Exp 2 - Candidate Elimination Algorithm

February 15, 2022

The candidate-elimination procedure searches the hypothesis space in both directions. It keeps track of a set of most specific hypothesis, S, and a set of most generic hypothesis, G, that are both consistent with the training data. On the version space, these two sets define two boundaries.

A concept is a group of objects with a clear definition. For example, the term "bird" refers to any animal that is a bird and excludes any animal that is not a bird.

Each concept contains a definition that describes all of the concept's members in detail and does not apply to objects belonging to other concepts. As a result, **a concept can be defined as a boolean function defined over a set of all possible objects that returns true if and only if a given object belongs to the idea.**

1 Experimental Description

1.1 Objective

To implement the Candidate-Elimination algorithm on an appropiate dataset to get the version space and interpret it's output.

1.2 Algorithm

1. The algorithm identifies the general boundaries G and specific boundaries S of the version space V which is denoted by V(G,S). In V, the most broad theories are found in G, while the most specific ones are found in S.
2. All of the hypothesis in V can be obtained by specializing those from G until we reach S, or by generalizing those from S until we reach G.
3. G holds only the most general hypothesis in H at the start of Eliminiation. In the same way, H only contains the most particular hypothesis in S.
4. Candidate Elimination goes through each identified object one by one. It specializes G and generalizes S as needed, to ensure that the V(G, S) correctly classifies.

1.3 Procedure

- A General Hypothesis and a Specific Hypothesis is first created from the given dataset.
- In each case, if the model is a good example and value of query == hypothesis value:
  - There is nothing more to be done.
  - The adjective value is replaced with '?'
- If the model is a bad example:
  - Make normal clarity clearer.

1.4 System Requirements

Windows/Linux OS/Mac OS with R. Required packages are "**dplyr**", "**foreach**"

1.5 Dataset Summary

For this project, we have used Japanese economy car dataset. Attribute information of this dataset are as follows:

1. Country: USA, Japan.
2. Brand: Toyota, Chrysler, Honda.
3. Colour: Blue, Red, White, Green.
4. Decade: The decade it was manufactured.
5. Type: If its a economy or sports car.

2 Code and Results

```
rm(list = ls())
version$version.string
```

```
## [1] "R version 4.1.2 (2021-11-01)"
```

```
# Install the "foreach", "dplyr" packages by uncommenting and running the following command
#install.packages("foreach")
#install.packages("dplyr")

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(foreach)
```

```
# Categorizing data into Concepts and Target
# Concepts are the data on which Candidate elimination will be performed.
# Loading dataset into variable
data <- read.csv("EconomyCar.csv")
concepts <- select(data,c(1:5))
concepts[0]
```

```
## data frame with 0 columns and 7 rows
```

```
target <- select(data,6)
# Displaying the TARGET and CONCEPTS
print(target)
```

```
##   Eco
## 1  No
## 2 Yes
## 3  No
## 4 Yes
## 5  No
## 6 Yes
## 7  No
```

```r
print(concepts)
```

```
##   Country    Brand Colour Decade    Type
## 1  Japan    Toyota  Green   1970  Sports
## 2  Japan    Toyota  Blue    1990 Economy
## 3     USA Chrysler    Red   1980 Economy
## 4   Japan Chrysler   Blue   1970  Sports
## 5     USA    Honda    Red   1990 Economy
## 6  Japan     Honda  White   1980 Economy
## 7   Japan   Toyota  Green   1980  Sports
```

```r
# Writing the Candidate Elimination Function
# learn() function uses concepts and target as input and finds the general and specific hypothesis
learn <- function(concepts, target) {
    specific_h <- concepts[1,]
    print("Initialization of specific_h and general_h")
    print("specific_h ")
    print(specific_h)
    # Creating a 2D Array  of general hypothesis with length of specific hypothesis
    general_h <- array(dim =c(length(specific_h),length(specific_h)))
    for (i in 1:length(specific_h)){
        for (j in 1:length(specific_h)){
            general_h[i,j] <- "?"
        }
    }
    print("general_h:")
    print(general_h)
    print("concepts: ")
    print(concepts)
    for (i in 1:length(concepts[,1])){
        h<-slice(concepts,(1))   # Creating the target
        print(paste("Target is ",target[i,]))
        #Uncomment below line to test output of above commands
        #print(concepts[i,])
        if (target[i,] == "Yes"){     # Specifying the case where the model is a good example
            for (x in 1:length(specific_h)){
                print(paste("h[x] is:",h[x]))
                print(paste("specific_h[x] is:",specific_h[x]))
                if (h[x] != specific_h[x]){

                    specific_h[x,] <- "?"
                    general_h[x,x] <- "?"
                }
            }

        }
        if (target[i,] == "No"){     # Specifying the case where the model is a bad example
            for (x in 1:length(specific_h)){
                if (h[x] != specific_h[1,x]) # If current hypothesis is not same as the specific hypoth
                  {
                    print(paste("h[x] is:",h[x]))
                    print(paste("specific_h[x] is:",specific_h[1,x]))
                    general_h[x,x] <- specific_h[x]
                }
                else{
                   general_h[x,x] <- concepts[i,x]
                }
            }

        }
        print(paste("\n Steps of Candidate Elimination Algorithm: ",i+1))
        print(paste("Specific_h: ",i+1))
        print(specific_h)
        print(paste("general_h :", i+1))
        print(general_h)

    }
}
```

```
learn(concepts, target)
```

```
## [1] "Initialization of specific_h and general_h"
## [1] "specific_h "
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h:"
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "?"  "?"  "?"  "?"  "?"
## [2,] "?"  "?"  "?"  "?"  "?"
## [3,] "?"  "?"  "?"  "?"  "?"
## [4,] "?"  "?"  "?"  "?"  "?"
## [5,] "?"  "?"  "?"  "?"  "?"
## [1] "concepts: "
##   Country    Brand Colour Decade    Type
## 1  Japan    Toyota  Green   1970  Sports
## 2  Japan    Toyota  Blue    1990 Economy
## 3     USA Chrysler    Red   1980 Economy
## 4   Japan Chrysler   Blue   1970  Sports
## 5     USA    Honda    Red   1990 Economy
## 6  Japan      Honda  White   1980 Economy
## 7   Japan   Toyota  Green   1980  Sports
## [1] "Target is  No"
## [1] "\n Steps of Candidate Elimination Algorithm:  2"
## [1] "Specific_h:  2"
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h : 2"
##      [,1]     [,2]     [,3]    [,4]   [,5]
## [1,] "Japan " "?"      "?"     "?"    "?"
## [2,] "?"      "Toyota" "?"     "?"    "?"
## [3,] "?"      "?"      "Green" "?"    "?"
## [4,] "?"      "?"      "?"     "1970" "?"
## [5,] "?"      "?"      "?"     "?"    "Sports"
## [1] "Target is  Yes"
## [1] "h[x] is: Japan "
## [1] "specific_h[x] is: Japan "
## [1] "h[x] is: Toyota"
## [1] "specific_h[x] is: Toyota"
## [1] "h[x] is: Green"
## [1] "specific_h[x] is: Green"
## [1] "h[x] is: 1970"
## [1] "specific_h[x] is: 1970"
## [1] "h[x] is: Sports"
## [1] "specific_h[x] is: Sports"
## [1] "\n Steps of Candidate Elimination Algorithm:  3"
## [1] "Specific_h:  3"
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h : 3"
##      [,1]     [,2]     [,3]    [,4]   [,5]
## [1,] "Japan " "?"      "?"     "?"    "?"
## [2,] "?"      "Toyota" "?"     "?"    "?"
## [3,] "?"      "?"      "Green" "?"    "?"
## [4,] "?"      "?"      "?"     "1970" "?"
## [5,] "?"      "?"      "?"     "?"    "Sports"
## [1] "Target is  No"
## [1] "\n Steps of Candidate Elimination Algorithm:  4"
## [1] "Specific_h:  4"
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h : 4"
##      [,1]  [,2]       [,3]  [,4]   [,5]
## [1,] "USA" "?"        "?"   "?"    "?"
## [2,] "?"   "Chrysler" "?"   "?"    "?"
## [3,] "?"   "?"        "Red" "?"    "?"
## [4,] "?"   "?"        "?"   "1980" "?"
## [5,] "?"   "?"        "?"   "?"    "Economy"
## [1] "Target is  Yes"
## [1] "h[x] is: Japan "
## [1] "specific_h[x] is: Japan "
## [1] "h[x] is: Toyota"
## [1] "specific_h[x] is: Toyota"
## [1] "h[x] is: Green"
## [1] "specific_h[x] is: Green"
## [1] "h[x] is: 1970"
## [1] "specific_h[x] is: 1970"
## [1] "h[x] is: Sports"
## [1] "specific_h[x] is: Sports"
## [1] "\n Steps of Candidate Elimination Algorithm:  5"
## [1] "Specific_h:  5"
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h : 5"
##      [,1]  [,2]       [,3]  [,4]   [,5]
## [1,] "USA" "?"        "?"   "?"    "?"
## [2,] "?"   "Chrysler" "?"   "?"    "?"
## [3,] "?"   "?"        "Red" "?"    "?"
## [4,] "?"   "?"        "?"   "1980" "?"
## [5,] "?"   "?"        "?"   "?"    "Economy"
## [1] "Target is  No"
## [1] "\n Steps of Candidate Elimination Algorithm:  6"
## [1] "Specific_h:  6"
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h : 6"
##      [,1]  [,2]    [,3]  [,4]   [,5]
## [1,] "USA" "?"     "?"   "?"    "?"
## [2,] "?"   "Honda" "?"   "?"    "?"
## [3,] "?"   "?"     "Red" "?"    "?"
## [4,] "?"   "?"     "?"   "1990" "?"
## [5,] "?"   "?"     "?"   "?"    "Economy"
## [1] "Target is  Yes"
## [1] "h[x] is: Japan "
## [1] "specific_h[x] is: Japan "
## [1] "h[x] is: Toyota"
## [1] "specific_h[x] is: Toyota"
## [1] "h[x] is: Green"
## [1] "specific_h[x] is: Green"
## [1] "h[x] is: 1970"
## [1] "specific_h[x] is: 1970"
## [1] "h[x] is: Sports"
## [1] "specific_h[x] is: Sports"
## [1] "\n Steps of Candidate Elimination Algorithm:  7"
## [1] "Specific_h:  7"
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h : 7"
##      [,1]  [,2]    [,3]  [,4]   [,5]
## [1,] "USA" "?"     "?"   "?"    "?"
## [2,] "?"   "Honda" "?"   "?"    "?"
## [3,] "?"   "?"     "Red" "?"    "?"
## [4,] "?"   "?"     "?"   "1990" "?"
## [5,] "?"   "?"     "?"   "?"    "Economy"
## [1] "Target is  No"
## [1] "\n Steps of Candidate Elimination Algorithm:  8"
## [1] "Specific_h:  8"
##   Country  Brand Colour Decade   Type
## 1  Japan  Toyota  Green   1970 Sports
## [1] "general_h : 8"
##      [,1]    [,2]     [,3]    [,4]   [,5]
## [1,] "Japan" "?"      "?"     "?"    "?"
## [2,] "?"     "Toyota" "?"     "?"    "?"
## [3,] "?"     "?"      "Green" "?"    "?"
## [4,] "?"     "?"      "?"     "1980" "?"
## [5,] "?"     "?"      "?"     "?"    "Sports"
```

From the output, the following can be interpreted:

- The algorithm works best when the data are compatible, i.e. there are no conflicting examples. It is very important to find the right place to find failure.

- The size of the translation space usually depends on the distance between the limits and also depends on the number of good examples to learn from. For example, Class 5 has very few good examples. As a result the S-border is very clear with fewer question marks, while the G-border is larger.

3 Conclusion

Candidate elimination Algorithm got implemented successfully over the given dataset.