IRIM: Interesting Rule Induction Module with Handling Missing Attributes Values

By

Debabrata Majhi

Overview

- Introduction
- IRIM: Interesting Rule Induction Module
- Handling Missing Attribute Values
- Project Architecture and Algorithm
- Analysis
- Future Work
- Conclusion

Introduction

- Aim of the project:
  - Implementation of IRIM
  - Handling different types of missing data
  - Inducing rules for dataset like IRIS

Data Mining

- Data Mining: It is the process of discovering interesting patterns from large amount of data, that are:
  - Valid: The patterns hold in general.
  - Novel: We did not know the pattern beforehand.
  - Useful: We can devise actions from the patterns.
  - Understandable: We can interpret and comprehend the patterns.

Why Data Mining?

Broadly, the data mining could be used to answer following:

- Forecasting
- Classification
- Association
- Clustering
- Anomaly detection

Rule Induction

- **Rule Induction:** It is a special area of data mining and machine learning in which we extract useful if-then rules from data based on statistical significance.

- The general expression of rule is [1]:

If (attribute_1, value_1) & (attribute_2, value_2) and ……

(attribute_n, value_n) → (decision, value)

IRIM: Interesting Rule Induction Module

- Background:
  - Introduced by Dr. Jerzy Grzymala-Busse, Jay Hamilton, and Dr. Zdzislaw Hippe in 2004 [4]
  - A component of LERS (Learning from Examples based on Rough Sets)

IRIM: Features

- Rules induced by IRIM are interesting and able to capture the hidden relationship
- Resembles the ALL RULES algorithm [4] and the EXPLORE algorithm [3]
- It induces the strongest possible rules which cover most of the concept

IRIM: Quality Assurance

- Quality Assurance: Ensures that the induced rules are strongest rule describing the concept
  - Minimum of conditional probability of a given concept also known as ratio parameter

$$\text{Conditional Probability} = \frac{Strength\ of\ the\ Rule}{Coverage\ of\ the\ Rule\ (LHS)}$$

IRIM: Specification

- Find all rules for the dataset, it will inversely affects its performance
- May be user doesn’t need all rules
- Certain rule specifications are introduced
  - Minimum length
  - Maximum length
  - Minimum strength

IRIM: Rule Induction

- To induce rules by IRIM, we need a training set where the decision values have already been classified by an expert.

- The rule induction process of IRIM can be divided into 3 major steps:
  - Attribute value pairs
  - Set of all possible conditions
  - Inducing rules

IRIM with Sample Example

Table 1: Sample dataset with Numerical and discrete attribute

Case No	Age	HyperTension	Complication	Delivery
1	26	no	none	preterm
2	26	yes	obesity	preterm
3	34	no	obesity	fullterm
4	34	yes	obesity	fullterm
5	28	yes	obesity	fullterm
6	28	yes	none	preterm

$$[(\text{Delivery, preterm})] = \{1, 2, 6\}$$
$$[(\text{Delivery, fullterm})] = \{3, 4, 5\}$$

IRIM: Rule Induction (Step 1)

- IRIM computes the set of blocks for all attribute value pair
  - Attribute Value: (a, v)

- In the above table 1, HyperTension might be the attribute and the value ‘Yes’ might be on of its value
  - Attribute value pair will be (HyperTension,Yes)

IRIM: Rule Induction (Step 1)

- Block for a given Attribute Value Pair: Set of all the training cases described by the attribute value pair.
  - [(HyperTension,Yes)] ={2, 4, 5, 6}
- Representation of blocks for Attribute Value Pair:

[(a, v)] = {x| a(x) = v} where

x is case in our training dataset

a is the attribute

v is value for the given attribute ‘a’

a(x) is the value of the attribute at case number ‘x’

IRIM: Rule Induction (Step 1)

- Numerical Attribute:
  - IRIM sorts all the values of the attribute, then it computes the cut-points as the average of consecutive values.
  - Then creates two blocks for each cutpoints, one consists of all the cases where the value is smaller than the cutpoint value and the other block with cases greater than or equal to the cutpoint value

[(Age, <27)] ={1, 2}

[(Age, >=27)] = {3, 4, 5, 6}

IRIM with Sample Example

Table 1: Sample dataset with Numerical and discrete attribute

Case No	Age	HyperTension	Complication	Delivery
1	26	no	none	preterm
2	26	yes	obesity	preterm
3	34	no	obesity	fullterm
4	34	yes	obesity	fullterm
5	28	yes	obesity	fullterm
6	28	yes	none	preterm

$$[(\text{Delivery, preterm})] = \{1, 2, 6\}$$
$$[(\text{Delivery, fullterm})] = \{3, 4, 5\}$$

IRIM: Step 1

- IRIM will identify distinct values in ‘Age’ column as numerical and sort them in order:-

  [26,28,34]

- The first cut-point will be 27 (i.e. (26+28)/2)
- The attribute value pair will be:

  (Age, < 27) and (Age, >=27)

IRIM: Step 1 (Cont.)

- Attribute value pair for the ‘Age’ attribute:

  [(Age, < 27)] = {1, 2}

  [(Age, >= 27)] = {3, 4, 5, 6}

  [(Age, < 31)] = {1, 2, 5,6}

  [(Age, >= 31)] = {3, 4}

IRIM: Step 1 (Cont.)

- Attribute value pair for the HyperTension attribute is:
  - [(HyperTension, no)] = {1, 3}
  - [(HyperTension, yes)] = {2, 4, 5, 6}

- Attribute value pair for the Complication attribute is:
  - [(Complication, none)] = {1, 6}
  - [(Complication, obesity)] = {2, 3,4, 5}

IRIM: Step 2

- **Set of all possible conditions:**
- IRIM takes the combination of all possible set of conditions

  – Checks for the min and max rule length condition specified by the user
  – Check minimum coverage requirement

IRIM: Step 2 (Cont.)

- There are
  - 4 distinct values for Attribute Age
  - 2 distinct values for Attribute HyperTension
  - 2 distinct values for Attribute Complication
-
- Because we chose Min Rule length = 2 and Max Rule length = 3:

  (4*2) + (4*2) + (2*2) + (4*2*2) = 52 combination possible

IRIM: Step 2 (Cont.)

Few of the possible rules identified in step 2, such that Coverage is greater than 2

1. [(Age, > 27)] & [(Complication, obesity)] = {3, 4, 5},
2. [(Age, > 31)] & [(Complication, obesity)] = {3, 4},
3. [(Age, < 31)] & [(Complication, none)] = {1, 6},
4. [(Age, > 27)] & [(HyperTension, yes)] = {4, 5, 6},
5. [(Age, < 31)] & [(HyperTension, yes)] = {2, 5, 6},
6. [(HyperTension, yes)] & [(Complication, obesity)] = {2, 4, 5},
7. [(Age, < 31)] & [(Complication, obesity)] = {2, 5},
8. [(Age, > 27)] & [(HyperTension, yes)] & [(Complication, obesity)] = {4, 5},
9. [(Age, < 31)] & [(HyperTension, yes)] & [(Complication, obesity)] = {2, 5}.

IRIM: Step 3

- **Inducing rules:**
  - IRIM induced rules from the list of all possible rules
  - Assigns the induced rules to the respective decision value by intersecting the block of the induced rule to that of decision block

$$[(a_1, v_1)] \cap [(a_2, v_2)] \cap \ldots. \cap [(a_n, v_n)] \subseteq [(d, d_i)]$$

  - Decision Block: $[d, d_i]$ is the concept or set of all cases in the dataset where the value of the decision is d_i

IRIM: Step 3 (Cont.)

- IRIM computes the conditional probability for the rules

$$\text{Conditional Probability} = \frac{Strength\ of\ the\ Rule}{Coverage\ of\ the\ Rule\ (LHS)}$$

- Store the rule if conditional probability is greater than the min Conditional Probability

IRIM: Step 3 (Cont.)

- Rule can be classified on the basis of:
  - Strength: The total number of cases correctly classified by the rule during training
  - Specificity: The total number of attribute-value pairs on the left-hand side of the rule
  - Support: The sum of scores of all matching rules from the concept, where the score of the rule is the product of its strength and specificity

IRIM: Step 3 (Cont.)

- (Age, > 27) & (HyperTension, yes) → (Decision, fullterm) = {4, 5},
  Strength =2,
  Specificity =2,
  Support =4,
  Conditional Probability =0.66.

- (Age, > 27) & (HyperTension, yes) → (Decision, preterm) ={6} (Discarded)

IRIM: Step 3 (Cont.)

- Induced Rules
  - (Age, > 27) & (Complication, obesity) → (Decision, fullterm)

    Strength =3,
    Specificity =2,
    Support =6,
    Conditional Probability =1.0.

  - (Age, > 31) & (Complication, obesity) → (Decision, fullterm)
  - (Age, < 31) & (Complication, none) → (Decision, preterm)

Handling Missing Attribute Values

- Method to handle missing data:
  - Sequential (preprocessing the dataset to handle the missing values)
  - Parallel (the missing values are considered during the main process of acquiring knowledge)

Handling Missing Attribute Values (Cont.)

- Different types of sequential method to handle missing values:
  - deleting case with missing value
  - replacing missing value with most common attribute value
  - assuming all possible values of the attribute for the missing value
  - replacing missing values by the mean of the numerical attribute
  - assigning missing value attribute from as closest case in the dataset

Missing Value denoted by “*”,”?” and “-”

- Missing value “*”: Replacing with all possible known values for the attribute
- Missing values “?”: Completely delete the missing values
- Missing values “- “: Replacing with all possible known values for the attribute restricted by concept

Example to Handle missing value(cont.)

Case No	Age	HyperTension	Complication	Delivery
1	26	no	none	preterm
2	26	-	obesity	preterm
3	34	no	obesity	fullterm
4	34	yes	obesity	fullterm
5	28	yes	*	fullterm
6	?	yes	none	preterm

Example to Handle missing value

- [(Age, < 27)] = {1, 2}
- [(Age, >= 27)] = {3, 4, 5}
- [(Age, < 31)] = {1, 2, 5}
- [(Age, >= 31)] = {3, 4}
- [(HyperTension, no)] = {1, 3}
- [(HyperTension, yes)] = {4, 5, 6}
- [(Complication, none)] = {1, 6}
- [(Complication, obesity)] = {2, 3,4}

[(Age,?)] = {6}
[(HyperTension, -)] = {2}
[(Complication, *)] = {5}

Example to Handle missing value(cont.)

- **Handling the missing value with '?':** Remove the case completely
- **Handling the missing value with '*':**
  - **Possible values:** none and obesity
  - [(Complication, none)] = {1, 5, 6}
  - [(Complication, obesity)] = {2, 3, 4, 5}

Example to Handle missing value(cont.)

- **Handling the missing value with '-':**
  - **Possible value restricted to concept**
  - For case number 2, we have (Delivery, preterm), covers case 1, 2 and 6
  - **Possible values:** "yes" and "no"
    - [(HyperTension, no)] = {1, 2, 3}
    - [(HyperTension, yes)] = {2, 4, 5, 6}

Example to Handle missing value(cont.)

- Rules Generated:
  - (Age, > 27) & (Complication, obesity) → (Decision, fullterm)
  - (Age, > 27) & (HyperTension, yes) → (Decision, fullterm)
  - (Age, < 27) & (HyperTension, no) → (Decision, preterm)
  - (Age, < 31) & (HyperTension, no) → (Decision, preterm)

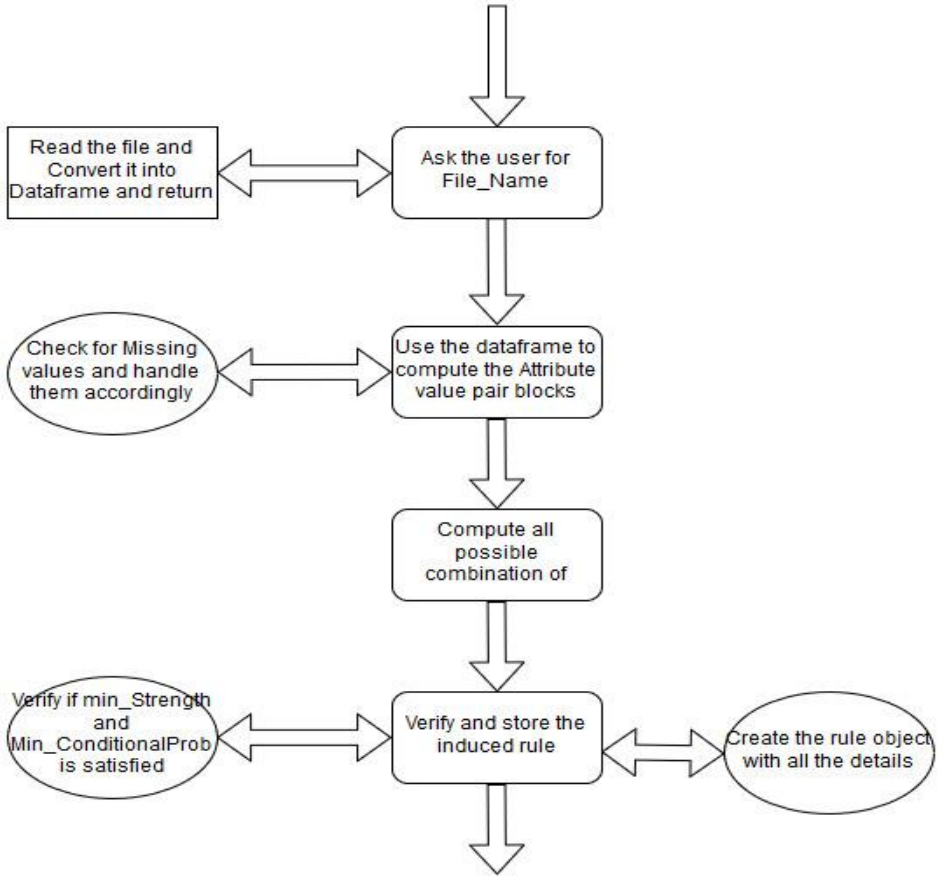
Project Architecture and Algorithm

- The project has been implemented using Python and pySpark library
  - **Class FileReader:** The main functionality of the class is to read the file and return the pySpark dataframe back to the RuleManager class
  - **Class RuleClass:** The purpose of the class is to store the detail about the rules like, Strength, Conditional probability, Coverage and Decision Cover.
  - **Class RuleManager:** All the actual rule induction and processing happens in the RuleManager class.

PySpark library

- Advantage of Pyspark:
  - Highly customized for big data
  - Structured Query

Flow of code


Read the file and Convert it into Dataframe and return
Ask the user for File_Name
Check for Missing values and handle them accordingly
Use the dataframe to compute the Attribute value pair blocks
Compute all possible combination of
Verify if min_Strength and Min_ConditionalProb is satisfied
Verify and store the induced rule
Create the rule object with all the details

Algorithm

```
Start IRIM
data_set = read_file(filename)                  //create a data set by reading the content from file
py_dataset=convertTableToSparkDataframe()       // Convert the dataset into pySpark Dataframe

//Creating Blocks for the complete dataset
foreach attribute in py_dataset(Attributes):
        datasetBlocks =createBlock()            //Creating Attribute value blocks, Dictionary for blocks
        data_type(attribute)                    //Store datatype for each attribute (Numerical or discreate)

//Handling Missing data
foreach attribute in py_dataset(Attributes):
        if '*' in datasetBlocks(attribute):
                //Assign all possible attribute value to the missing attribute
        if '?' in datasetBlocks(attribute):
                //delete and ignore the case completely
        if '-' in datasetBlocks(attribute):
                // Assign all possible attribute value to the missing attribute restricted to the concept
```

Algorithm (Cont.)

```
//Converting Numerical data to Conceptual data
foreach attribute in py_dataset(Attributes):
        //Compute the cutpoint for the attributes
        foreach cut_point in the cutPoint_list():
                //create two blocks "< cut_point" and ">=cut_point"
//Computing all possible combination of Attribute value pair
for len in range (min_rule_length, max_rule_length):        //Here the max and min length of the rule is entered by user
        //take all possible combination of attributes and values
        // store the possible combination in a list
//Induce rule from the combination of attribute value pair
foreach possible_rule in combination_list():
        if possible_rule.Strength > min_Strength and possible_rule.con_prob > min_Conditional_prob
                induced_rules = induced_rules U possible_rule
                //Storing the induced rule in a list
//Display the induced rules
induced_rules = sort_Rules_Strength(induced_rules)      //sorting the rules based on Strength factor
induced_rules = sort_Rules_Cond_prob(induced_rules)     //again sorting the rules based on Cond_prob
print(induced_rules)
end_IRIM
```

Result

- Dataset : IRIS
  - Min rule length: 1
  - Max rule length: 1
  - Min Strength: 10
  - Min Conditional probability: 1
- Rules Induced: 29

Result

- Top few Rules induced from IRIS dataset:
  - (petal_length, <= 2.45) --> (class, Iris-setosa)
  - (petal_width, <= 0.80) --> (class, Iris-setosa)
  - (petal_width, <= 0.55) --> (class, Iris-setosa)
  - (petal_length, <= 1.80) --> (class, Iris-setosa)
  - (petal_length, > 5.15) --> (class, Iris-viginica)

Analysis

- When compared to other rule induction systems, IRIM usually takes more computational time. But it provides all possible rules for a dataset that satisfy the user specification.

- The time complexity of the IRIM algorithm is exponential proportional to the number of attribute and the number of distinct values of attributes.

Analysis

- With this project, we observe that we can improve the performance by giving condition such as max and min rule length or defining min strength for the rule.

Future Work

- Future work need to be done on the cutpoint method for numerical data, as we are introducing many redundant rules, for example:
  - (Age, < 27) & (HyperTension, no) → (Decision, preterm)
  - (Age, < 31) & (HyperTension, no) → (Decision, preterm)

Conclusion

- The goal of the project was to understand the IRIM rule induction in exploring the hidden knowledge about the data

- Most rule induction systems aim to cover all cases to ensure the rule set are consistent. IRIM induce the strongest possible rules which covers most of the concept

References

- [1] Grzymala-Busse, J. (2009). Rule Induction. Maimon O., Rokach L.: Springer, Boston, MA. doi:https://doi.org/10.1007/978-0-387-09823-4_13
- [2] Grzymala-Busse, J., & Grzymala-Busse, W. (2005). Handling Missing Attribute Values. Maimon O., Rokach L.: Springer, Boston, MA. doi:https://doi.org/10.1007/0-387-25465-X_3
- [3] Grzymala-Busse, J., Grzymala-Busse, W., & Hamilton, J. (2005). Discriminant versus Strong Rule Sets (pp 67-76). Klopotek M.A., Wierzchon S.T., Trojanowski K.: Springer, Berlin, Heidelberg. doi:https://doi.org/10.1007/3-540-32392-9_8
- [4] Grzymala-Busse, J., Hamilton , J., & Hippe, Z. (2004). Diagnosis of Melanoma Using IRIM, a Data Mining System. Rutkowski L., Siekmann J.H., Tadeusiewicz R., Zadeh L.A.: Springer, Berlin, Heidelberg. doi:https://doi.org/10.1007/978-3-540-24844-6_155
- [5] J.W., G.-B. (1991). On the unknown attribute values in learning from examples. Springer, Berlin, Heidelberg. doi:https://doi.org/10.1007/3-540-54563-8_100
- [6] Usama, F., Gregory, P.-S., & Padhraic, S. (1996). From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence.

Questions?

Thank You

- Rule specification:
  - Min length of rule = 2
  - Ratio parameter = 1
  - Min Strength = 2
- Set of all possible conditions
  - 4 distinct values for Attribute Age
  - 2 distinct values for Attribute HyperTension
  - 2 distinct values for Attribute Complication

(4*2) + (4*2) + (2*2) + (4*2*2)
= 52 combination possible