Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 63 (2013) 18525-18531

ISSN: 2229-712X

Multi Relational Learning (Classification) Based on Relation Data Using Weighted Voting Combination Technique

Amit Thakkar* and Y P Kosta

Department of Information Technology, Faculty of Technology & Engineering, Charotar University of Science & Technology, Charusat.

ARTICLE INFO

Article history:
Received: 22 June 2013;
Received in revised form:
29 September 2013;
Accepted: 15 October 2013;

Keywords
Data mining,
Machine learning,
Relational data.

ABSTRACT

Classification is an important task in data mining and machine learning, in which a model is generated based on training dataset and that model is used to predict class label of unknown dataset. Today most real-world data are stored in relational format. So to classify objects in one relation, other relations provide useful information. Relational data are the popular format for structured data which consist of tables connected via relations (primary key/ foreign key). Relational data are simply too complex to analyze with a propositional (single table learning) algorithm of data mining. So to classify from relational data there is a need of multi relational classification which is used to analyze relational data and predict unknown pattern automatically. This paper contains Multi Relational Classification with weighted voting algorithm for learning from relational data which result in increase accuracy. Also to decrease the running time voting technique is used compared to stacking as a combination method. The experimental study along with results demonstrate the effectiveness of algorithm respect to other existing techniques.

Introduction

Most of today's real-world data is structured: such data has no natural representation in a single tabular table and instances in these data are more naturally represented by structured terms than fixed-length feature vectors [13]. Learning from structured data needs to take the information encoded in the structure of the data into account, since such structure presents how different objects in the data relate to one another and may demonstrate some useful patterns in mining tasks.

Multirelational classification algorithms search for patterns across multiple interlinked tables in a relational data. This type of method searches for relevant features from a target relation in which each tuple is associated with a class label and relations related to the target table, in order to better classify tuples in the target relation. Today most real-world data are stored in relational format. So to classify objects in one relation, other relations provide crucial information. To classify data from relational data there is a need of multi relational learning (classification) which is used to analyze relational data and used to predict behavior and unknown pattern automatically.

Multi Relational Learning Based On Relational Data

Multi relational learning means learning from relational data. It is also called Multi relational or relational classification. Classification in data mining is a two step process as shown in fig. 1.

Step 1: To learn classification model from Training Set.

Step2: To classify Testing set using the classification model.

In multi relational classification, given a collection of tables (relational data), each table contain a set of attribute, only target table contains class attribute. As shown in figure 1, we have to find a model for class attribute based on details of other tables and classification technique. The goal of classification is to predict a class to unknown records (test set) accurately.

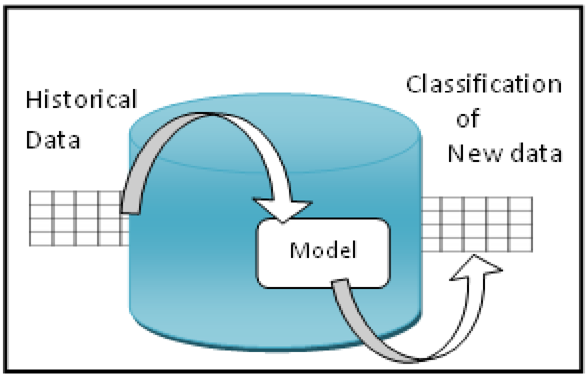

Figure 1. Classification Model

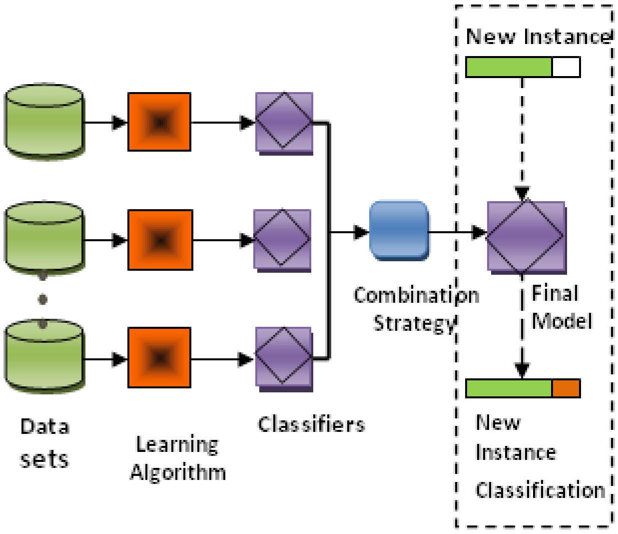

Figure 2. Classification Model for Relational Learning

Tele:
E-mail addresses: amitthakkar.it@gmail.com

Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

In multi relational classification, there is one target table that has a special role. Each record in this table corresponds to exactly one individual. The target table will be connected to other background tables through foreign key relations. By following these keys the remaining data concerning individuals may be looked up. The target table will always be the starting point for searching interesting patterns. In order to collect all parts belonging to a given individual, one will have to join over the foreign key relations. So objective of relational classification is to build a model that accurately predicts the target concept as shown in figure 2.

Related Work

We will first present the review of related work for Multi relational learning. There are many algorithms available for classification but they are applied only on single/flat file. So if we want to deal with relational data, either we have to upgrade propositional algorithms (single table learning) or we have to convert relational data into single file then apply any single table learning algorithm on it.

The first category of algorithms for multi relational domains aims to upgrade propositional methods to handle relational representations. This method extends propositional classifiers to search features in multiple relations. That is, by designing new relational learning techniques for existing conventional data mining methods.

TILDE

The TILDE method [15] employs the divide-and-conquer searching technique. The TILDE algorithm extends the popular C4.5 prepositional classifier to tackle relational representations. Research has shown that the divide-and-conquer approach embedded in the decision tree construction makes the TILDE method an efficient one, compared to many traditional ILP algorithms [15].

Cross Mine

Scalability is a big challenge to approaches using ILP techniques [16]. The scalability problem is observed since the size of the hypothesis search space increases rapidly with the number of relations. To tackle this problem, Yin et al. [16] present the Cross Mine algorithm. The authors propose an idea to implement virtual joins in a database, instead of physical joins. Through virtual join, this approach demonstrates high scalability and accuracy, as presented in [16].

The second category of strategic used to handle relational domains is the so-called propositionalization methods. Propositionalization algorithms flatten multiple relations into a universal single one. Subsequently, this flattened relation is presented to propositional classifiers. Formally, the transformation process can be defined as follows.

The general idea is to enable wide range of efficient and accurate propositional learning algorithms from the single-table learning community, to be applied to mine structured data [19].

Roll Up

Knobbe and Siebes developed the RollUp algorithm to perform the "flattening" of multiple relations in a relational database [17]. The RollUp approach employs a depth-first search (DFS) in a relational database to aggregate the information in deep-level relevant tables (i.e. tables far from the target relation) to their parent tables (i.e. tables less far from the target relation), using aggregate functions, such as count, min, sum, max, and mean, found in SQL. This summarization procedure stops if attributes from all relevant tables in the DFS search are aggregated onto the target table. The resulting universal table is then used as input by a prepositional learning algorithm. The drawbacks of this approach are that the "roll up" process may aggregate the same background table multiple times onto the target relation.

Rel Aggs

Following the same ideas as the RollUp approach, Krogel and Wrobel introduced a so-called RelAggs propositionalization method [17]. This algorithm overcomes the above-mentioned limitations of the RollUp algorithm. In the Rel Aggs strategy, sophisticated aggregate operators are used to construct new features for a single-table propositional classifier. In this way, crucial information from background relations can be represented by the newly constructed attributes in the target table. Below table represents the comparative study of both the approaches of relational learning.

Table 1: Comparative Analysis

Comparative Analysis of Upgrading Versus Flattening Strategies	Advantages	Disadvantages
Upgrade Conventional Data Mining Methods	Normalized Structure Expressive Power Little pre processing of Data	Poor scalability when dealing with complex database schemas. Poor support for noisy or numeric values in real-
Flatten multiple tables into a single flat file	Learning Using Traditional Single Table Algorithm Integration of Advance Techniques Handling real world data Dealing with Noisy values	Extensive Preprocessing of Data Transform relation into flat file Lose Normalized compact structure Result in large number of null values Generate new Attributes exponentially.

Problem Definition

For relational classification, we need relational data which consist of tables connected through primary key/foreign key relationship. Multi-relational classification can directly look for patterns that involve multiple relations from a relational data. So we can say relational data R is a collection of tables $R = \{R_1, R_2, ... R_n\}$. A table R_i consists of a set of tuples T_R and has at least one key attribute, either the primary key attribute and/or the foreign key attribute. Foreign key attributes link to key attributes of other tables. This link specifies a join between two tables. Foreign key relationship may be directed or undirected between tables. For relational classification, we have one target relation R_t and other background relations $R_{b1}, R_{b2}, ... R_{bn}$. Each tuple $x \in T_{Rt}$ includes a unique primary key attribute x.k and a categorical variable (target variable) y. The aim of relational classification is to find a function F(x) which maps each tuple x of the target relation R_t to the category y such that:

$$\mathbf{Y = F(x, R_t, R_{b1}, R_{b2}, ... R_{bn}), x \in T_R} \quad \textbf{1.1}$$

Introduction to the Proposed System

Multirelational classification aims at discovering useful patterns across multiple inter-connected tables (relations) from relational data. Multirelational classification makes use of different relation of relational data and applies propositional data mining algorithm on individual relation. In this section we proposed an algorithm called ‘Multi Relational Classification with weighted voting (MRC-WV). In MRC-WV, weighted voting is applied to relations which will be used in testing phase. Results of different relations are then combined by voting technique. The aim is to improve the predictive accuracy of algorithm.

General Process

In general, Following steps are executed to mine the data from relational database.

1. Firstly, the tuple IDs and the target concepts from the target relation are propagated to other background relations, based on the foreign links (key references) between them.
2. Secondly, aggregation functions are applied to each background relation in order to handle the one-to-many relationships. Each background relation is then used as training data for a particular relational classifier.
3. Thirdly, conventional single-table data mining methods are used in order to learn the target concept from each relation of the data separately.
4. Then weighted voting technique is used to give appropriate vote or importance to all relations as per their predictive performance using best worst weighted voting technique.
5. Lastly, using classifier combination method the result of individual classifier is combined and class label is assigned to unlabeled tuple.

This algorithm is able to use any conventional method to mine data from relational data.

Weighted Vote Combination Scheme

In this scheme, the best and the worst classifiers in the relational learning are identified using their estimated misclassification error. A maximum weight 1 is assigned to the best classifier and a minimum weight 1/c, where c is no of classes is assigned to the worst classifier if misclassification error of worst classifier is < = 1/c else weight zero is assigned. The rest of classifiers are rated linearly between these extremes as shown below.

The values for weights are calculated as follows:

$$Weight\ i = \begin{cases} \frac{(Error_w - Error_i)}{(Error_w - Error_b)} * \left(1 - \frac{1}{c}\right) + \frac{1}{c} \\ 1\ if\ \ Error_w = Error_b \\ 0\ if\ \ Error \geq \frac{1}{c} \end{cases} \quad \ldots\ldots\ldots\ldots 1.2$$

Where $Error_b$ is misclassification error of best classifier and $Error_w$ is error of worst classifier and c is number of classes.

Classifier Combination Methods

The combinations methods can be separated into mathematical and behavioral approaches [18]. The mathematical combinations try to construct models and derive combination rules using logic and statistics. The behavioral methods assume discussions between experts, and direct human involvement in the combination process. The mathematical approaches gained more attention with the development of computer expert systems.

Score Matrix can be used to visualize the combination algorithm. The output of classifiers can be represented as vectors of numbers where the dimension of vectors is equal to the number of classes. As a result, the combination problem can be defined as a problem of finding the combination function accepting N-dimensional score vectors from M classifiers and outputting N final classification scores as shown in Figure 3, where the function is optimal in some sense, e.g. minimizing the misclassification cost.

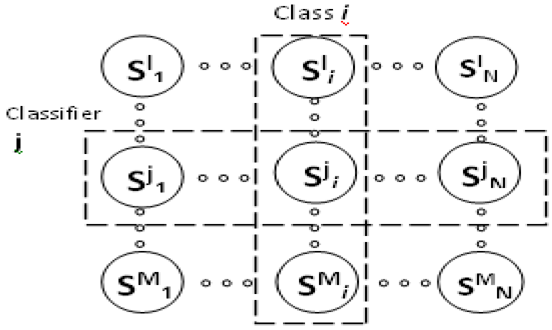

Figure 3. Score Matrix

As shown in Figure 4 Classifier combination takes a set of s_i^j - score for class i by classfier j and produces combination scores S_i for each class i.

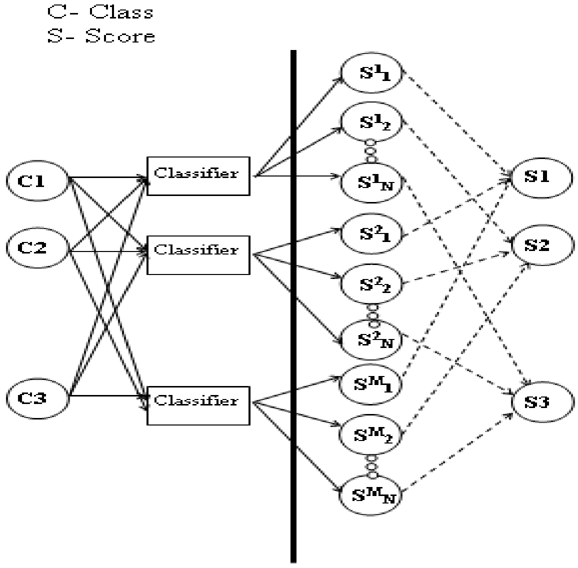

Figure 4. Classifier Combinations [22]

Example:

Classifier	Misclassification Error	Weight
C0	0.65	0
C1	0.42	0.5
C2	0.32	0.625
C3	0.20	0.775
C4	0.08	0.925
C5	0.02	1

Table 2. Weight Assignment to Classifier

Classifier	Class A	Class B
C1	0.7	0.3
C2	0.62	0.38
C3	0.1	0.9
C4	0.3	0.7
C5	0.8	0.2
∑Ci	2.52	2.48

Table 3. Score Matrix

Classifier	Class A	W_i	Class B	W_i
C1	0.7	0.5	0.3	0.5
C2	0.62	0.625	0.38	0.625
C3	0.2	0.775	0.8	0.775
C4	0.3	0.925	0.7	0.925
C5	0.82	1	0.18	1
∑WiCi	1.89		1.93	

Table 4. Final Prediction using Best Worst weighted vote technique

Based on the final result the unknown tuple is assigned the class label with highest prediction power. In the above example class label assigned is B as combined predictive capability for class B is more than Class A. Simple Majority rule assign class label A.

Final Classifier Combination Scheme

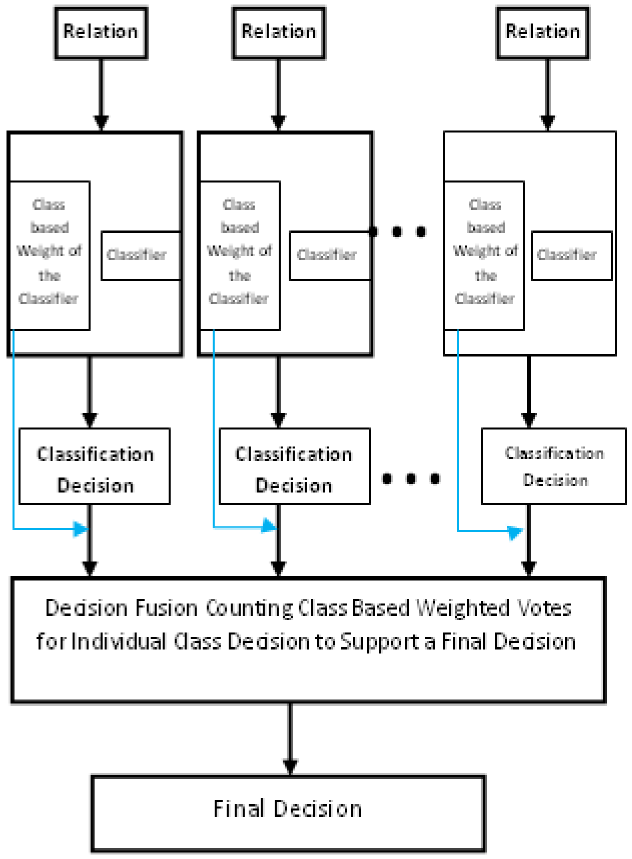

Figure 5. Combination Scheme

As shown in the figure 5 combination function takes as parameters not only the scores related to this class, but weight associated to each classifier. Combination scores for each class are calculated using the same Data Fusion Function.

Proposed Algorithm MRC-WV (Training Phase)

Input: A relational database $DB = \{T_{target}, T_1, T_2, \ldots T_n\}$, consisting of a target relation T_{target} (with target variable Y and primary key ID) and a set of background relations T_i,

n denotes the number of background tables

w denotes the number of possible values of Y.

A conventional single-table learning algorithm L.

Output: Class label of unknown tuples.

Procedure

1.Divide tuples of T_{target} into sets $S_{train(k)}$ and $S_{test(k)}$

2.Call algorithm L, providing it with $S_{train(k)}$, obtaining Accuracy for Relation V_0 *(Target Relation)*

3. Propagate Class Label Y and Primary Key ID of each tuple in $S_{train(k)}$ to every background relation T_i, in order to form the data set D_i.

4. Apply aggregation functions on D_i, grouping by ID.

5. Call algorithm L, providing it with D_i, obtain accuracy for Relation V_i.

6. Apply weighted vote to every relation.

7. Assign weight=1/c to lowest accurate Classifier having accuracy >=1/c and assign weight=1 to highest accurate Classifier. The rest of the Classifiers are rated linearly between these extremes and are calculated using following equation:

$weight_i$ = 0 if Misclassification Error >= $\frac{1}{c}$ where c is no of classes

$$weight_i = \frac{(Error_w - Error_i)}{(Error_w - Error_b)} * \left(1 - \frac{1}{c}\right) + \frac{1}{c} \text{ where } c = \text{no of classes}$$

Weight = 1 if $Error_w = Error_b$

Where $Error_b$ is misclassification error of best classifier and $Error_w$ is error of worst classifier.

In this section MRC with weighted voting algorithm is presented.In the 1st step, records of target tables are divided into training set and testing set using k-fold cross validation. Now the algorithm proceeds to construct one multi-relation training set from each background relation, and one from the target table. In multi-relational learning framework, it is important to provide each classifier with sufficient knowledge in order for it to learn the target concept. To achieve this goal, class labels and the tuple IDs are propagated from the target relation to all other background relations. The tuple IDs identify each tuple in the target relation. After constructing the individual training data sets, the algorithm call upon the individual classifiers to learn the target concept from each of these sets. Each classifier constructs a different hypothesis based on the data it is given. Any traditional single-table learning algorithms, such C4.5, naïve bayes can be applied. In this way, all classifiers make different observations on the target concept based on their perspective. After that Best-worst weighted vote is assigned to every classifier as per equation 1.2.

First of all weight = 1/c is assigned to lowest accurate learner and weight = 1 to highest accurate learner. The rest of the learners are rated linearly between these extremes. The equation 1.1 is based on scaling the weights to a range established by the best and the worst classifier that's why named Best-worst weighted vote. Among all the voting schemes tested, the approaches based on scaling the weights to a range established by the best and the worst classifiers have shown the best classification accuracy in most of the data sets.

Description of Proposed Approach

In testing phase, for each training tuples x from the target table, a classifier V_i will retrieve the tuple from background knowledge table T_i corresponding to the key reference. If a corresponding tuples is found, aggregation operations are applied on tuples and observation probability is obtained. Observation probability is Average probabilities distribution for each class label of R_A. Average probability is the probability distribution given to each class label after training. Then voting technique is applied to combine result of different relations. Here weight acquired during voting *is* added to every Observation probability of observations and then sum of probability is applied on observation probability which may return accurate class label to unknown testing tuples.

Proposed Algorithm MRC-WV (Testing Phase)

1. Do for $x = 1, 2, \ldots, m$, where m is the number of tuples in $S_{test(k)}$
2. Classify x using V_0.
3. Return Score Matrix $\{P^{Yq}_{V0}(x)\}$, where $q \in \{1, 2, \ldots w\}$.
4. Do for T = $1, \ldots, V$, where V is the number of relations learned
5. Retrieve corresponding tuple from background table T_i, yielding R_A Classify R_A using V_T.
6. Return Score Matrix $\{P^{Yq}_{VT}(x)\}$, where $q \in \{1, 2, \ldots w\}$.

End do

End Do.

7. Apply weighted voting technique.

Do for i = $1, \ldots, p$, where p is the number of Relations learned

Get Score Matrix $\{P^{Yq}_{Vi}(x)\}$, where $q \in \{1, 2, \ldots w\}$.

8. Add $Weight_i$ to each value of Score Matrix.
9. Apply Sum of probability for final prediction value.

End Do

10. Return class label with highest predication value.

End Procedure.10. Return class label with highest predication value.

End Procedure.

Performance Study

Dataset Details

For algorithm, Relational data is used to test performance evaluation. The three benchmark databases, namely the Financial l [22], Mutagenesis [20], and Thrombosis [21] are used for testing algorithm. All three database comes from different application domains, have variant relational structures, consist of different numbers of tuples in the entire database and in the target relation, and present varying degree of class distribution in the target relation. Next section describes characteristics of these databases.

Financial Database – PKDD 1999 Discovery Challenge [22]

Financial database is from financial domain and was used in the PKDD 1999 discovery challenge. The database was offered by a Czech bank and contains typical business data. The original database is composed of eight tables. The target table, i.e. the *Loan* table consists of 682 records, including a class attribute *status* which indicates the status of the loan. The background information for each loan is stored in the relations *Account*, *Client*, *Order*, *Transaction*, *Card*, *Disposition* and *District*. All background relations relate to the target table through directed or undirected foreign key chains. Learning problem derived from this database is to classify whether the loan is good or bad from the 682 instances. For this database, eight relations, namely, the *loan*, *account*, *client*, *order*, *transaction*, *card*, *disposition*, and *district* were constructed in each task.

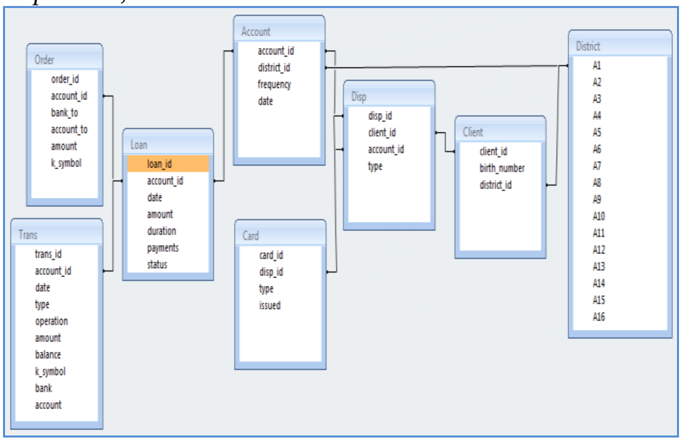

Figure 6. Schema of Financial Database

Table 5. Financial Database

Relations	Tuples	Attributes
Loan (target)	682	7
Account	4500	4
Client	5369	4
Order	6471	6
Disposition	5369	4
District	77	16
Card	892	4
Transaction	1056320	8

Thrombosis Database – PKDD 2001 Discovery Challenge [21]

Thrombosis database is from medical domain and was used in the PKDD 2001 Discovery Challenge. This database is organized using five relations. We set the *Antibody_exam* relation as the target table, and *Thrombosis* as the target concept. This concept describes the different degrees for Thrombosis, i.e. *None*, *Most Severe*, *Severe*, and *Mild*. The target table has 770 records describing the results of special laboratory examinations performed on patients. Our task here is to determine whether or not a patient is thrombosis free. So class value for this database is Negative and Positive. For this task, we include four relations for our background knowledge, namely *Patient info*, *Diagnosis*, *Ana pattern*, and *Thrombosis*. All four background relations are linked to the target table by foreign keys.

Mutagenesis [20]

Mutagenesis data set is composed of the structural descriptions of 188 Regression Friendly molecules that are to be classified as mutagenic or not (Of the 188 instances 125 tuples are positive and 63 are negative). The background relations of this learning problem consist of descriptions about the atoms and bonds that make up the molecules, which include 4893 atoms and 4891 bonds. The *Atom* and *Bond* relations link to the target relation *Molecule* through the *Molecule Atom* relation which only contains key attributes. A summary of the characteristics for the learning data set is given in below tables. For this database, 3 relations, namely *molecule*, *bond*, and *atom* are constructed in the algorithm.

Table 6. Thrombosis Database

Relations	Tuples	Attributes
Antibody_exam (target)	770	10
Patient_info	1229	6
Diagnosis	1240	2
Ana_pattern	770	2
Thrombosis	124	3

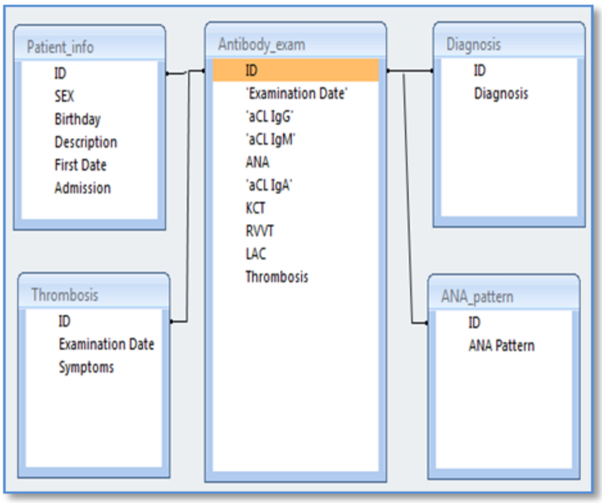

Figure 7. Schema of Thrombosis Database

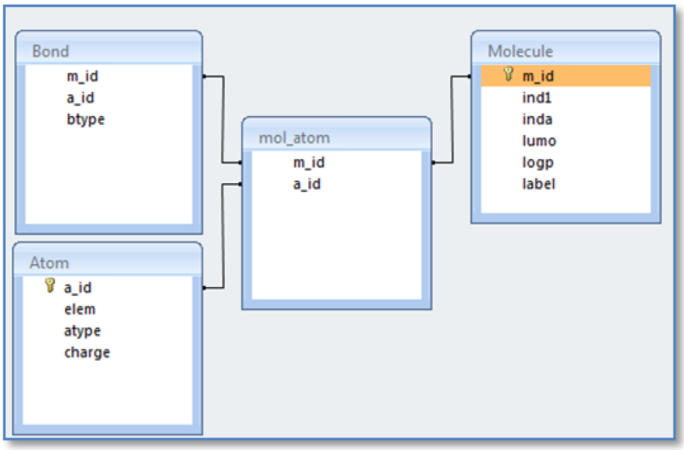

Figure 8. Schema of Mutagenesis Database

Table 7. Mutagenesis Database

Relations	Tuples	Attributes
Molecule (target)	188	6
Atom	4893	4
Bond	4891	3

Experimental Analysis

In this section, results from all three learning task are presented and compared with traditional Approach using Single Table, Flattering Approach ,Upgrade Approach and Multi relational Learning with weighted Voting. Summary of three learning tasks are shown in Table 8 All experiments on MRC with weighted voting algorithm are applied on Decision Tree Classifier. Each of these experiments produces accuracy results using ten-fold cross validation.

Table 8. Summary of dataset used in experiment

Dataset	#tuples in target table	#tables	target class distribution
Financial	682	8	606:76
Mutagenesis	188	3	125:63
Thrombosis	770	5	695:75

Experiment 1:

In this experiment, we examine the performance of the MRC-WV algorithms in terms of accuracy obtained. In this experiment, Decision Tree Classifier is used as a traditional single table classifier, as the propositional learners in Rel Aggs approach and as a classifier by the MRC-WV algorithm. Published Result of well known Upgrading Approach like TILDE and CROSSMINE is used. We present the predictive accuracy obtained for each of the three learning tasks in Table 9.

Table 9. Performance comparison with other approach

Database	Accuracy (%)				
	Single Table Learning	Flattening Approach Rel Aggs	Cross Mine [23]	TILDE [23]	MRC with weighted voting
Financial	86.21	92.08	89.8	81.3	89.44
Mutagenesis	89.89	89.87	87.7	87.7	90.42
Thrombosis	92.59	92.85	90	90.4	93.24

When considering the comparison with the "flattening" based Rel Aggs and Upgrading techniques like CROSS MINE and TILDE the MRC-WV algorithm gives comparable predictive performance as it takes into consideration the efficiency of individual relation into final result. As individual relation has different capability to classify the test tuple, adding appropriate weight to relation will further improve the performance.

So our experimental results here imply that MRC- with weighted voting is improving the predictive performance of the final model induced by the Decision tree method.

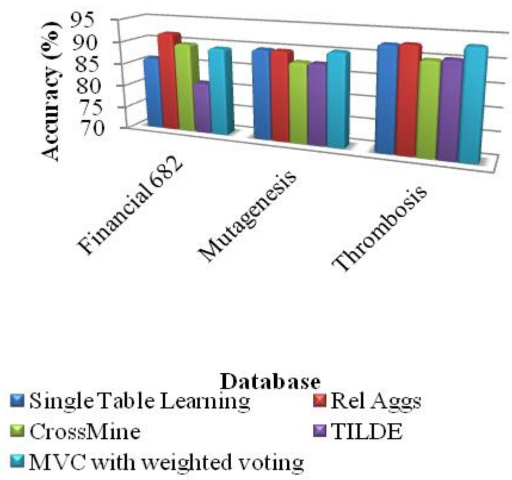

Figure 9. Performance of Naive Bayes and other upgrading approach

The predictive performance results, as presented in Table, show that the MRC with weighted voting algorithm appears to consistently reduce the error rate for almost all of the databases.

Experiment 2:

In this experiment, we examine the performance of the MRC algorithms in terms of True Positive Rate (TP Rate) and False Positive Rate(FP Rate) with traditional and Flattening Algorithm. True positive rate is the proportion of positive tuples that are correctly identified and false positive rate is the proportion of negative tuples that are incorrectly identified. In this experiment, Decision Tree Classifier is used as a classifier by the MRC algorithm. We present the performance measure obtained for each of the three learning tasks in Table.

In this experiment, table represents different values of true positive rate and false positive rate. As different classifier have different capability for positive and negative classes in their relation, using this information and assigning appropriate weight to them before making final decision will improve the performance.

Table 10. Comparison of TP rate and FP rate

Database	Algorithm	Acc	TP Rate (in %)		FP Rate (in %)	
			A	B	A	B
Financial	Single Table Learning	86.21	0.954	0.132	0.868	0.063
	Flattening Approach	92.08	0.937	0.789	0.211	0.063
	MRC with Voting	89.44	100	0.053	0.947	0.0
Mutagenesis	Single Table Learning	89.89	91.2	87.3	12.7	8.8
	Flattening Approach	89.87	92.2	83.7	16.3	7.8
	MRC with Voting	90.42	89.6	92.1	7.9	10.4

Thrombosis	Single Table Learning	92.59	99.00	33.33	66.7	1.00
	Flattening Approach	92.85	98.7	38.7	61.3	1.3
	MRC with Voting	93.24	99	40.0	60.0	1

So In the case of weighted voting algorithm, value of TP rate and FP Rate is better/comparable than Traditional and Rel Aggs algorithm in case of all databases which implies that MRC with weighted voting algorithm is correctly classifying positive tuples as positive and negative tuples as negative.

Conclusion

This paper contains literature study of multi relational classification and various approaches. In this paper, algorithm called 'MRC with weighted voting' is proposed. The proposed algorithm is faster and accurate because of voting technique used in the algorithm. Also in weighted voting, weighted vote is given to relations based on their accuracy and added in probability so that individual performance of relation can be considered based on their contribution.

Future Extensions

Pre-processing techniques can be incorporated to algorithm which removes irrelevant relations and features from database and may improve result. Future work also includes experiment with databases with different and same number of background relations which would result impact of background relations on accuracy and running time. Performance of algorithm for multi class classification is needed to investigate in future.

References

[1] Knobbe A.J., "Multi-Relational Data Mining", Ph.D. dissertation, 2004, http://www.kiminkii.com/thesis.pdf

[2] Dr. M. Thangaraj; C.R.Vijayalakshmi, "A Study on Classification Approaches across Multiple Database Relations", International Journal of Computer Applications (0975 – 8887), Volume 12– No.12, DOI: 10.5120/1740-2366, January 2011

[3] SašoDžeroski, "Multi-relational data mining: an introduction", Published in ACM SIGKDD Explorations Newsletter Homepage archive, Volume 5 Issue 1, July 2003

[4] Yin X; Han J; Yu PS, "CrossMine: Efficient Classification across Multiple Database Relations". In Proceedings of 20th Int. Conf. on Data Engineering (ICDE'04), 2004

[5] Yin X.; Han J.; Yang J.; Yu P.S., "Efficient classification across multiple database relations: a CrossMine approach", IEEE, Knowledge and Data Engineering, IEEE Transactions, Volume: 18 Issue: 6, 01 May 2006

[6] ZiltonCordeiro Junior; Gisele L. Pappa, "A PSO algorithm for improving multi-view classification", IEEE Congress on Evolutionary Computation 925-932, 2011

[7] Yingqin Gu; Hongyan Liu; Jun He; Bo Hu; Xiaoyong Du, "MrCAR: A Multi-relational Classification Algorithm Based on Association Rules", IEEE, Web Information Systems and Mining, 2009. WISM 2009. International Conference, 31 December 2009

[8] Zhen Peng; Lifeng Wu; Xiaoju Wang, "Research on Multi-Relational Classification Approaches", International Conference on Computational Intelligence and Natural Computing, IEEE, 2009

[9] Arno J. Knobbe; Arno Siebes; Danïel van der Wallen, "Multi-relational Decision Tree Induction", PKDD '99 Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery 1999.

[10] Francisco Moreno-Seco; Jose M. Inesta; Pedro J. Ponce de Leon; Luisa Mico, "Comparison of classifier fusion methods for Classification in pattern recognition tasks", 2006

[11] Hongyu Guo; Herna L. Viktor, "Multirelational Classification A Multiple View Approach", ACM, knowledge and information systems, Volume 17, Number 3, 287-312, DOI: 10.1007/ s10115-008-0127-5, 2008

[12] A. Atramentov, H. Leiva, and V. Honavar, "A multirelational decision tree learning algorithm: Implementation and experiments", in Proceedings of the 13th International Conference on Inductive Logic Programming, Springer-Verlag, 2003.

[13] T.Gartner, ASurvey of Kernels for Structured Data, SIGKDD Explor. Newsl., 5(l):49-58, 2003.

[14] J. R. Quinlan and R. M. Cameron-Jones. Foil: A midterm report. In ECML '93: Proceedings of the European Conference on Machine Learning, pages 3-20, 1993.

[15] Blockeel and L. D. Raedt. Top-down induction of first-order logical decision trees. Artificial Intelligence, 101(l-2):285-297, 1998.

[16] X. Yin, J. Han, J. Yang, and P. S. Yu. Crossmine: Efficient classification across multiple database relations. In ICDE '04: Proceedings of the 20th International Conference on Data Engineering, Boston, 2004.

[17] M.-A. Krogel and S. Wrobel. Transformation-based learning using multirelational aggregation. In ILP '01: Proceedings of the 11th International Conference on Inductive Logic Programming, pages 142 155, 2001.

[18] Sergey Tulyakov, Stefan Jaeger, Review of Classifier Combination Methods, Center for Unified Biometrics and Sensors, University at Buffalo, Amherst, NY 14228, USA,2007

[19] S. Dzeroski and L. D. Raedt. Multi-relational data mining: an introduction. SIGKDD Explor. NewsL, 5(1): 1-16, 2003

[20] A. Srinivasan, S. II. Muggleton, M. J. E. Sternberg, and R. D. King. Theories for mutagenicity: a study in first-order and feature-based induction. Artif. Intell., 85(l-2):277-299, 1996.

[21] I. Coursac, N. Duteil, and N. Lucas. PKDD 2001 Discovery Challenge - medical domain. In: The PKDD Discovery Challenge 2001, 3(2), 2002.

[22] P. Berka. Guide to the financial data set. In A. Siebes and P. Berka, editors, PKDD2000 Discovery Challenge, 2000.

[23] X. Yin, J. Han, J. Yang, and P. S. Yu. Crossmine: Efficient classification across multiple database relations. In ICDE '04: Proceedings of the 20th International Conference on Data Engineering, Boston, 2004