Entropy Based Reduct (EBR) Computation Using Sequential Backward Elimination(SBE) Technique

Submitted By:-
Parwat Singh Anjana
14MCMT34
M.Tech(CS)

Introduction:-

In this reduct computation technique we compute the Reduct of the given Decision System based on the entropy calculation, developed from work carried out in Jensen and Shen (2004a).EBR is concerned with examining a dataset and determining those attributes that provide the most gain in information. The entropy of attribute A (which can take values a 1 ,...,a m) with respect to the conclusion C (of possible values c 1 ,...,c n) is defined as: follows :-

$$H(C \mid A) = -\sum_{j=1}^{m} p(a_j) \sum_{i=1}^{n} p(c_i \mid a_j) \log_2 p(c_i \mid a_j)$$

Where:-
m= is number of unique objects with respect to selected set of attribute, and
n=number of unique decision with respect to decision attribute

Assumptions:-

In our implementation we assume that

1. Last Attribute of given D.S. is decision attribute
2. System is consistent

Language Used:-

R Language

Function Implemented:-

Two functions are implemented named as
1.reduct()
2.entropy()

1.reduct():-

This function compute the reduct set based on the Sequential Backward Elimination.
Initially we consider **entire set of conditional attribute as reduct set** and compute the **entropy** for that set with respect to decision attribute by calling **entropy()** function. We called this entropy value as **“oldentrp”** in our reduct function, and if **“oldentrp>0”** we simply print that “unable to find out reduct because initial

entropy is greater then 0”.
when “**oldentrp==0”** then we enter into the loop to perform actual reduct computation where we first compute the **entropy for each individual attribute**(to decide which attribute to be selected first to remove from initial reduct set i.e from all conditional set). and store it into array of attrbiute where index represent attribute **ent_redAtr[i].**
We remove the attribute having higest entropy value among all conditional attribute and then we again find out the entropy of **new reduct set(T1) .**
if “newEntopy==0” then we continue this process (repeat) until we have number of **conditional attribute greater then or equal to 1**(total attribute 2 ,one conditional and one decission) or **“newEntopy==0” .**
in each iteration at the end if that attribute is successful removed from reduct set we make that attribute entropy equal to 0 in **ent_redAtr[i]** so that we can select next attribute having highest attribute from remaining.

when **“total attribute(tA)<2”** or **“newEntopy>0”** we break the loop and come out side the loop.

entropy.R × reductComp.R ×

Source on Save

```
reduct=function()
{
  dectab=read.csv("input.csv")   #file name to read data
  print("Decision System :")
  print(dectab)
  tA=ncol(dectab) #total number of attributes
  cA=tA-1         #conditioanl attribute
  oldentrp=entropy(tA,dectab) #calculate for entire conditional Atri Set
  print("Entropy Of All Cond Attrbiute Set:")
  print(oldentrp)
  reduct_set=c(1:cA)  #initial reduct set
  ent_redAtr=c(1:cA)  #to hold value of individual Cond attribute
  if(oldentrp==0) #DO tis only if initial entropy of all attribute is equal to 0
  {
    for(i in 1:cA)
    {
      reduct_set[i]=i #INITALIZATION OF INITIAL REDUCT SET
      ent_redAtr[i]=entropy(2,dectab[,c(i,tA)]) #Compute Entropy Of Individual Cond. Attribute
    }
    print("Entropy Of Individual Attribute::")
    print(ent_redAtr)
    print("Initial Reduct Set::")
    print(reduct_set)
    repeat #REDUCT COMPUTAION ALGORITHEM IMPLEMENTATION
    {
      val=which.max(ent_redAtr)
      print("Attribute Selected to remove::")
      print(val)
      T1=setdiff(reduct_set, val) #set excluding selected attribute
      tA1=length(T1)+1
      T2=union(T1,tA)
      dectab1=dectab[,T2]
      if(tA1>=2)
      {
        print("New Decision System::")
        print(cbind(dectab1))
        newEntopy=entropy(tA1,dectab1) #COMPUT ENTROPY OF NEW REDUCT SET
```

```
        newEntopy=entropy(tA1,dectab1) #COMPUT ENTROPY OF NEW REDUCT SET
        print("Entropy::")
        print(newEntopy)
        if(newEntopy==0)
        {
          reduct_set=T1
          print("New Reduct Is:")
          print(reduct_set)
          ent_redAtr[val]=0   #change the entropy of selected attribute to 0
        }
        else
        {
          print("Imp Attribute for Reduct Can't Remove::")
          print(val)
          T1=union(reduct_set, val)  #add that attribute back to reduct Set
          break
        }
      }
      else
      {
        print("Can't Remove This Attribute,Only 1 Cond Attribute left::")
        T1=union(reduct_set, val)
        break #when only one conditional attribute left
      }
      reduct_set=T1
    }
    reduct_set=T1
    print("Final Reduct Is:")
    print(reduct_set)
  }
  else # when all cond attribute set entropy >0
  {
    print("No reduct for This Dec System,Initial Entropy of All Cond.Attribute >0::")
    print(oldentrp)
  }
}
```

Note:-
finding out the entropy of individual attribute and removing highest entropy attribute from reduct set saves lot of computation time.

2.entropy() :-

Take two arguments **1st total number of attributes(tA1)** and **2nd dec system(dectab1)** for which we want to compute the entropy.

In this function **table1= cbind(dsctab[,-dA])** is use to convert the vacter into row wise when onle single attribute left in condtional attribute set and -dA is used to exclude the decsion attribute from D.S to create onlt conditioanl attribute sysetm.

“uniquetable” is conditional attribute table with unique object entry for this **unique(table1)** function is used.

“val1” or (aj) one object of unique cond attribute table i.e object of **“uniquetable”. “val2”** object of normal conditioanl attribute table. Then compare val1 and val2 to find out how many time val1 repeated in cond attribute table to find out probability of (aj). And increase the counter **(count)** if both are same.

```
entropy=function(dA,dsctab)
{
  cA=dA-1 #number of conditional attribute
  table1=cbind(dsctab[,-dA]) #to convert vactor into row
  uniqtable=unique(table1)  #Unique Object table of cond System
  m=0 # counter
  count2=0 #to find out probability of each element in 2nd fun
  manprob=0
  entp1=0
  for(n in 1:nrow(uniqtable))
  {
    count=0
    val1=uniqtable[n,] #selcet each unique object
    for(y in 1:nrow(table1))#loop to find out probability of individual unique object(p(aj))
    {
      val2=table1[y,]
      if(all(val1==table1[y,]))# compare val1 with each object of cond table
      {
        count=count+1 #number of time val1 object repeate
      }
    }
    unique_ds=unique(dsctab[,dA]) #find out total number of decision
    sum1=0
    for(y in 1:length(unique_ds))  #to find out all values p(Ci/aj)logp(Ci/aj) for each unique aj
    {
      count2=0
      val3=unique_ds[y] # select one decision
      unio=merge(val1,val3) # combine dec value with selected unique object
      for(m in 1:nrow(dsctab))
      {
        if(all(dsctab[m,]==unio))
        {
          count2=count2+1 #number of time that unique object with decision repeate in dec table
        }
      }
     if(count2!=0)
     {
       p2=(count2/count)*(log(count2/count)/log(2))#find prob value for that object
       sum1=sum1+p2 # add to privious dec attribute value
     }
    }
    entp1=entp1+(-((count/nrow(table1))*sum1)) #final entropy
  }
  return(entp1)
}
```

And then we find out **Sumation of P(ci/aj)log(p(ci/aj)).**
Where **“unique_ds”** is vactor of unique decison attribute.
“sum1” is used to hold the sum of all values from **(1 to m) for each decision** w.r.t. **aj**
“val3” is selected **unique object(aj)**+ selected **Decision**
and for each val3 we find out number of time that will occure in decision system and increas the counter if match is found.

then we compute p2{**Sumation of P(ci/aj)log(p(ci/aj)).**}
and add it to sum1.
Then add this sum1 value to the variable **entp1** which is used to store the **entropy value**
repeate this for all unique object of cond attribute table and finaly return the entropy value(entp1).

Results:-
Following is the results of the given decision system in the paper **(Chapter 3 Rough Set -Based Feature Selection:A review by Richard Jensen)**

In our example dataset, EBR first evaluates the entropy of initial reduct set {A1,A2,A3,A4},

Entropy
{A1,A2,A3,A4}= 0

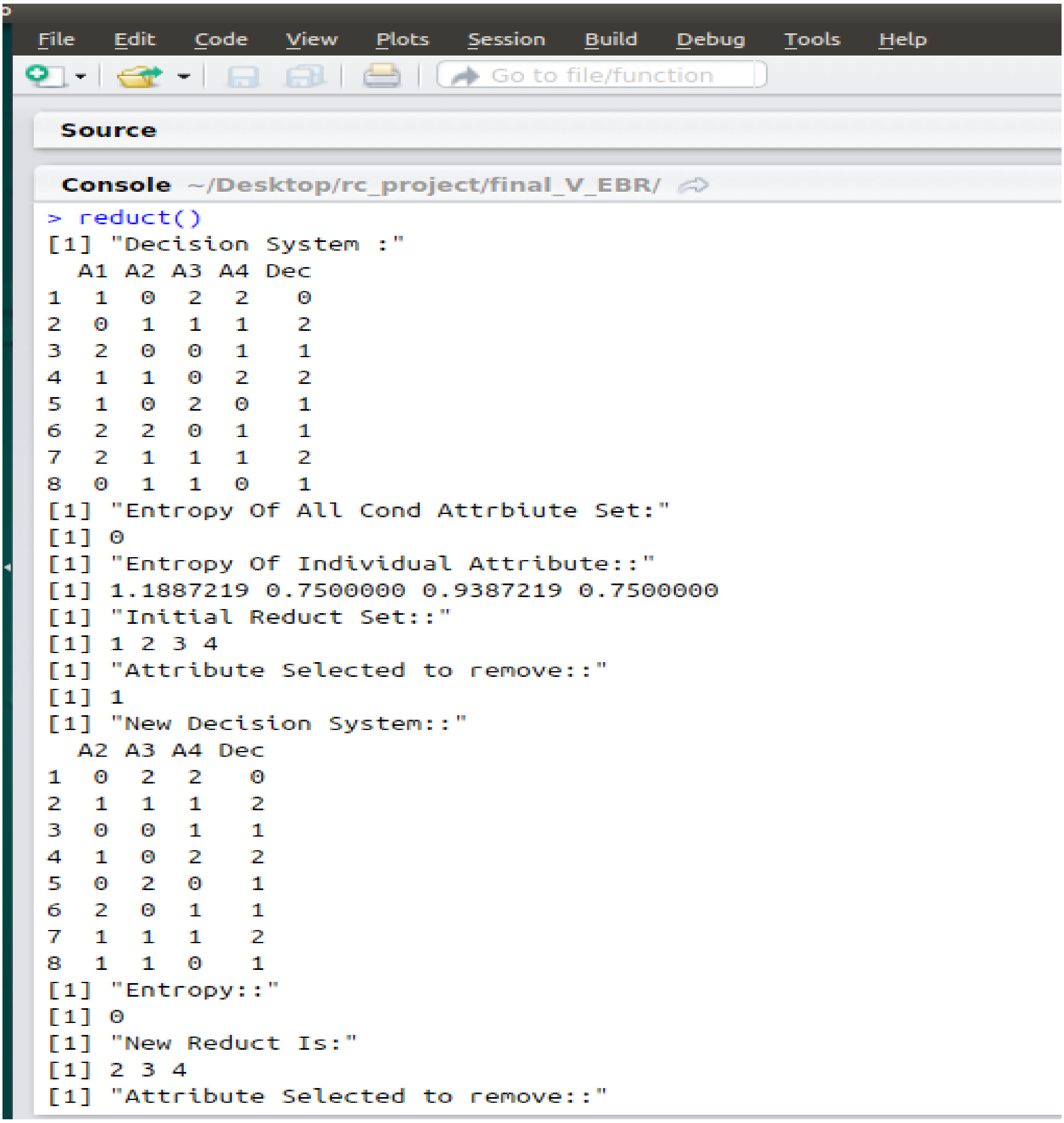
if initial reduct set entropy==0, Then calculate entropy for each individual attribute:
Entropy
{A1} 1.1887219
{A2} 0.75
{A3} 0.9387219
{A4} 0.75

The Attribute with higest entropy here is {A1}. The algorithm selects attribute {A1} to remove from Initial reduct set {A1,A2,A3,A4}, So set become{A2,A3,A4} then calculate entropy for this subset:
Entropy
{A2,A3,A4}=0
Hence new reduct become {A2,A3,A4}.

Again select the attribute (from {A2,A3,A4}) having higest entropy and that is {A3}, The algorithm selects attribute {A3} to remove from current reduct set{A2,A3,A4}, So subset become{A2,A4} then calculate entropy for this subset:
Entropy
{A2,A4}=0
Hence new reduct become {A2,A4}.

Again select the attribute (from {A2,A4}) having higest entropy Here both A2 and A4 have same entropy 0.75, So algorithm selects attribute {A2} to remove from current reduct set{A2,A4}, So subset become{A4} then calculate entropy for this subset:
Entropy
{A4}>0

Stopping criterion has been met, add A2 back to the reduct set and return final Reduct as {A2,A4} as this value equals the entropy for the entire feature set (H(D|{A2,A4}) = 0 = H(D|C)). The algorithm terminates and returns this feature subset; the dataset can now be reduced to these features only. As the resulting entropy is zero, the returned subset is a rough set reduct.




```
> reduct()
[1] "Decision System :"
  A1 A2 A3 A4 Dec
1  1  0  2  2   0
2  0  1  1  1   2
3  2  0  0  1   1
4  1  1  0  2   2
5  1  0  2  0   1
6  2  2  0  1   1
7  2  1  1  1   2
8  0  1  1  0   1
[1] "Entropy Of All Cond Attrbiute Set:"
[1] 0
[1] "Entropy Of Individual Attribute::"
[1] 1.1887219 0.7500000 0.9387219 0.7500000
[1] "Initial Reduct Set::"
[1] 1 2 3 4
[1] "Attribute Selected to remove::"
[1] 1
[1] "New Decision System::"
  A2 A3 A4 Dec
1  0  2  2   0
2  1  1  1   2
3  0  0  1   1
4  1  0  2   2
5  0  2  0   1
6  2  0  1   1
7  1  1  1   2
8  1  1  0   1
[1] "Entropy::"
[1] 0
[1] "New Reduct Is:"
[1] 2 3 4
[1] "Attribute Selected to remove::"
```

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function

Source

Console ~/Desktop/rc_project/final_V_EBR/

```
[1] "New Reduct Is:"
[1] 2 3 4
[1] "Attribute Selected to remove::"
[1] 3
[1] "New Decision System::"
  A2 A4 Dec
1  0  2   0
2  1  1   2
3  0  1   1
4  1  2   2
5  0  0   1
6  2  1   1
7  1  1   2
8  1  0   1
[1] "Entropy::"
[1] 0
[1] "New Reduct Is:"
[1] 2 4
[1] "Attribute Selected to remove::"
[1] 2
[1] "New Decision System::"
  A4 Dec
1  2   0
2  1   2
3  1   1
4  2   2
5  0   1
6  1   1
7  1   2
8  0   1
[1] "Entropy::"
[1] 0.75
[1] "Imp Attribute for Reduct Can't Remove::"
[1] 2
[1] "Final Reduct Is:"
[1] 2 4
> 
```