An Application of Predictive Data Mining Technique for Classifying EcoGene Data

Onima Tigga[1], Amrita Priyam[1]

Birla Institute of Technology, Ranchi, Jharkhand, India

Abstract

Predictive data mining technique is an effective way of classification. In this paper, an algorithm of constructing classification rule is presented. It is based on the logical relationship between attributes. The EcoGene database has been considered and it contains updated information about the E. coli K-12 genome and proteome sequences, including extensive gene bibliographies[13]. In this paper three attributes of EcoGene database have been considered for rule development which in turn has given three classes. For these three classes experimental results are also validated.

Keywords: *Predictive Data Mining, Classification, Rule-Based Method, EcoGene data*

1. Introduction

Data mining in general is a term which refers to extracting or "mining" knowledge from huge amounts of records. It is the discipline of finding novel remarkable patterns in vast quantity of data. It is also sometimes called knowledge Discovery in databases (KDD) [1]. It requires sharp technologies and the compliance to discover the possibility of hidden knowledge that resides in the data. The two primary goals of data mining are prediction and description. Its approaches seem ideally suited for Bioinformatics, since it is data-rich, but lacks a comprehensive theory of life's organisation at the molecular level[10]. The extensive database of biological information crafts both challenges and opportunities for development of novel KDD methods. Mining biological data helps to extract useful knowledge from massive datasets gathered in biology, and in other related life sciences areas such as medicine and neuroscience etc [10].

Classification is one of the predictive data mining task. Classification rule, as an important content of data mining, is widely used in areas like marketing, financial investment, astronomy, geographical analysis, bioinformatics etc. This can be extracted by using Rule-Based Methods, Decision Tree, rough set theory, Genetic Algorithm and so on[14]. An IF- Then rule for classification is used because of its simplicity.

In this paper, we took dataset from EcoGene, contains three attributes length (len), Centisome (Cs) and molecular weight (MW) and analyzed those data and extracted rules from them. Rules have been generated from them which had good accuracy and minimum errors.

1.1 Predictive Data Mining

Predictive data mining is the most familiar and most popular data mining technique. The main predictive data mining tasks are: Classification, Regression, Time Series analysis, Prediction. Classification has been chosen for classifying EcoGene database and also for predicting new tuples to a correct class. Classification maps a data item into one of several predefined classes.

A classification problem is a supervised learning problem in which the output information is a discrete classification, if given an object and its input attributes, the classification output is one of the possible mutually exclusive classes of the problem. The aim of the classification task is to discover some kind of relationship between the input attributes and the output class, so that the discovered knowledge can be used to predict the class of a new unknown object [1].

(Definition): Given a database D= {t1, t2,…, tn} of tuples (items, records) and a set of classes C = {C1,…,Cm}, the classification problem is to define a mapping f: D→ C where each ti is assigned to one class. A class Cj, contains precisely those tuples mapped to it,

i.e. $Cj = \{ti \mid f(ti) = Cj, 1 \leq i \leq n, ti \in D\}$.

1.2 Rule Based Classification

Rules are a good way of representing information or bits of knowledge. A rule-based classifier uses a set

of IF-Then rules for classification. An IF-Then rule is an expression of the form

IF condition Then conclusion.

The "IF" – part of a rule is known as the rule antecedent. The "Then" – part is the rule consequent. In the rule antecedent, the condition consists of one or more attribute tests that are logically ANDed. The rule's consequent contains a class prediction. If the condition in a rule antecedent holds true for a given tuple, we say that the rule antecedent is satisfied and that the rule covers the tuple. A rule R can be assessed by its coverage and accuracy. Given a tuple X, n_{covers} is the number of tuples covered by R, $n_{correct}$ is the number of tuples correctly classified by R and |D| is the number of tuples in D.

We can define the Coverage and Accuracy of R as

$$Coverage(R) = n_{covers} / |D|$$

$$Accuracy(R) = n_{correct} / n_{covers}$$

That is, a rule's coverage is the percentage of tuples that are covered by the rule and rule's accuracy is the number of tuples correctly classified[3].

2. Sequential Covering Algorithm

Algorithm: Sequential Covering
Input: D, a data set class-labeled tuples; Att_vals, the set of all attributes and their possible values.
Output: A set of IF- Then rules.

Method:

1. Rule-set ={ }; // initial set of rules learned is empty
2. for each class C do
3. repeat
4. Rule = Learn-One-Rule (D, Att_vals, C);
5. Remove tuples covered by Rule from D;
6. Until terminating condition;
7. Rule-set = Rule-set+ Rule; // add new rule to set rule
8. end for
9. Return Rule-set;

2.1 Example Analysis

The example consists of dataset contains 15 tuples and has three attributes length, Centisome (Cs) and molecular weight (MW). The Output 1 in Table 2 is assumed on the basis of the following conditions taken together for each class as shown in Table 1.

Table1. Conditions Taken for Classes.

Length	class
700≤ length	good
350<length< 700	average
length ≤ 350	bad
Cs	class
65 ≤ Cs	good
35 ≤ Cs < 65	average
Cs≤ 35	bad
MW	class
65000 ≤ MW	good
35000 ≤ MW < 65000	average
MW < 35000	bad

Table 2. A Sample Example of EcoGene dataset.

Sl. No.	Length	Cs	MW	Output 1
1	229	17.14	24938.82	bad
2	376	48.72	39647.83	average
3	258	0.12	49057.75	bad
4	440	41.78	73339.24	average
5	646	73.27	87982.91	good
6	778	72.18	19858.88	good
7	183	10.57	55018.23	bad
8	504	42.71	39137.45	average
9	361	52.68	19150.83	average
10	174	8.74	85850.66	bad
11	777	80.01	81260.12	good
12	715	93.85	50512.5	good
13	450	39.12	22622.62	average
14	204	9.64	63998.12	bad
15	570	62.21	63998.12	average

To generate rules, sequential covering algorithm is used as they generate the best rule possible by optimizing the desired classification probability. Usually the "best" Att-value pair is chosen to show good class. The basic format for the rule is then

If ? Then class = good

The objective of the algorithm is to replace the "?" in this statement with predicates that can be used to obtain the "best" probability of being good. We have used 1R(One Rule) classification to choose the best attribute to perform classification and best is defined here by counting the number of errors[3].

Table 3. 1R Classification.

Sl	Att	Rules	Error	Total Errors
1.	Length	(0,200] → bad	0/2	
		(200, 300] → bad	0/2	
		(300, 400] → average	0/2	1/15
		(400, 500] → average	0/3	
		(500, 600] → good	1/1	
		(600, ∞] → good	0/4	
2.	Cs	(0, 20] → bad	0/5	
		(20,30] → bad	0/0	
		(30, 40] → bad	1/1	2/15
		(40, 50] → average	0/3	
		(50, 60] → average	0/1	
		(60, 70] → good	1/1	
		(70, ∞] → good	0/4	
3.	MW	(0, 25,000] → bad	0/4	
		(25,000, 35,000] → bad	0/1	
		(35,000,45,000]→ average	0/2	2/15
		(45,000,55,000]→average	0/2	
		(55,000, 65,000] → good	2/2	
		(65,000, ∞] → good	0/4	

Thus length attribute is chosen first to generate rules with minimum error rate. Probability of putting a tuple in the good class based on the given Attribute-value pair.

Table 4. Probability of putting tuple in the classes.

Rules	Probability	
length ≤ 200	0/2	
200 < length ≤ 300	0/3	Coverage=0.266
300 < length ≤ 400	0/2	Accuracy=100%
400 < length ≤ 500	0/3	
500 < length ≤ 600	0/1	
600 < length	4/4	
Cs ≤ 20	0/5	
20 < Cs ≤ 30	0/1	
30 < Cs ≤ 40	0/0	Coverage=0.40
40 < Cs ≤ 50	0/3	Accuracy=100%
50 < Cs ≤ 60	0/1	
60 < Cs ≤ 70	0/1	
70 < Cs	4/4	
MW ≤ 25000	0/4	
25000 < MW ≤ 35000	0/1	Coverage=0.333
35000 < MW ≤ 45000	0/2	Accuracy=100%
45000 < MW ≤ 55000	0/2	
55000 < MW ≤ 65000	0/2	
65000 < MW	4/4	

Learn-One-Rule finds that the attribute test length > 600 best improves the accuracy of our current (empty) rule. After appending it to the condition, the current rule becomes

If length > 600 Then class = good

Each time we add an attribute test to a rule, the resulting rule should cover more of the "good" tuples. During the next iteration, we again consider the next attribute Cs.

Probability of putting a tuple in the good class based on the given Att-val pair. Learn-One-Rule finds that the attribute test Cs > 70 best improves the accuracy of our current rule. After appending it to the condition, the current rule becomes.

If length > 600 and Cs > 70 Then class = good

By adding an attribute MW to the Rule, the resulting rule covers the more of the tuples and the tuples covered by the sequential covering algorithm are removed from the Table1 and the processing is continued with the remaining tuples. The above rule R1 covers 4 tuples from the 15 tuples and next rule for the average class is generated based on remaining 11 tuples and the same process is repeated and again the process is repeated for the bad class[3].

The rules developed for the classification purposes are as follows:

Rule 1: If length > 600 and Cs > 70 and MW > 65000 Then class = good

Rule 2: If 300 < length ≤ 600 and 20 < Cs ≤ 70 and 35000 < MW ≤ 65000 Then class = average

Rule 3: If length ≤ 300 and Cs ≤ 20 and MW ≤ 35000 Then class = bad

2.2 Observation and Results

The Coverage and accuracy are calculated and shown in the following Table:

Table 5. Coverage and accuracy.

Rules	Coverage	Accuracy
Rule R1	0.266	100%
Rule R2	0.40	100%
Rule R3	0.333	100%
Total	0.999	100%

3. Testing

For testing the following dataset were considered.

Table 6. Data set for testing.

Length	Cs	MW	Output 2
472	43.73	53464.8	average
273	0.61	28756.61	bad
400	18.97	43608.91	-
338	54.76	37614.57	average
475	61.9	5255847	average
571	47.85	64612.33	average
878	97.92	96482.39	good
692	61.48	78467.95	-
271	13.33	29784.11	bad
715	92.58	79223.81	good

After testing the accuracy and error were found as shown in Table 7:

Table 7. Accuracy and error of 10 tuples.

Accuracy	Error
0.80	0.20

This shows that the rules being considered are 80% accurate and having 20% errors.

Predictions were done for new tuples and there classes were shown in the following table 8:

Table 8. Prediction for new tuples.

Length	Cs	MW	Output
1034	73.54	111454.4	Good
156	7.73	17048.84	bad
338	39.85	37127.24	average
333	36.65	36397.46	average

4. Conclusion

Rule based method is highly expressive, easy to interpret, easy to generate, can easily handle missing values and numeric attributes. It is an efficient tool of classification rule extraction. Extraction of classification rule algorithm is presented on the basis of its accuracy. Testing results show that classification rule generated are evaluated based on its accuracy and coverage and it was found that classifying classes are accurate and with minimum error.

References:

1. M.H. Dunham & S. Sridhar- Data Mining: Introductory and Advanced Topics, Pearson education, 2006
2. Pujari, Arun: Data Mining Techniques, Nancy: Universities Press, 2001.
3. Jiwali Han, Kamber –Data Mining Concepts & Techniques, Morgan Kaufmann Publishers, 2008.
4. Cohen, W, "Fast Effective Rule Induction", In Proceedings of the 12th International Conference on Machine Learning, pp 115 – 123, 1995.
5. Kamber, M., Winstone, L., et al, "Generalization and decision tree induction: efficient classification in data mining", Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE'97) High Performance Database Management for Large-Scale Applications, pp, 111, April 07 – 08, 1997.
6. T.M. Mitchell, "Machine learning and data mining," commun. ACM, vol. 42, no. 11, 1999.
7. U. Fayyad, G.P. Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," Commun. ACM 39, pp 27 – 34, 1996.
8. Sankar K.Pal, S. Bandyopadhyay, S.S. Ray, "Evolutionary Computation in Bioinformatics: A Review", IEEE Transactions on systems, Vol. 36, No. 5, September 2006.
9. ZoheirEzziane, "Application of Artificial Intelligence in Bioinformatics: A Review", Expert systems with applications 30 (2006) 2-10.
10. P. K. Vaishali, Dr. A. Vinayababu, "Application of Data Mining and Soft

Computing in Bioinformatics", IJERA, ISSN:2248-9622, Vol. 1, Issue 3, pp. 758-771.

11. WlodzislawDuch, "Rule-Based Methods", wduch@is.umk.pl
12. Khalid Raza, "Application of Data Mining in Bioinformatics", Indian Journal of Computer Science and Engineering,ISSN : 0976 – 5166 117, Vol 1 No 2, 114-118.
13. Berlyn M.B., Low.K.B. and Rudd.K.E. (1996) In Neidhardt.F.C.,Curtiss.R., Ingraham.J.L., Lin.E.C.C., Low.K.B., Magasanik.B., Reznikoff.W.S., Riley.M., Schaechter.M. and Umbarger.H.E. (eds), Escherichia coli and Salmonella: Cellular and Molecular Biology. ASM Press, Washington, DC, Vol. 2, pp. 1715-1902.
14. Haifeng Yang, Zhihai Xu, Jifu Zhang and Jianghui Cai, "A Constructing Method of Decision Tree and Classification Rule Extraction for Incomplete Information System", International Conference on Computational Aspects of Social Networks (2010).