International Journal of Hybrid Information Technology
Vol.8, No.11 (2015), pp.363-374
http://dx.doi.org/10.14257/ijhit.2015.8.11.32

Improved Multi-relational Decision Tree Classification Algorithm

Juan Li

School of Computer Engineering, Jinling Institute of Technology, Nanjing, 211169, China
iamlj6@jit.edu.cn

Abstract

For multi-relational data mining, efficiency is always a focus of research. The main bottleneck to improve the efficiency of the algorithm is hypothesis space. This paper presents the improved multi-relational decision tree algorithm, MRDTL-2, whose efficiency is improved. First, the tuple ID propogation technology is applied to the multi-relational decision tree algorithm. Secondly, under user's guide, when a data item is greater than the transmitting threshold, set the null relationship Ra. And transmit the primary key, the background attributes, the class label to Ra, then Ra involves in other multi-relational decision tree algorithms instead of the background relations. Finally, the paper has carried on the experiments to verify the improved multi-relational decision tree algorithm MRDTL-2.

Keywords: *Data mining, multi-relational data mining, Decision tree, Tuple ID propogation*

1. Introduction

Data mining [1] is a new field, which extracts the implicit, unknown, non-trivial and potential application valuable information from large database or data warehouse. There are a lot of data mining technologies in the traditional data mining. But with the expansion of processing object range of data mining technology, the classic learning methods have a certain limitation. Multiple relational data mining [2] has been an emerging research hotspots in recent years. It can find the complex patterns involving multiple relations from complex structured data.

In 1990, the FOIL system [3] developed by Quinlan et al. can automatically generate classification rules to classify relational data. This is also the earliest relational data classification system. In 1998, Blockeel and De Readt put forward the multi-relational data mining algorithm TILDE [4], that updates the decision tree induction method C4.5. The inductive multi-relational decision tree and propositional decision tree are identical on the structure. The internal nodes contain test, and the leaf nodes contain predicted values.

In multi-relational decision trees, use the selection map to represent the nodes of decision trees. In 1999, Knobbe *et al.*, proposed MRDTL [5] (Multi-relational decision tree algorithm), which has improved TILDE algorithm proposed by Blockeel. MRDTL and TILDE have the same mentality in the aspects of determiners and the reasoning process of trees. However, TILDE algorithm uses the first-order predicate to represent nodes in the decision tree. MRDTL uses the select map to represent nodes in the decision tree, that improves the defect of TILDE algorithm, so as to handle Tables in relational databases.

Later, S. Ruggieri puts forward the improved algorithm of C4.5, namely EC4.5 (Efficient C4.5) [6]. The results show that when producing the same decision tree, the efficiency of EC4.5 is six times bigger than that of C4.5, but EC4.5 takes up more memory than C4.5 [7]. In 2003, C. Olaru came up with a new fuzzy decision tree

ISSN: 1738-9968 IJHIT

classification method, i.e. soft decision tree [8]. Soft decision tree synthesizes the generating and pruning of soft decision tree to decide its own structure. And use rehabilitation and running-in to improve the tree induction ability. Therefore, the accuracy of the soft decision tree is higher than that of the general decision tree. In 2003, Saso Dzerroski summarized and elaborated the main theories and research contents of MRDTL method [9].

In recent years, there are many major achievements [10, 11] in relational data mining. Typically, Yin Xiaoxin [12] proposed CrossMine, which is the most representative in recent multi-relational classification method studies. CrossMine mixed ILP technology with relational database system, that effectively improved the FOIL efficiency of the traditional ILP method. In addition, it improved the technology efficiency by ID propagation technology.

In 2007, Huo Zheng, *et al.*, [13, 14] put forward the classification algorithm for relational data with user's guidance by expanding Naïve Bayesian classification algorithm. It improves the classification accuracy, and can directly support the relational database. The running time is far less than that of the relational data classification algorithm based on ILP technology.

For multi-relational data mining algorithms, the main bottleneck to improve the efficiency of the algorithm is hypothesis space all the time. A lot of attributes that have nothing to do with the classification are stored in the background knowledge relations. The search space becomes larger, leading to low classification efficiency. So in order to improve the efficiency of multi-relational data mining algorithm, the key is to reduce the hypothesis space. To solve the above problems, this paper presents the multi-relational decision tree classification algorithm, namely MRDTL-2 algorithm. And the main improvement is that under user's guide, apply tuple propagation technology into the multi-relational decision tree algorithm. The second part of the paper introduces the ideas of the improved algorithm. The third part is the concrete implementation of the improved algorithm, including the necessary basic theories. The fourth part conducts experiments and analysis. The advantage of the improved algorithm is proved in this section. Finally, the fifth part makes a conclusion for the full paper.

2. The Thought of the Improved Algorithm

The following database mentioned is shown in Figure 1.

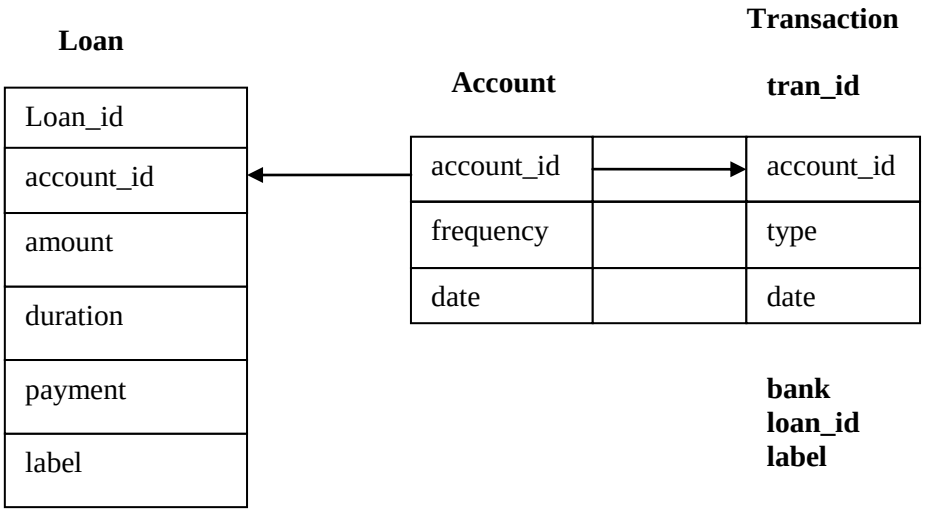

Figure 1. Database

Table 1. Loan

Loan					
Loan_id	account_id	amount	duration	payment	label
1	124	1000	12	120	+
2	124	4000	12	350	+
3	108	10000	24	500	-
4	45	12000	36	400	-
5	45	2000	24	90	+

Table 2. Account

Account			
account_id	frequency	…	date
124	monthly	…	960227
108	weekly	…	950923
45	monthly	…	941209
67	weekly	…	950101

Table 3. Transaction

Transcation						
tran_id	account_id	type	date	bank	loan_id	label
1	124	Pruem	950302	LJ	1,2	2+,0-
2	124	Vardy	950324	OP	3	0+,1-
3	108	Pruem	950523	YZ	4,5	1+,1-
4	108	Darey	940415	EF	6,7	1+,1-
5	136	vardy	960102	UV	8	1+

Multi-relational data mining algorithm is to find the complex patterns involving multiple relations in relational database. The database is composed of the relation sets, one of which is the target relation with the class identifier associated with other tuples. Other relations are non-target relations. Each relation has a primary key and a few foreign primary keys that point to the primary keys of other relations.

The traditional multi-relational data mining is very easy to cause the loss of semantics. If connect multiple relations directly and conduct data mining works, data mining faces a very large amount of data. The efficiency is so low and the scalability is poor. While, in multi relational data mining tuple ID propagation technology, the most important data mining technology, has solved this problem. Tuple propagation technology can not only keep all data semantic meanings unchanged, improving efficiency, and having good scalability.

The core idea of tuple ID propagation technology is that the data is still stored in the database, not like the traditional ILP technology to transform into a logic program. It assumes that the representation language is still using the first-order predicate logic. The relationship only conducts one connection operation between each other in the database, without multiple connections and the materialization of the join results.

In MRDTL-2 algorithm only consider a few kinds of connections: one is the connection that the primary key K points to the foreign key connection of K; the other is the connection of foreign key K1 and K2 which all point to the same primary key K. Because the other possible connections do not represent strong semantic links of entities in database, ignore other connections.

MRDTL-2 algorithm mainly includes two cases:

The first case, under user's guide, when a background relational data item is smaller than or equal to the transmitting threshold p, in this case, it is also divided into the tuple

propagation between target relation and non-target relation and the tuple propagation between non-targets.

The tuple propagation between target relation and non-target relation: assuming the target relation R1 (the primary key is R1.id) and non-target relation can be connected through the attributes R1.A and R2.A, transmit R1.id and the class label from the target Table R1 to R2. For each record u in relation R2, idset (u) denotes all records linked to u in the target Table.

For example, as shown in Figure 1, the target relation Loan (Loan_id) and the non-target relation Account can be connected through attributes Loan.account_id and Account.account_id, then transmit Loan.Loan_id and Loan.label to relation Account. For each record u in Account, idset(u) denotes all records having a contact with u in the target Table.

The tuple propagation between non-target relations: assuming non-target relations R2 and R3 can be connected through the attributes R2.A and R3.A, for each tuple v in R2, idset (v) denotes the set of target tuples connected with v. According to the connection condition of R2.A=R3.A, transmit the target tuple ID and class label from the relation R2 to the relation R3. For each tuple v in R3, idset (v) denotes the tuple set that can be connected to v.

For example, non-target relation Transaction （Tran_id） and non-target relation Account can be connected through the attribute Transaction.account_id and Account.account_id,then transmit Transaction.Loan_id and Transaction.label to Account. For each record v in Account, idset(v) denotes all records having a contact with v in the target Table.

The second case: under user's guide, when a data item in background relation is bigger than the transmitting threshold P, set null relationship Ra. Under user's guide, transmit the primary key, the background attributes, the class label in the target relation to Ra, then Ra involves in other multi-relational decision tree algorithms instead of the background relations. This case also includes the tuple propagation between target relation and non-target relation and the tuple propagation between non-targets. The tuple propagation between target relation and non-target relation:

Assume that the target relation R1 (the primary key is R1.id) and the background relation R2 specified by users of non-target relation can be connected through the attributes, R1.A and R2.A. And when the data item in background relation R2 is greater than the transmitting threshold, transmit the primary key in R2, background attributes to the null relation Ra2. Then transmit R1.id and class label from the target Table R1 to Ra2. For each record u in relation Ra2, idset (u) denotes all records linked to u in the target Table.

3. The Concrete Implementation Process of MRDTL-2 Algorithm

3.1. Basic Theory

The calculation of information gain is the most important part in the algorithm. First of all, introduce the definitions and formulas used in information gain calculation.

Definition 1: the amount of information required in classification. Suppose that S is a set containing s samples. The category attributes can take m different values corresponding to m different classifications Ci (i is a integer greater than zero). If Si is the number of sample in class Ci, The amount of information needed to classify a given data object is shown as formula (1).

$$I(s1, s2, \cdots, sm) = -\sum pi \log(pi) \quad i \in [1, m] \tag{1}$$

In the above formula, pi=Si/S is the probability of any data sample belonging to the category Ci in subset Si.

Definition 2: information entropy. Suppose that a attribute A can take v different values (a1,a2,…,av). Using attribute A can divide set S into v subsets (s1,s2,…,sv), in which si contains the data samples with ai values of attribute A in set S. If the attribute is chosen as the test attribute, set sij as the sample set of class Ci in subset Sj. The information entropy to divide the current sample set using attribute A is showed as formula (2).

$$E(A) = \sum (s1j + s2j + \cdots smj) \Big/ s * I(s1j, s2j, \cdots, smj) \qquad j \in [1, v] \tag{2}$$

Definition 3: Information gain. The obtained information gain to carry on the corresponding sample set partitioning of current branch nodes using attribute A is showed as formula (3).

$$Gain(A) = I(s1, s2, \cdots, sm) * E(A) \tag{3}$$

For the rule r, use the P (r) and N (r) said meet r the number of positive and negative sample. Assume that the current rules for r, r + p said to r p built after optimizing operation rules, optimize operation p foil gain are defined as follows:

Definition 4: foil gain. For the rule r, P(r) and N(r) expresses the number of positive and negative samples meeting r. Assume that the current rule is r, and the established rule after adding the optimized operation is expressed by r+p. The foil gain of the optimized operation P is defined as follows:

$$I(r) = -\log P(r) \Big/ (P(r) + N(r))$$

$$foil\ gain(p) = p(r+p) * [I(r) - I(r+p)]$$

Among them, p(r+p),N(r+p) and the foil gain of p can be calculated by the search result, Seleet attribute_list from Table_list where join_list and condition_list.

Definition 5: background relation. In a database D, for a classification task T specified by users, the relation Table which is related to the classification task and the target relation Table is called the background relation, *i.e.*, {R1,R2,…,Rm}.

In the Figure 1, if users specify Table Loan as the target relation Table, relation Table Account is the background knowledge Table. If users specify Table Account as the target relation Table, relation Table Loan is the background knowledge Table. So the target relation Table and the background knowledge Table are relative.

Definition 6: background attributes. In the classification task, attributes {A1, AZ,…,An}, which are related to classification and specified by users in the background knowledge Table are called background attributes.

Definition 7: data items. In relations the product of the number of attribute and the number of record is called data item.

Definition 8: transmitting threshold P. When a data item is bigger than a certain value P in a certain relation, then in the optimization operation process, transfer the background attribute, the tuple ID and class label to the null relation for searching. This searching efficiency is far smaller than the search efficiency using tuple ID propogation technology directly. The value P is called the transmitting threshold.

Definition 9: Background attribute transfer. It assumes that there is a null relation set {Ral,RaZ,…,Ram} corresponding to the background relation { Rl,R2,…,Rm} . If the data item in the background relation Rt is bigger than P, transfer the background attributes and the primary key of Rt to Rat. Exchange the relation name between Rt and Rat. In the following optimization process, continue the next series of operations with Rt as the background relation.

Theorem 1: it assumes that relation R1 and R2 can be connected through attribute R1.A and R2.A. Thereinto, R1 is the target relation, whose primary key is R1.id. R2 is the background relation specified by users. And when the data item in R2 is bigger than the transmitting threshold, transmit the primary key and the background attributes of R2 to

Ra2, then transfer to the record ID in Ra2 from the target Table R1. For each record u in relation Ra2 , idset(u) denotes all records connected to u in the target Table.

Proof: according to the above definitions, $idset(u) = Ut \in R1, t.A = u.Aidset(t)$, *i.e.*, idset(u) shows all records connected to u in the target Table based on the connection path specified by the current rules.

Inference: it assumes that relation R1 and relation R2 can be connected through attribute R1.A and R2.A. R1 is the target relation, in which all tuples satisfies the current rules. R2 is the background relation specified by users. And when the data item in R2 is bigger than the transmitting threshold, transmit the primary key and the background attributes of R2 to Ra2. If the label in R1 is propagated to Ra2, the foil gain of each optimization operation in Ra2 can be calculated through using the labels propagated to Ra2.

Proof: it assumes that there is a current rule r, for the candidate optimization operation P in Ra2, for example, Ra2.B=b, the foil gains of P(r), N(r), P(r+p) and v(r+p) can be calculated as follows:

(1) All tuples that satisfy the condition of Ra2.B=b in Ra2.
(2) According to the labels of the target tuples transmitted in Ra2, find the target tuples which can connect with the tuples in (1).
(3) Calculate foil gain based on (2).

Theorem 2: it supposes that non-target relations R2 and R3 can be connected by attributes R2.A and R3.A. All tuples in R2 satisfy the current rules, and R2 is the background relation specified by users. When the data item in R2 is larger than the transmitting threshold, transmit the primary key and the background attributes in R2 to Ra2. According to the condition of Ra2.A=R3.A, transmit the target tuple labels from R3 to Ra2. For each tuple v in Ra2, idset(v) denotes the target tuple set that can connect with v (using the connection path of the current rules and Ra2.A=R3.A).

Proof: it assumes that the tuples in R3 can connect with the tuples v1,v2,…,vm in Ra2 according to the connection condition Ra2.A=R3.A. Then $idset(v) = \cup_{i=1}^{m} idset(vi)$. If and only if the target tuple t satisfies the condition of $t.id \in \bigcup_{i=1}^{m} idset(vi)$,. the target tuple t connects with one of v1,v2,…,vm. So, if and only if $t.id \in idset(v)$, the target tuple t can connect with v (using the connection path of the current rules and Ra2.A=R3.A).

It assumes that the current rule is $r = R1(L,+) : -R1(L,A,?,?,?,?), Ra2(A,?,monthly,?),$,in which R1 is the target relation Loan and R2 is the relation obtained after the treatment of finding the data items greater than the transmitting threshold in relation Account. For each forecast $"Ra2(A,?,monthly,?)"$, tuple{124, 45} which satisfies this prediction can be found in Ra2. After that, in the target relation R1, it can find the tuple {1,2,4,5} that connects with the two tuples, so that acquire the label of the target tuple. There are three "+" and a "-" in tuple {1,2,4,5}. It is easy to calculate the foil gain of $Ra2(A,?,monthly,?)$ by using such information.

3.2. The Implementation of MRDTL-2 Algorithm

The pseudo codes of the improved multi-relational decision tree algorithm are showed as follows:

Input: database D, the target relation Ri, the set of the background attributes.
Output: binary decision tree.
Begin Tree_Growing(L).
Set Ri as active;

Root=L;/* Take the learning sample set as the root node corresponding to all records in the target Table */

Call Procedure Split_to_tl_tr(root);/* Call branch process to divide a node into two child nodes */

End Tree_Growing;

Split_to_tl_tr(t) algorithm takes advantage of stack to realize the recursion operation in the process of establishing the decision tree. The end condition of the recursion operation is when judging the stack is empty, end running, and output the results of last mining, namely the decision tree.

At the beginning, the program takes the learning sample set as the root nodes, corresponding to all the records in the target Table, and assign a value to t, into the stack. If the stack is not empty, judge that whether the current case satisfies the following conditions: the number of sample in node t is smaller than or equal to Nmin, all samples in node t belong to the same kind, or there is no optimization operation to branch node t. Thereinto, Nmin is the results through many experiments, i.e., if the number of sample is smaller than this value, classify this sample, then the final data mining results of multi-relational decision tree will be no meaning.

If node t satisfies any of the above conditions, it will produce the leaf node by using this algorithm. Then continue to conduct the judging empty operation for stack. If node t dose not satisfy any condition, produce the left and right subtree after a series of operations and press them into the stack.

/* The process for node t to be branched into the left subtree and the right subtree */

Begin Split_to_tl_tr(t).

best_gain=0;

Ptest=empty; /*Ptest is the best branch rules*/

If the number of sample in node t is smaller than or equal to Nmin, all samples in node t belong to the same kind, or there is no optimization operation to branch node t.

Then return leaf;

Else

For each aetive relation Ra

For each candidate predicate P

If foil_gain(P)>best_gain

then best_gain=foil_gain(P);

Ptest=P;

Endif

End for

End for

For each key/foreign_key K of Ri/*Ri is the non-active relation which can connect with Ra through the foreign key Ri.K*/

If Ri can connect with an activity relation using Ri.K.

Calculate p as the data item of the background relation.

If p is greater than P, the transmitting threshold of the background attributes, then transmit the background attributes and the primary keys from Ri to Ai.

/*Ai is the null relation of the corresponding back-up Ri, the following is the optimization operation of Ai instead of Ri.*/

Transmit the target tuple labels from Ra to Ai;

Else

Transmit the target tuple labels from Ra to Ri;

Endif

Endif

End for

If the optimization operation is P of bordering, *i.e.*, Ra.K=Ri.K and P is larger than the background attribute transmitting threshold P.

Then set Ai as active;
Else
Set Ri as active;
Endif

Record the optimization operation Ptest corresponding to the optimization branch rules.

Tl is the left subtree produced under the action of Ptest.

Tr is the right subtree produced under the action of Ptest.

Call Procedure Split_to_tl_tr(tl);

Call Proeedure Split_to_tl_tr(tr);

Endif

End Split_to_tl_tr

4. Experiment Analysis

4.1. The Experimental Running Environment

Environment settings to verify MRDTL - 2 algorithm are showed as follows.

Operating system: Windows XP Professional. Internal memory: 1G. Hard disk:80G. CPU: Genuine Intel(R) 1.73GHZ. Programming language and database: Visual Foxpro 6.0.

The comparative tests between MRDTL and MRDTL-2 algorithms are conducted in the actual database PKDD CUP '99((as shown in Figure 2).).

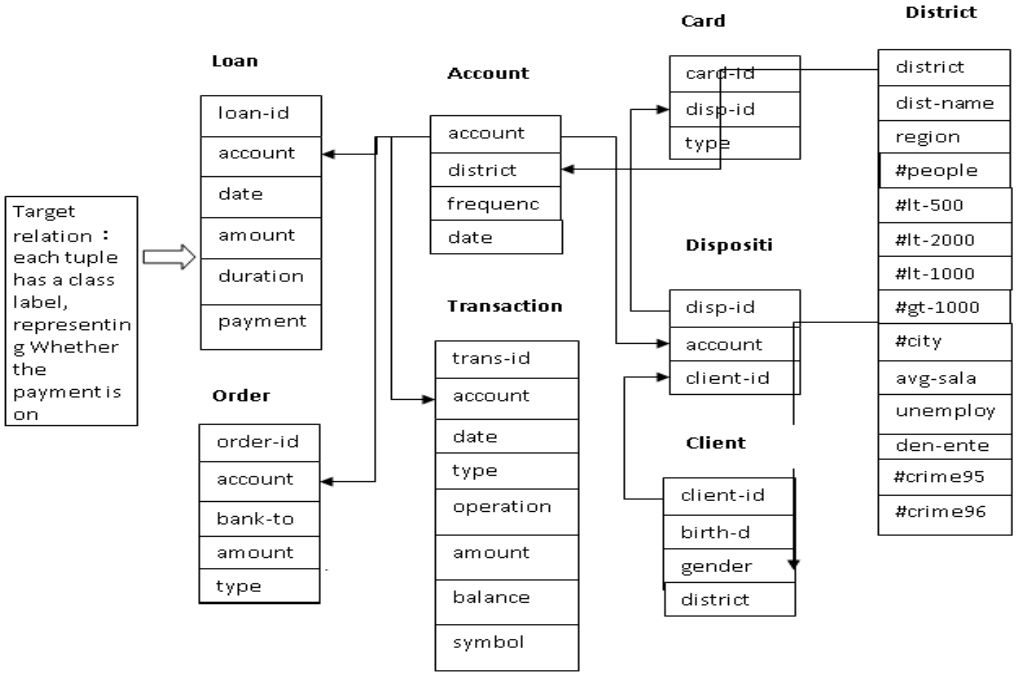

Figure 2. PKDD CUP'99 Financial Database

4.2. The Experimental Data

In the experiment, we select a part of financial data in PKDD CUP'99 as the test data. And data in relation Loan, Account, Transaction are chosen as the test data (as shown in Figure 2). The test data includes 1061502 records.

The target relation Loan includes 682 records. Statusl is the class label, which means failing to pay for loans on schedule. In the original relation, there are A,B,C,D four status attributes in Loan relation. As a result of the need of experiment, carry out simple processing of status. Update values A, B with "P", which means being able to pay for loans. And update values C, D with "N".

The background relation Account contains 4500 records. The primary key of relation Account is account_id. It can establish contact with relation Loan and relation Transaction through this keyword respectively. district_id is its foreign primary key. The background attributes specified by users are frequency attributes.

The background relation Transaction contains 1026320 records. The primary key of relation Account is trans_id. account_id is the foreign primary key. The background attributes specified by users are type and account attributes.

4.3. The Experiment Process and Results

In MRDTL-2 algorithm, the following two methods are adopted to complete the experiments.

The experiment process of the first method: fix the recording number of three relations unchanged.The original Loan contains 6 attributes; Account contains 4 attributes; Transaction contains 10 attributes. 10,20,25,30,40,50 attributes are added to each relation respectively for experiments. Thereinto, the number of background attributes relevant to classification in two relations Account and Transaction is 10. In order to prevent the abnormal operation time, we perform 10 experiments. In order to prevent the abnormal operation time, we performed 10 experiments. The experiment results are obtained by average of 10 cross validations, as shown in Table 4.

Table 4. The Running Time Comparison Table of MRDTL and MRDTL-2 Along with the Change of Attribute Number (Unit: Second)

Algorithm / The number of attributes increased	MRDTL	MRDTL-2
0	7.54	27.56
10	12.36	29.21
20	18.32	30.07
25	39.01	31.46
30	41.78	31.27
40	61.69	35.04
50	79.51	38.18

The experiment process of the second method: fix the number of attributes in three relations unchanged. Loan contains 50 attributes; Account contains 58 attributes; Transaction contains 76 attributes. Thereinto, the number of background attributes relevant to classification in two relations Account and Transaction is 10.

After some experiments, it is found that the records obtained after sorting the background attribute amount in the background relation Transaction have little influence on the experiment results. Therefore, using " select * top 10500000 from transaction order by amount into Table trans " to acquire 1050,800,500,200, 100 thousand records of relation Transaction for experiments. In order to prevent the abnormal operation time, the experiment results are obtained by average of 10 cross validations, as shown in Table 5.

Table 5. The Running Time Comparison Table of MRDTL and MRDTL-2 Along with the Change of Record Number (Unit: Second)

Algorithm / The number of attributes increased	MRDTL	MRDTL-2
10	1.24	7.87
20	9.76	15.56
50	42.13	29.60
80	65.27	36.92
105	80.63	39.14

4.4. Experiment Analysis

Through the above experiment results, we can find that when MRDTL-2 satisfies certain conditions, its running time is much smaller than that of MRDTL, without rapidly increasing with the increase of the number of attributes and the number of records.

The first experiment is the time comparison result between two algorithms along with the increase of the number of attributes in case of the fixed number of records. According to the Table 4, when the number of attributes is small, the running time of MRDTL algorithm is smaller than that of MRDTL-2. And with the increase of the number of attributes, the running times of MRDTL and MRDTL-2 all increase gradually. In the end, the running times of two algorithms are the same at a certain number of attributes. When it is greater than this value, the running time of MRDTL-2 algorithm increases slowly, while the running time of MRDTL-2 algorithm grows up quickly. What's more, the running time of MRDTL-2 algorithm is bigger than that of, MRDTL. It assumes that the data item is P when MRDTL and MRDTL-2 run with the same time. When the data item of algorithm is smaller than P, the efficiency of MRDTL-2 is low. At this moment, the running efficiency of MRDTL is high. When it is equal to P, the efficiencies of two algorithms are the same. When the data item is bigger than P, the efficiency of MRDTL-2 raises greatly, and the running time is not affected by the increase of attributes. While the running efficiency of MRDTL is low and is affected greatly.

The second experiment is under the condition of the fixed attribute number and the increasing record number, as shown in the Table 5. It can be concluded from the Table that when the number of records is small, the running time of MRDTL algorithm is smaller than that of MRDTL-2. And with the increase of the number of record, the running times of MRDTL and MRDTL-2 all increase gradually. Finally, the running times of two algorithms are the same at a certain number of record, which is denoted by Q. When the data item of algorithm is smaller than Q, the efficiency of MRDTL-2 is lower. While, the running efficiency of MRDTL is higher. When the data item of algorithm is equal to Q, the efficiencies of two algorithms are the same. When the data item is bigger than Q, the efficiency of MRDTL-2 raises greatly, and the running time is not affected by the increase of records. While the running efficiency of MRDTL is low and is affected greatly by the number of record.

Based on the above experimental data, it can be found that if the data item of algorithm running is bigger than P(the data item in the background relation is bigger than the transmitting threshold P), the efficiency of MRDTL-2 algorithm is higher, and the running is not affected by the increase of data items. While the efficiency of MRDTL algorithm is lower, and the running time is strongly influenced by the increase of data items.

5. Conclusion

This paper put forwards a improved multi-relational decision tree algorithm, MRDTL-2, that improves the efficiency of algorithm the user's satisfaction. Firstly, it adopts multi-relational decision tree. Secondly, under user's guide, when a data item is greater than the transmitting threshold, set the null relation Ra. Transmit the primary key, the background attributes, the class label of the target relation to Ra, then Ra involves in other multi-relational decision tree algorithms instead of the background relations. Finally, the paper has carried on the experiments to verify the improved multi-relational decision tree algorithm. The experiment results show that the proposed algorithm is superior to the existing similar algorithms, achieving the anticipated goals of research. Compared with the traditional data mining algorithms, it obviously improves the user's satisfaction and the operation efficiency. In conclusion, relational data mining is a very meaningful work and the task is very arduous, expected to further study.

References

[1] D. T. Larose, "Discovering knowledge in data: an introduction to data mining", John Wiley & Sons, **(2014)**.
[2] M. Rouane-Hacene, M. Huchard and A. Napoli, "Relational concept analysis: mining concept lattices from multi-relational data", Annals of Mathematics and Artificial Intelligence, vol. 67, no. 1, **(2013)**, pp. 81-108.
[3] J. R. Quinlan, "Learning Logical Definitions from Relations", Machine Learning, vol. 5, **(1990)**, pp. 239-266.
[4] H. Blockeel, "Top-Down Induction of First Order Logical Decision Trees", Artificial Intelligence, **(1998)**, pp. 285-297.
[5] J. K. Arno, A. Siebes and D. D. Wallen, "Multi-Relational Decision Tree Induction", Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery, Prague, **(1999)**, pp. 378-383.
[6] S. Ruggieri, "Efficient C4.5", IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 2, **(2002)**, pp. 438-444.
[7] L. H. Luan and G. L. Ji, "Research on the decision tree classification technology", Computer Engineering, vol. 30, no. 9, **(2004)**, pp. 94-96.
[8] C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique", Fuzzy Sets and Systems, vol. 138, no. 2, **(2003)**, pp. 221-254.
[9] S. Dzerroski, "Multi-Relational Data Mining: An Introduction", ACM SIGKDD Explorations Newsletter, vol. 5, no. 1, **(2003)** July, pp. 1-16.
[10] Y. Kavurucu, P. Senkul and I. H. Toroslu, "Confidence-based concept discovery in multi-relational data mining", Proceedings of the International Multi Conference of Engineers and Computer Scientists, **(2008)**, pp. 1.
[11] W. Zhang, "Multi-Relational data mining based on higher-order inductive logic programming", Intelligent Systems, GCIS'09, WRI Global Congress on, IEEE, vol. 2, **(2009)**, pp. 453-458.
[12] X. Yin, J. Han, J. Yang, *et al.*, "Efficient classification across multiple database relations: A crossmine approach", Knowledge and Data Engineering, IEEE Transactions on, vol. 18, no. 6, **(2006)**, pp. 770-783.
[13] H. Zheng, "Research on the relational data classification algorithm based on background knowledges", He Bei: YanShan University, **(2007)**, pp. 3034.
[14] H. Zheng, J. F. Guo and J. Y Wang, "A relational data classification algorithm with user guide", Microelectronics & Computer, vol. 23, no. 9, **(2006)**, pp. 141-143.

Authors

Juan Li, she received the Bachelor Degree from Soochow University in 2003, the Master Degree from Nanjing University of Science and Technology in 2009. From 2009 to now, she is a lecturer at the School of Computer Engineering, Jinling Institute of Technology. Her current research interests include data mining and information management.