Mining Knowledge Of The Directed Acyclic Graph (DAG) And Dataset Using The Hill Climbing Algorithm

Munirah, Aslan Alwi

Abstract: Hill Climbing Algorithm is used by people to produce bayes (bayes network) symptoms in the form of directed acyclic grpah (DAG). With this algorithm look for the optimal DAG of a dataset. However, a DAG is a symptom of causality / causation of bayes so that the optimal DAG search of a dataset is equivalent to the search for symptom causality that is most likely (optimum) between attributes or data variables. This means finding knowledge in the form of a causal relationship. Therefore, it is reasonable to mine the form of knowledge expressed in the form of rules from DAG by converting trending arrows between nodes as if-then relationships between variables. In this study, it was proposed how to mine knowledge (set of rules) from the dataset by using the optimal DAG from a dataset assuming that the optimal DAG produces the most optimal set of rules. Rule mining in this way uses hill climbing algorithms as a tool to produce optimal DAG. There are algorithms other than hill climbing such as ACO or Genetic algorithms, but the choice is dropped on hill climbing algorithms as the first trial of research.

Index Terms:Hill Climbing algorithm, Set of rules, Mining Knowledge, Datasets, Optimal DAG, IF-Then, Bayes Network.

1. INTRODUCTION

Rule-based system is a system that rests on a set of rules used to carry out knowledge inference. Basically, the system collects knowledge into knowledge representation in the form of a rule. However, the knowledge in the form of the rule is inductive, meaning that the rule can be built by algorithms by studying a limited number of cases then the rule of induction from a limited number of cases then generalizes it to reality in general over time. In a rule-based system, each rule is arranged in an IF-THEN form. Rule consisting of antecedents (conditions) and consequent (conclusions). The facts of the antecedents and the consequent constituents can be either single facts or a combination of several facts at once which are connected by logic connectors such as AND or OR. As an effort is made to translate and model human knowledge into a form that can be processed by a machine (computer), it takes a form that is flexible enough to represent knowledge (rules) as well as simple enough to be implemented into programming languages. The use of the IF-THEN model is considered effective by associating facts into premises. So, in general the term premise is used to represent facts in its representation.

Many rules have been generated at this time to be implemented into smart-based system applications. A set of rules is formed for a case or more by a number of experts from a number of applicable data. A number of experts have succeeded in forming a set of rules with the results of analysis in their fields accompanied by supporting data available and even some that only use experience in their fields. For example, an ophthalmologist can diagnose the type of eye disease suffered by his patient simply by concluding from a complaint given by the patient, this can be said as intuition-based knowledge and the experience of the doctor who has been in the field for quite a long time. Based on that, the doctor can produce a number of rules to be translated into a collection of knowledge acquired into a knowledge-based intelligent system, one of which is an expert system. Meanwhile, at the stage of knowledge acquisition in the form of a number of rules into an intelligent knowledge-based system, it has also used many algorithm variants that have been developed to produce a set of rules. A number of studies related to rule mining have been produced, including using decision tree models [1], [2], [3]. Rule induction algorithms used in decision tree models include CN2 algorithm [4], algorithm ID3 [5], also C4.5 algorithm which is a classic algorithm but is believed to be very good for extracting rules with minimal error rates [6], [7] Meanwhile, there are also those who mine knowledge by using association rules to produce a number of rules [8], with the AIS algorithm [9], with a priori algorithm [10], with the ECLAT algorithm [11], FP-Growth algorithm [12], and ARM algorithm [13]. There are also those who use learning machines with knowledge neural networks to extract rules from a number of datasets while at the same time making improvements to a number of rules produced with the concept of learning [14], [15]. This method was developed and optimized into a Fuzzy Logical Rules based concept using Gaussian uncertainty measurements [16]. The use of the rough set concept has also been developed to mine a number of rules [17], [18]. There is also one that is applied by using a domination approach with the algorithm produced, namely DOMLEM which can induce a set of general decision rules that are consistent with the principle of dominance [19]. In this study, we want to introduce another way to mine knowledge from the dataset. That is, by first mining the most probable causal relationship between the variables in the dataset, then from the causality relationship it is converted to the if-then set or even to rule. Each relationship if it has a probability value, therefore it is necessary to specify a constant which is a probability threshold that may be accepted. This method reduces the number of possible rules obtained from the optimum causality relationship (DAG) of a dataset.

- *Munirah, Aslan Alwi*
- *Department of Informatics Engineering, Faculty of Engineering, Universitas Muhammadiyah Ponorogo, Indonesia*
- *Email: munirah.mt@gmail.com*

2 PROCEDURE FOR EXTRACTING DAG AND RULE

Hill Climbing is basically an optimization algorithm that works how to test each node or data point. When the data point meets the target function that evaluates the data point and its value is better than the previous data point, the hill climbing algorithm will set the data point as the optimum, and so on he tests all the closest data points. If there is something better, the better data points will be crowned as the next best data point. In the bnlearn library belonging to the R language on Rstudio is a library that helps to build a bayes network model from a given dataset. There are several algorithms to construct the Bayes network model. One of them is the hc function. The hc function is a function that produces the optimum bayes network model from a dataset using a hill climbing algorithm. In this case, hill climbing is used to look for the directed acyclic graph (DAG) model from the given dataset. DAG is a Bayes causality model. The role of hill climbing algorithms only reaches here, which is constructing a DAG model. Furthermore, there are other functions from the literature library to utilize the model to perform learning and prediction parameters. Learning parameters are the learning process of the DAG model that has been created so that all probability values (conditional probability or joint probability) can be obtained. The idea of this research is to utilize the DAG model produced by the hc algorithm algorithm hill climbing function. By extracting the DAG model from the data it is the same as producing the optimum causality model from the data. This causality model is then translated as a if-then model or rule that states a causal relationship. Each model if it is obtained is having a probability value that can be known by doing learning parameters on its DAG model. By setting an acceptable limit of probabilities, the set of rules can be obtained from the if-then model that happens. This is indirectly, research uses Algorithm hill climbing to harvest the rule from the dataset. The flow of harvesting rules or extracting these rules is as follows:

1. Start
2. Prepare a dataset
3. Import into Rstudio
4. Convert dataset to format factor
5. Bayes network construction uses hill climbing algorithm (learning structure uses hc function in library bnlearn)
6. Bayes network (bn) model is also obtained, which is also a DAG model as well as the Bayes causality model
7. Extract if-then models or causality from DAG
8. Set a minimum probability value to accept that a rule instance is feasible (Pmin)
9. Perform learning parameters by using the bn.fit function to calculate the probability of an if-then relationship
10. Show the probability of each rule instance (conditional probability, cp)
11. Extract instance rules from each if-then model
12. Reducing the set of rule instances on each model if-then using Pmin previously set. That is by cutting all rule instances that have cp < Pmin
13. Obtained a set of rules that have cp >= Pmin
14. Extraction rule is complete
15. End

The steps of the rule extraction process are carried out by using an example of a default dataset from the R language on Rstudio and it is also planned to test it on datasets that come from open sources that can be obtained on sites that provide open data that is free to use for research. These steps do not directly use the hill climbing algorithm to produce a rule, but use the hill climbing algorithm to first extract the most possible causal relationship (optimum) from the dataset, then from the DAG model or optimum causality obtained, then extract the model- if-then rule model. Then from the rule models if-then, extract the rules of the instance. Namely the rule model, if any atomic proposition has an instance value.

For example, the if-then model is "IF A AND B THEN C". The form of the rule instance of this model is:

1. "IF A = 2 AND B = 100 THEN C = no"
2. "IF A = 5 AND B = 50 THEN C = yes"
3. "IF A = 2 AND B = 150 THEN C = no"
4. "IF A = 7 AND B = 10 THEN C = yes"
5. etc...

The relationship of the "IF A AND B THEN C" model is that this mode can be extracted from the following DAG example:

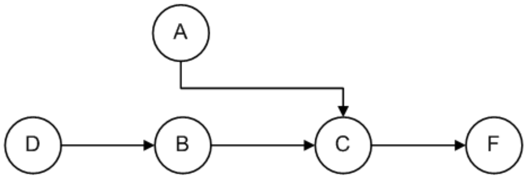


***Figure 1**. Example of a network bayes (DAG) model*

From the model in figure 1, the model can be extracted as follows:

1. IF D THEN B
2. IF B THEN C
3. IF C THEN F
4. IF A THEN C
5. IF A AND B THEN C

Each probability of the model if-then is calculated using a conditional probability from the dataset, as follows:

1. IF D THEN B with cp P(B|D)
2. IF B THEN C with cp P(C|B)
3. IF C THEN F with cp P(F|C)
4. IF A THEN C with cp P(C|A)
5. IF A AND B THEN C with cp P(C|A,B)

This cp value is calculated by calculating the rule instance. Example:
IF D THEN B has an instance IF D = 3 THEN B = 40 then the probability is P (B = 40 | D = 3).

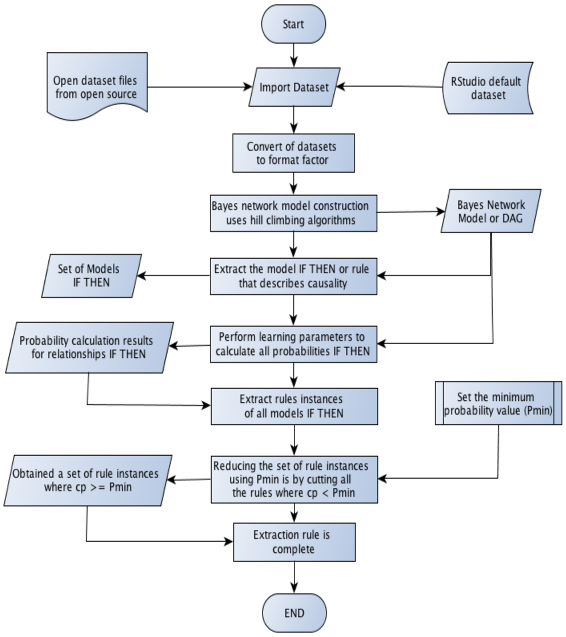

Start
Open dataset files from open source
Import Dataset
RStudio default dataset
Convert of datasets to format factor
Bayes network model construction uses hill climbing algorithms
Bayes Network Model or DAG
Set of Models IF THEN
Extract the model IF THEN or rule that describes causality
Probability calculation results for relationships IF THEN
Perform learning parameters to calculate all probabilities IF THEN
Extract rules instances of all models IF THEN
Set the minimum probability value (Pmin)
Obtained a set of rule instances where cp >= Pmin
Reducing the set of rule instances using Pmin is by cutting all the rules where cp < Pmin
Extraction rule is complete
END

Figure 2. *Flow chart extraction rule*

The steps to extracting this rule can be displayed in a flow chart as shown in Figure 2. above. This flow chart states the steps as the 15 steps of harvesting the rules stated earlier on page 2.

3 RULE EXTRACTION DEMONSTRATION

The following is a demonstration about how to extract the rule as a flow chart 4.2. given above.
Step 1:
Open Rstudi then activate the library, if it doesn't already exist, then install it.
#Library installation installed if it is not already installed
> install.packages("bnlearn")

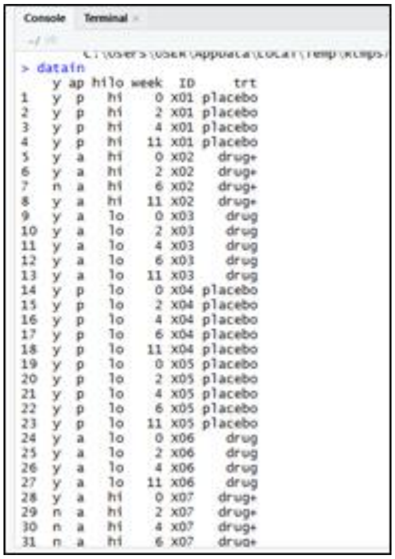
#Activate library material
#Core library to construct the Bayes network model
>library(bnlearn,quietly=TRUE,verbose=FALSE, warn.conflicts = FALSE)
#Dplyr library installation if it is not already installed
> install.packages("dplyr")

#Activate dplyr library
#This library is to convert if there is a data value that is NULL or NA # becomes 0 so that the dataset can be processed
> install.packages("dplyr")

#Take this built-in data (default) in the R language, here data is selected which tests the existence of H. influenzae bacteria in children with otitis media in the Northern territory of Australia
> datain=MASS::bacteria

#Display data

> datain



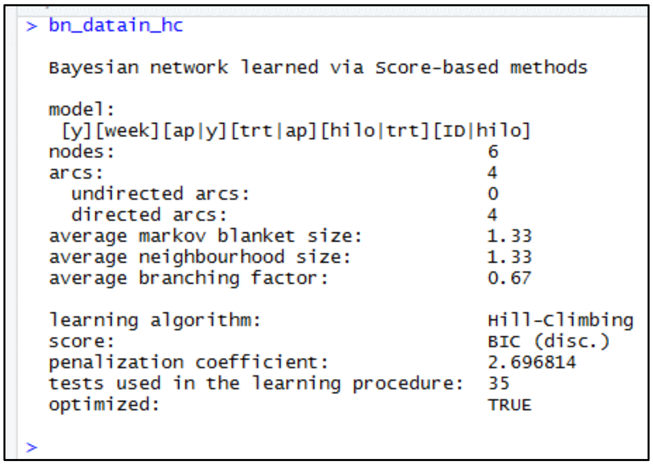
```
     C:\Users\USER\AppData\Local\Temp\RtmpS
> datain
    y ap hilo week  ID     trt
1   y  p   hi    0 X01 placebo
2   y  p   hi    2 X01 placebo
3   y  p   hi    4 X01 placebo
4   y  p   hi   11 X01 placebo
5   y  a   hi    0 X02   drug+
6   y  a   hi    2 X02   drug+
7   n  a   hi    6 X02   drug+
8   y  a   hi   11 X02   drug+
9   y  a   lo    0 X03    drug
10  y  a   lo    2 X03    drug
11  y  a   lo    4 X03    drug
12  y  a   lo    6 X03    drug
13  y  a   lo   11 X03    drug
14  y  p   lo    0 X04 placebo
15  y  p   lo    2 X04 placebo
16  y  p   lo    4 X04 placebo
17  y  p   lo    6 X04 placebo
18  y  p   lo   11 X04 placebo
19  y  p   lo    0 X05 placebo
20  y  p   lo    2 X05 placebo
21  y  p   lo    4 X05 placebo
22  y  p   lo    6 X05 placebo
23  y  p   lo   11 X05 placebo
24  y  a   lo    0 X06    drug
25  y  a   lo    2 X06    drug
26  y  a   lo    4 X06    drug
27  y  a   lo   11 X06    drug
28  y  a   hi    0 X07   drug+
29  n  a   hi    2 X07   drug+
30  n  a   hi    4 X07   drug+
31  n  a   hi    6 X07   drug+
```

Figure 3. *Dataset bacteria*

#Convert the dataset into factor type, because it is required by the library bnlearn
>datafac=data.frame(y=factor(datain$y),ap=factor(datain$ap), hilo=factor(datain$hilo),week=factor(datain$week),ID=factor(datain$ID),trt=factor(datain$trt))

#Construction of the bayes network using the hill climbing function (hc)
>bn_datain_hc=hc(datafac)

#Next, a bn_datain_hc bayes network model is obtained, to see the details of this model as follows:
>bn_datain_hc



```
> bn_datain_hc

  Bayesian network learned via Score-based methods

  model:
   [y][week][ap|y][trt|ap][hilo|trt][ID|hilo]
  nodes:                                 6
  arcs:                                  4
    undirected arcs:                     0
    directed arcs:                       4
  average markov blanket size:           1.33
  average neighbourhood size:            1.33
  average branching factor:              0.67

  learning algorithm:                    Hill-Climbing
  score:                                 BIC (disc.)
  penalization coefficient:              2.696814
  tests used in the learning procedure:  35
  optimized:                             TRUE

>
```

Figure 4. *Bayes network model (DAG)*

The Bayes network model obtained can be expressed in the form of:

[y][week][ap|y][trt|ap][hilo|trt][ID|hilo]

[y] states that the y attribute stands alone is not caused by another attribute.
[week] as well as week attribute.
[ap | y] states that the ap attribute is caused by the y attribute, or has a causality relationship with y.

Likewise the causality relationship is expressed by [trt | ap], [hilo | trt] and [ID | hilo].

# To be able to see how the DAG form of this bayes network model, # can be used by the command:
> plot(bn_datain_hc)

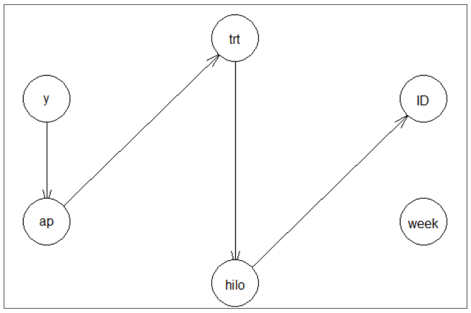


***Figure 5**. Directed Acyclic Graph (DAG)*

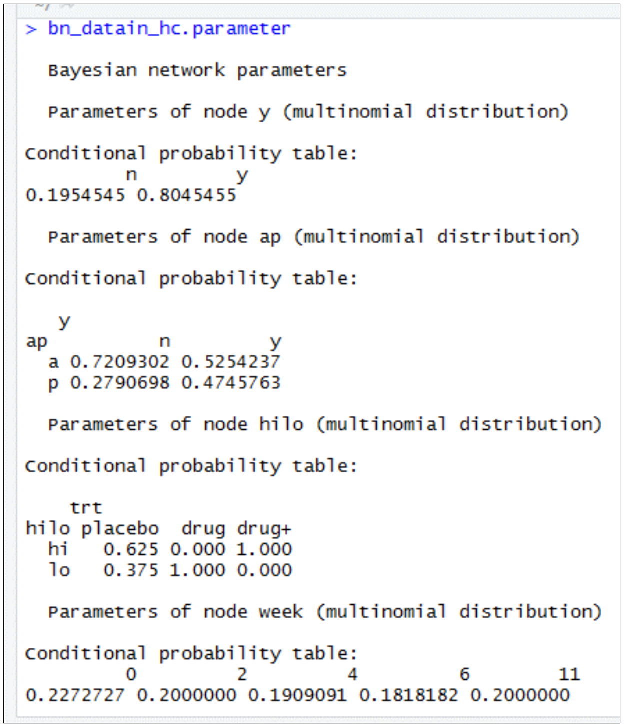
From the construction of this bayes network, models can be extracted IF THEN as follows:

1. IF y THEN ap
2. IF ap THEN trt
3. IF trt THEN hilo
4. IF hilo THEN ID

Next is the extraction rule instance of models IF THEN this can be done by first doing learning parameters.

#To perform learning parameters, that is, the probability calculation of DAG can be done as follows:
> bn_datain_hc.parameter = bn.fit(bn_datain_hc, datafac)

# Details of the overall probability calculation can be done with the command:
> bn_datain_hc.parameter

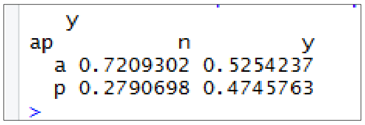


```
> bn_datain_hc.parameter

  Bayesian network parameters

  Parameters of node y (multinomial distribution)

Conditional probability table:
         n         y
0.1954545 0.8045455

  Parameters of node ap (multinomial distribution)

Conditional probability table:

   y
ap         n         y
  a 0.7209302 0.5254237
  p 0.2790698 0.4745763

  Parameters of node hilo (multinomial distribution)

Conditional probability table:

    trt
hilo placebo  drug drug+
  hi   0.625 0.000 1.000
  lo   0.375 1.000 0.000

  Parameters of node week (multinomial distribution)

Conditional probability table:
        0         2         4         6        11
0.2272727 0.2000000 0.1909091 0.1818182 0.2000000
```

***Figure 6.** Conditional probability calculation results*

The rule extraction process for the rule model IF y MAK is ap along with all probability values each instance is used by the command:
> bn_datain_hc.parameter$ap$prob

Obtained the probability table as well as instances of the model rule as follows:



```
   y
ap          n         y
  a 0.7209302 0.5254237
  p 0.2790698 0.4745763
>
```

***Figure 7.** Conditional probability P (ap | y)*

Based on the probability table above, the rules and their probabilities can be extracted as follows:
IF y=n THEN ap=a P(ap=a|y=n) = 0.7209302
IF y=n THEN ap=p P(ap=p|y=n) = 0.2790698
IF y=y THEN ap=a P(ap=a|y=y) = 0.5254237
IF y=y THEN ap=p P(ap=p|y=y) = 0.4745763

By setting the minimum probability limit that can be accepted to become a rule is Pmin = 0.5, the rule that can be accepted as an extract for the model rule IF y THEN AP is the following rules:
IF y=n THEN ap=a P(ap=a|y=n) = 0.7209302 $\geq$ 0.5=P_{min}
IF y=y THEN ap=a P(ap=a|y=y) = 0.5254237 $\geq$ 0.5=P_{min}

In the same way, the rules for the model rule can be extracted as follows:
IF ap THEN trt
IF trt THEN hilo
IF hilo THEN ID

For the week attribute there is no causal relationship to it so that a number of rules cannot be taken from it.

Consecutive commands are given to extract probabilities and instances of each model rule as follows:

```
> bn_datain_hc.parameter$trt$prob
          ap
trt       a      p
placebo 0.0    1.0
drug    0.5    0.0
drug+   0.5    0.0
```

Obtained the following rules:
IF ap=a THEN trt=drug
because P(ap=a|trt=drug)=0.5 $\geq$ 0.5 = P_{min}
IF ap=a THEN trt=drug+
because P(ap=a|trt=drig+)=0.5 $\geq$ 0.5 = P_{min}
IF ap=p THEN trt=placebo
because P(ap=p|trt=placebo)=1.0 $\geq$ 0.5 = P_{min}

```
> bn_datain_hc.parameter$hilo$prob


               trt
hilo    placebo  drug     drug+
 hi     0.625    0.000    1.000
 lo     0.375    1.000    0.000
```

Obtained the following rules:
IF trt=drug THEN hilo=lo
because P(trt=drug|hilo=lo)=1.000 $\geq$ 0.5 = P_{min}
IF trt=drug+ THEN hilo=hi

because P(trt=drig+|hilo=hi)= 1.000 $\geq$ 0.5 = P_{min}
IF trt=placebo THEN hilo=hi
because P(trt=placebo|hilo=hi)=0.625$\geq$0.5 = P_{min}

```
> bn_datain_hc.parameter$ID$prob
                    hilo
ID          hi                   lo
  X01   0.03278689    0.00000000
  X02   0.03278689    0.00000000
  X03   0.00000000    0.05102041
  X04   0.00000000    0.05102041
  X05   0.00000000    0.05102041
  X06   0.00000000    0.04081633
  X07   0.04098361    0.00000000
  X08   0.04098361    0.00000000
  X09   0.00000000    0.05102041
  X10   0.01639344    0.00000000
  X11   0.04098361    0.00000000
  X12   0.00000000    0.05102041
  X13   0.04098361    0.00000000
  X14   0.00000000    0.03061224
  X15   0.00000000    0.05102041
  X16   0.04098361    0.00000000
  X17   0.04098361    0.00000000
  X18   0.00000000    0.05102041
  X19   0.00000000    0.05102041
  X20   0.03278689    0.00000000
  X21   0.04098361    0.00000000
  Y01   0.04098361    0.00000000
  Y02   0.00000000    0.05102041
  Y03   0.04098361    0.00000000
  Y04   0.00000000    0.04081633
  Y05   0.02459016    0.00000000
  Y06   0.00000000    0.04081633
  Y07   0.00000000    0.05102041
  Y08   0.03278689    0.00000000
  Y09   0.00000000    0.04081633
  Y10   0.00000000    0.05102041
  Y11   0.04098361    0.00000000
  Y12   0.01639344    0.00000000
  Y13   0.00000000    0.04081633
  Y14   0.02459016    0.00000000
  Z01   0.00000000    0.03061224
  Z02   0.04098361    0.00000000
  Z03   0.04098361    0.00000000
  Z05   0.04098361    0.00000000
  Z06   0.01639344    0.00000000
  Z07   0.00000000    0.05102041
  Z09   0.03278689    0.00000000
  Z10   0.04098361    0.00000000
  Z11   0.00000000    0.05102041
  Z14   0.04098361    0.00000000
  Z15   0.00000000    0.03061224
  Z19   0.04098361    0.00000000
  Z20   0.00000000    0.04081633
  Z24   0.04098361    0.00000000
  Z26   0.04098361    0.00000000
```

For the IF hilo rule model, there is no rule that can be taken because there is no rule that exceeds the probability of Pmin.

4 CONCLUSIONS AND RECOMMENDATIONS

This study does not use hill climbing algorithms to harvest rules directly from the dataset. However, using hill climbing is only for harvesting DAG models or the best bayes causality model in the form of bayes network. Then from bayes netowrk, rule models can be extracted, namely models IF THEN are possible. Then from the models if it is, the forms of instance models are extracted complete with the conditional probability. Based on the lower limit of the specified conditional probability value, then the collection of rule instances obtained is reduced in number based on the specified conditional probability limit. In this way, a set of rules is obtained from the optimum bayes causality model of a dataset. All these rules can be used to do expert system reasoning if needed. Even available probability values if probabilistic reasoning is to be done. For future research, an algorithm is needed to automatically harvest the rule instances of the DAG models obtained, so there is no need to extract manually from the DAG model.

ACKNOWLEDGMENT

The author is grateful to the LPPM Muhammadiyah University in Ponorogo as supporting both funding and morale for the ongoing research.

REFERENCES

[1] Quinlan, J.R., 1986. Induction of Decision Trees. Expert Systems, pp.81–106
[2] Wang, Y. & Witten, I.H., 1997. Inducing Model Trees for Continuous Classes. European Conference on Machine Learning (ECML), pp.1–10. Available at: http://www.cs.waikato.ac.nz/~ml/publications/1997/Wang-Witten-Induct.pdf
[3] Provost, F., 2003. Tree Induction for Probability-Based Ranking. , 5, pp.199–215
[4] Clark, P. & Niblett, T., 1989. The CN2 Induction algorithm. Machine Learning, 3(4), pp.261–284. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.3672&rep=rep1&type=pdf
[5] Khera, D. et al., 1991. Knowledge Verification of Machine-learning Procedures Based on Test Structure Measurements. , 4(1), pp.145–149
[6] Cohen, W.W., 1995. Fast Effective Rule Induction. Machine Learning: Proceedings of the Twelfth International Conference
[7] Frank, E. & Witten, I.H., 1998. Generating accurate rule sets without global optimization. Proceedings of the Fifteenth International Conference on Machine Learning, pp.144–151. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.143.8073&amp;rep=rep1&amp;type=pdf
[8] Agrawal, R. & Srikant, R., 1994. Fast algorithms for mining association rules. In Proceedings of the 20th VLDB conference, pp.487–499
[9] Agrawal, R., Imieliński, T. & Swami, A., 1993. Mining association rules between sets of items in large databases. ACM SIGMOD Record, 22(2), pp.207–216. Available at: http://portal.acm.org/citation.cfm?doid=170036.170072
[10] Kavsek, B., Lavrac, N. & Jovanoski, V., 2003. APRIORI-SD : Adapting Association Rule. Learning, pp.230–241
[11] Zaki, M.J., 2000. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3), pp.372–390

[12] Han, J. et al., 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery, 8(1), pp.53–87

[13] Djenouri, Y. et al., 2014. An Efficient Measure for Evaluating Association Rules. , pp.406–410

[14] Towell, G.G. & Shavlik, J.W., 1993. Extracting Refined Rules from Knowledge-Based Neural Networks. Machine Learning, 13(1), pp.71–101

[15] Craven, M.W. & Shavlik, J.W., 1994. Using sampling and queries to extract rules from trained neural networks. Machine Learning: Proceedings of the Eleventh International Conference, pp.37–45

[16] Duch, W., Adamczak, R. & Grąbczewski, K., 2001. A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. IEEE Transactions on Neural Networks, 12(2), pp.277–306

[17] Pawlak, Z., 1996. Rough sets and data analysis. Soft Computing in Intelligent Systems and Information Processing. Proceedings of the 1996 Asian Fuzzy Systems Symposium, 147, pp.1–12

[18] Grzymala-busse, J.W., 2005. Rough Set Theory with Applications to Data Mining. Real World Applications of Computational Intelligence, Volume 179, p.pp 221-244

[19] Greco, S. et al., 2001. An algorithm for induction of decision rules consistent with the dominance principle. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2005, pp.304–313