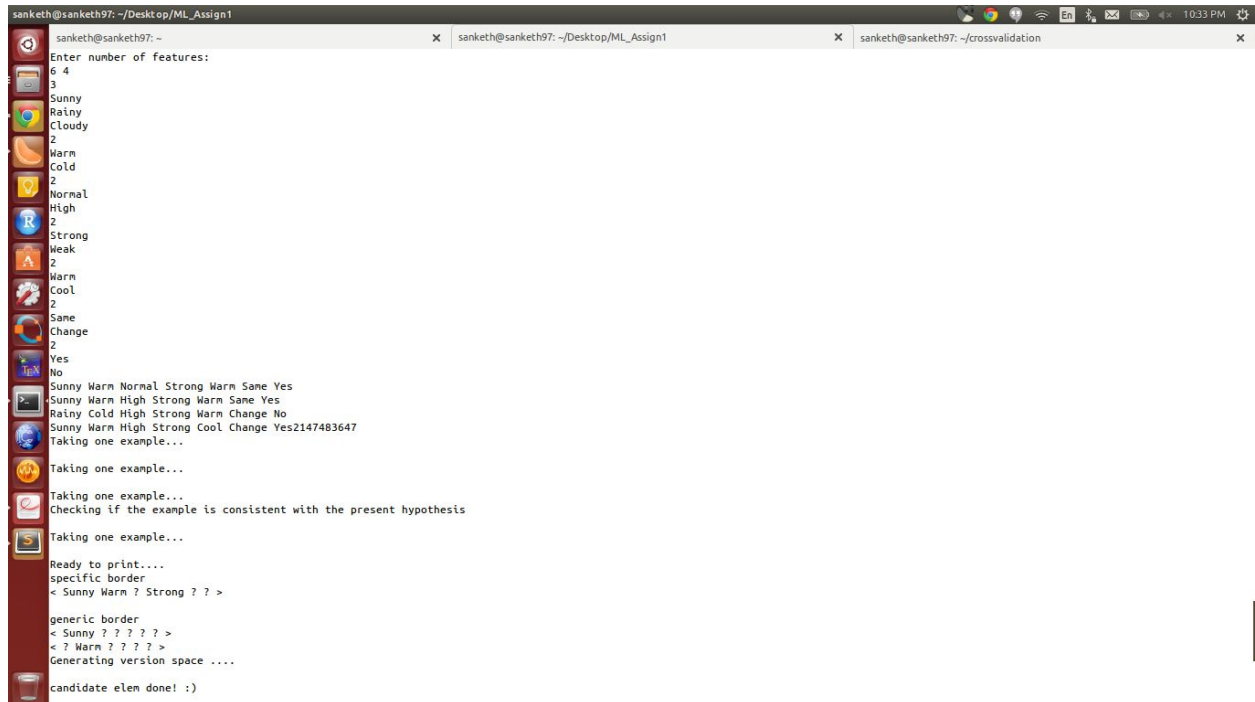
### Output Format:

The code implements candidate elimination and outputs most Specific hypothesis “S” and most Generic hypothesis “G”.

If S and G converge to single hypothesis, then it is shown as output. If they do not converge then “Inconsistent” is shown as output.

The following are the outputs for each test case.

### Test Code 1:



```
sanketh@sanketh97: ~/Desktop/ML_Assign1
Enter number of features:
6
4
3
Sunny
Rainy
Cloudy
2
Warm
Cold
2
Normal
High
2
Strong
Weak
2
Warm
Cool
2
Same
Change
2
Yes
No
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 Yes2147483647
Taking one example...
Taking one example...
Taking one example...
Checking if the example is consistent with the present hypothesis
Taking one example...
Ready to print....
specific border
< Sunny Warm ? Strong ? ? >
generic border
< Sunny ? ? ? ? ? >
< ? Warm ? ? ? ? ? >
Generating version space ....
candidate elem done! :)
```

## Test Code2:

```
sanketh@sanketh97: ~/Desktop/ML_Assign1
sanketh@sanketh97: ~
Taking one example...
Ready to print...
specific border
< Sunny Warm ? Strong ? ? >
generic border
< Sunny ? ? ? ? >
< ? Warm ? ? ? ? >
Generating version space ....
candidate elem done! :)
sanketh@sanketh97:~/Desktop/ML_Assign1$ g++ candidate_elimination.cpp
sanketh@sanketh97:~/Desktop/ML_Assign1$ ./a.out
Enter number of features:
4 4
2
YELLOW
PURPLE
2
SMALL
LARGE
2
STRETCH
DIP
2
ADULT
CHILD
2
T
F
YELLOW SMALL STRETCH ADULT T
YELLOW SMALL STRETCH CHILD T
YELLOW SMALL DIP ADULT T
YELLOW SMALL DIP CHILD F2147483647
Taking one example...
Taking one example...
Taking one example...
Checking if the example is consistent with the present hypothesis
Ready to print...
Inconsistent data..
sanketh@sanketh97:~/Desktop/ML_Assign1$
```

## Assumptions:

The negative training examples which are inconsistent with S having only one hypothesis are not considered.

The positive example's attribute is to be taken first. i.e the input order while entering the parameters should be first positive and second negative.