DOI: 10.1051/matecconf/201712504017

Using Machine Learning Methods Jointly to Find Better Set of Rules in Data Mining

Hyontai SUG[1*]

[1]Division of Computer Engineering, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 47011, Korea

Abstract. Rough set-based data mining algorithms are one of widely accepted machine learning technologies because of their strong mathematical background and capability of finding optimal rules based on given data sets only without room for prejudiced views to be inserted on the data. But, because the algorithms find rules very precisely, we may confront with the overfitting problem. On the other hand, association rule algorithms find rules of association, where the association resides between sets of items in database. The algorithms find itemsets that occur more than given minimum support, so that they can find the itemsets practically in reasonable time even for very large databases by supplying the minimum support appropriately. In order to overcome the problem of the overfitting problem in rough set-based algorithms, first we find large itemsets, after that we select attributes that cover the large itemsets. By using the selected attributes only, we may find better set of rules based on rough set theory. Results from experiments support our suggested method.

1 Introduction

Rough sets are one of widely accepted machine learning technologies. Good property of rough set theory is that it can describe uncertain facts solely based on data. As a result, there is no room for prejudiced views to be inserted in the discovered knowledge [1, 2, 3]. For example, assume that we have a decision table T like table 1. A decision table T is a table having conditional attributes and decision attributes in a fixed setting. Each attribute has corresponding values. In our example T contains the universe U of objects, x_1, x_2, x_3, x_4, and x_5. The set of attributes A has 2 attributes, a_1 and a_2. The set of decision class d has two values, 1 and 2. V_i is the set of values that each attribute has where i = 1, 2, and $V_1 = \{1, 2, 3\}$, $V_2 = \{1, 2\}$.

Table 1. A decision table T.

U	a_1	a_2	d
x_1	1	1	1
x_2	1	2	1
x_3	2	1	2
x_4	2	2	2
x_5	3	2	1

Possible set of minimal rules for T are as follows :

Rule 1. If (a_1 = 1) Then (d = 1)

Rule 2. If (a_1 = 2) Then (d = 2)

Rule 3. If (a_1 = 3) Then (d = 1)

As we can see from the example, rules based on rough set theory reflect the available data quite honestly.

There are also some other data mining or machine learning techniques solely based on data. Association rules [4, 5, 6] are one of representative techniques of such kind. Association rules find rules of association, where the association resides between sets of items in database. For example, assume that we have a simple database of transaction records like table 2 that shows the records of items purchased together in a supermarket.

Table 2. A simple transaction database.

Tr #	items
t_1	1, 3
t_2	2, 3
t_3	1, 2, 3
t_4	2, 5

[*] Corresponding author: sht@gdsu.dongseo.ac.kr

If minimum support that represents how many times an itemset occurs in a transaction record is two, the large itemset of length one can be found as in table 3. Itemset means the set of items that occur in the same records, and large itemset means that the itemset that occurs at least over given minimum support.

Table 3. Large itemset of length one.

itemset	support
{1}	2
{2}	3
{3}	3

Table 4 shows large itemset of length two.

Table 4. Large itemset of length two.

itemset	support
{2, 3}	2

There are no large itemset of length three, so we can stop the iteration. From the found large itemsets, we may generate association rules like; if item 2 has been purchased, then item 3 will be purchased also with the confidence of 67%, and if item 3 has been purchased, then item 2 will be purchased also with the confidence of 67%. As we see from the example, association rule algorithms find itemsets that occur more than given minimum support. This fact allows the algorithms to find the itemsets practically even for very large databases by supplying the minimum support appropriately.

2 Method and experiment

Because rough set theory-based algorithms find rules very thoroughly, it may confront with the overfitting problem. Overfitting training data set is a very well-known problem [7]. Overfitting in machine learning algorithms occurs, because a training data set usually does not cover data space fully. For example, assume that we have a data set that has 10 attributes and each attribute can have 10 discrete values. So the possible data space is 10^{10} = 10,000,000,000. If we have one instance for each data point in the 10 dimensional data space, we have 10 billion objects. Assuming each object occupies four bytes, the size becomes 40GB. As we see from this even simple example, the training data from real world usually occupy very small portion of their data space.

Therefore, if we apply the rough set-based rule discovery method directly to real world data sets, we may not get such good results as we expected, especially the size of data set is small compared to the domain of the data set, and the data set values are very specific for some attributes values. In other words, the data values are subdivided very much. In order to prove our assertion we'll perform an experiment with a real world data set. For our experiment, a data set called 'zoo' from UCI machine learning depository [8] is used. Zoo data has 17 conditional attributes and one decision attribute. The decision attribute has 7 different class values which classify animals in a zoo. The total number of instances is 101. Table 5 shows the meaning of each attribute.

Table 5. The attributes of data set 'zoo'.

No.	attribute	domain
1	Animal name	Unique for each instance
2	Hair	Boolean
3	Feathers	Boolean
4	Eggs	Boolean
5	Milk	Boolean
6	Airbone	Boolean
7	Aquatic	Boolean
8	Predator	Boolean
9	Toothed	Boolean
10	Backbone	Boolean
11	Breathes	Boolean
12	Venemous	Boolean
13	Fins	Boolean
14	Legs	Numeric (0, 2, 4, 5, 6, 8)
15	Tail	Boolean
16	Domestic	Boolean
17	Catsize	Boolean

In order to generate rough set-based rules for the data set MODLEM [9] algorithm was used, and test was based on 10-fold cross validation. MODLEM is a standard method that can find rules based on rough set theory.

MODLEM generated 7 rules with the accuracy of 42.6% as follows:

Rule 1. If milk in {1} Then class = 1 (41/41, 100%)

Rule 2. If name in {chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren} Then class = 2 (20/20, 100%)

Rule 3. If name in {pitviper, seasnake, slowworm, tortoise, tuatara} Then class = 3 (5/5, 100%)

Rule 4. If name in {bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna} Then class = 4 (13/13, 100%)

Rule 5. If name in {frog, newt, toad} Then class = 5 (4/4, 100%)

Rule 6. If name in {flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp} Then class = 6 (8/8, 100%)

Rule 7. If name in {clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm} Then class = 7 (10/10, 100%)

The percentage in parentheses shows the confidence of the rule, and the fraction represents the number of classified instances over the number instances having the same condition part. As we see in the rules, rough set-based method found rules precisely and rules are very accurate. But, the test result is not good.

Even though MODLEN can handle continuous values, because association rule algorithms cannot deal with continuous values, discretization method by Fayyad et al. [11] was applied for later application of the association rule algorithms. MODLEM generated the same 7 rules and with the same accuracy for the discretized data also like the original data set.

In order to find association rules class association rule algorithm [11] was used, because the data set has several conditional attributes and a decision attribute. Table 6 shows the large itemsets when minimum support is 10.

Table 6. Large itemsets.

itemset	support
hair=1 feather=0 eggs=0 milk=1 airborne=0 aquatic=0 predator=1 toothed=1 backbone=1 breathes=1 venomous=0 fins=0 legs='(3-4.5]' tail=1 domestic=0	12
hair=1 feather=0 eggs=0 milk=1 airborne=0 aquatic=0 predator=1 toothed=1 backbone=1 breathes=1 venomous=0 fins=0 legs='(3-4.5]' tail=1 catsize=1	11
hair=1 feather=0 eggs=0 milk=1 airborne=0 aquatic=0 predator=1 toothed=1 backbone=1 breathes=1 venomous=0 fins=0 legs='(3-4.5]' domestic=0 catsize=1	12
hair=1 feather=0 eggs=0 milk=1 airborne=0 aquatic=0 toothed=1 backbone=1 breathes=1 venomous=0 fins=0 legs='(3-4.5]' tail=1 domestic=0 catsize=1	16
hair=1 feather=0 eggs=0 milk=1 airborne=0 predator=1 toothed=1 backbone=1 breathes=1 venomous=0 fins=0 legs='(3-4.5]' tail=1 domestic=0 catsize=1	11

According to the large itemsets in table 6, the collection of items in the found large itemsets is { hair, Feather, eggs, milk, airborne, aquatic, toothed, backbone. breathes, venomous, fins, legs, tail, catsize}. MODLEM algorithm was applied using these attributes only to see the effect of attribute selection. Eight rules with the accuracy of 96.0% were generated as follows:

Rule 1. If milk in {1} Then class = 1 (41/41, 100%)

Rule 2. If feather in {1} Then class = 2 (20/20, 100%)

Rule 3. If toothed in {1} AND fins in {0} AND legs in {'(-inf-1]'} Then class = 3 (3/3, 60%)

Rule 4. If legs in {'(3-4.5]'} AND hair in {0} AND aquatic in {0} Then class = 3 (2/2, 40%)

Rule 5. If fins in {1} AND eggs in {1} Then class = 4 (13/13, 100%)

Rule 6. If aquatic in {1} AND legs in {'(3-4.5]'} AND hair in {0} AND toothed in {1} Then class = 5 (4/4, 100%)

Rule 7. If legs in {'(4.5-inf)'} AND aquatic in {0} AND eggs in {1} Then class = 6 (8/8, 100%)

Rule 8. If backbone in {0} AND airborne in {0} Then class = 7 (10/12, 100%)

Comparing the previous two results from the original data set and the modified data set by selecting the 15 attributes only, we doubt that the attribute 'name' might have negative effect for accuracy, because the attribute occurs very often in the rule set from the original data set. After omitting 'name' attribute, and we ran MODLEM again, resulting in 10 rules with accuracy of 94.1% as follows:

Rule 1.If milk in {1} Then class = 1 (41/41, 100%)

Rule 2. If feather in {1} Then class = 2 (20/20, 100%)

Rule 3. If toothed in {1} AND fins in {0} AND legs in {'(-inf-1]'} Then class = 3 (3/3, 60%)

Rule 4. If legs in {'(3-4.5]'} AND hair in {0} AND aquatic in {0} Then class = 3 (2/2, 40%)

Rule 5. If fins in {1} AND eggs in {1} Then class = 4 (13/13, 100%)

Rule 6. If aquatic in {1} AND legs in {'(3-4.5]'} AND hair in {0} AND toothed in {1} Then class = 5 (4/4, 100%)

Rule 7. If legs in {'(4.5-inf)'} AND predator in {0} Then class = 6 (7/7, 87.5%)

Rule 8. If legs in {'(4.5-inf)'} AND airborne in {1} Then class = 6 (6/6, 75%)

Rule 9. If backbone in {0} AND airborne in {0} AND predator in {1} Then class = 7 (8/8, 80%)

Rule 10. If backbone in {0} AND legs in {'(-inf-1]'} Then class = 7 (4/4, 40%)

When we removed attribute 'predator' only that is the other attribute not having been selected, MODLEM generated the same result with the original data discretized and un-discretized as well. The above experiments prove that our assertion is true and show the property of overfitting in rough set theory-based rule generation method.

One more experiment after eliminating the two attributes, 'name' and 'predator' in the original data set without discretization was performed, and we found similar result with accuracy of 96.0% by MODLEM, and the found 8 rules have slight different shape with the one after discretization as follows:

Rule 1. If milk in {1} Then class = 1 (41/41, 100%)
Rule 2. If feather in {1} Then class = 2 (20/20, 100%)
Rule 3. If toothed in {1} AND fins in {0} AND legs < 1Then class = 3 (3/3, 60%)
Rule 4. If legs >= 3 AND tail in {1} AND eggs in {1} AND aquatic in {0} Then class = 3 (2/2, 40%)
Rule 5. If fins in {1} AND eggs in {1} Then class = 4 (13/13, 100%)
Rule 6. If aquatic in {1} AND legs >= 3 AND toothed in {1} AND hair in {0} Then class = 5 (4/4, 100%)
Rule 7. If legs >= 5.5 AND aquatic in {0} AND eggs in {1} Then class = 6 (8/8, 100%)
Rule 8. If backbone in {0} AND airborne in {0} Then class = 7 (10/12, 100%)

Our method is also effective for a representative rule learner like RIPPER [12] as we can see in table 17. The experiment was also based on 10-fold cross validation

Table 7. The result of RIPPER.

Data set	**Accuracy (%)**	**Number of rules**
The original	88.1	6
Data set whose 'name' and 'predator' attribute have been removed	91.1	7

Table 8 summarizes all the previous experiments using MODLEM for the data set.

Table 8. The summary of the experiments with rough set algorithm.

	Data set	**Accuracy (%)**	**Number of rules**
No discretization	The original	42.6	7
	Data set whose 'name' and 'predator' attribute have been removed	96.0	8
With discretization	The original	42.6	7
	Data set whose 'predator' attribute have been removed	42.6	7
	Data set whose 'name' attribute have been removed	94.1	10
	Data set whose 'name' and 'predator' attribute have been removed	96.0	8

3 Conclusions

Rough sets are one of widely accepted machine learning technologies because of their strong mathematical background and capability of finding optimal rules based on given data sets. Good property of rough set theory is that it can describe uncertain facts solely based on data. As a result, there is no room for prejudiced views to be inserted in the found knowledge from the data. But, because rough set theory-based algorithms find rules very thoroughly, it may confront with the overfitting problem. Overfitting may occur because the theory find rules very precisely, but, on the other hand, it is not easy to find a training data set that covers its data space fully, even for big data.

On the other hand, association rules find rules of association, where the association resides between sets of items in database. Association rule algorithms find itemsets that occur more than given minimum support. The algorithms can find the itemsets practically in reasonable time even for very large databases by supplying the minimum support appropriately.

In order to overcome the problem of overfitting problem in rough set theory-based algorithms, we find large itemsets using class association rule algorithm, then we select attributes that cover the large itemsets. By using the selected attributes only, we may find better set of rules in accuracy. Various experiments have been done using a public data set called ‘zoo’ in UCI machine learning repository to support our suggested method.

This work was supported by Dongseo University, “Dongseo Frontier Project” Research Fund of 2016.

References

1. Z. Pawlak, Int. J. of Parallel Programming, **11**, 5, 341-356 (1982)
2. M. Bal, Inf. Sci. Lett., **2**, 1, 35-47 (2013)
3. J. Stefanowski, *Rough Sets in Knowledge Discovery 1: Methodology and Application,* 500-529. (Heidelberg: Physica-Verlag, 1998)
4. R. Agrawal, T. Imieliński, A. Swami, *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93,* 207-216 (1993)
5. J. Hipp, U. Gụntzer, G. Nakhaeizadeh, ACM SIGKDD Explorations Newsletter, **2**, 58, 58-64 (2000)
6. M. Hasher, http://michael.hahsler.net/research/association_rules/measures.html (2015)
7. D.J. Leiwever, The J. of Investing, **16**, 15-22 (2007)
8. A. Frank, A. Suncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Sciences (2010)
9. J. Stefanowski, *6th European Congress on Intelligent Techniques and Soft Computing*, 109-113 (1998)
10. U. M. Fayyad, K.B. Irani, *Thirteenth International Joint Conference on Articial Intelligence*, 1022-1027 (1993)
11. L.T.T. Nguyen, B. Vo, T. Hong, H. Thanh, Expert Systems with Applications, **40**, 6, 2305-2311 (2013)
12. T. Lehr, J. Yuan, D. Zeumer, S. Jayadev, M.D. Ritchie, BioData Mining, **4**, 4, doi : 10.1186/1756-0381-4-4 (2011)